

How to write UTF-8 characters using bulk insert in SQL Server?

Asked 12 years, 9 months ago Modified 9 months ago Viewed 134k times



25

I am doing a BULK INSERT into sqlserver and it is not inserting UTF-8 characters into database properly. The data file contains these characters, but the database rows contain garbage characters after bulk insert execution.



My first suspect was the last line of the format file:



```
10.0
3
1 SQLCHAR 0 0 "{|}" 1 INSTANCEID ""
2 SQLCHAR 0 0 "{|}" 2 PROPERTYID ""
3 SQLCHAR 0 0 "[|]" 3 CONTENTTEXT "SQL_Latin1_General_CP1_CI_AS"
```

But, after reading [this official page](#) it seems to me that this is actually a bug in reading the data file by the insert operation in SQL Server version 2008. We are using version 2008 R2.

What is the solution to this problem or at least a workaround?

[sql-server](#) [utf-8](#) [sql-server-2008-r2](#) [bulkinsert](#)

Share Improve this question Follow

asked Mar 31, 2011 at 9:53



Goran Jovic

9,448 3 45 75



Your privacy

By clicking "Accept all cookies", you agree Stack Exchange can store cookies on your device and disclose information in accordance with our [Cookie Policy](#).

Accept all cookies

Necessary cookies only

Customize settings

Highest score (default)





67



I came here before looking for a solution for bulk inserting special characters. Didn't like the workaround with UTF-16 (that would double the size of csv file). I found out that you definitely CAN and it's very easy, you don't need a format file. This answer is for other people who are looking for the same, since it doesn't seem to be documented well anywhere, and I believe this is a very common issue for non-english speaking people. The solution is: just add CODEPAGE='65001' inside the with statement of the bulk insert. (65001=codepage number for UTF-8). Might not work for all unicode characters as suggested by Michael O, but at least it works perfect for latin-extended, greek and cyrillic, probably many others too.

Note: MSDN documentation says utf-8 is not supported, don't believe it, for me this works perfect in SQL server 2008, didn't try other versions however.

e.g.:

```
BULK INSERT #myTempTable
FROM 'D:\somefolder\myCSV.txt'+
WITH
(
    CODEPAGE = '65001',
    FIELDTERMINATOR = '|',
    ROWTERMINATOR = '\n'
);
```

If all your special characters are in 160-255 (iso-8859-1 or windows-1252), you could also use:

```
BULK INSERT #myTempTable
FROM 'D:\somefolder\myCSV.txt'+
WITH
(
    CODEPAGE = 'ACP',
```



Your privacy

By clicking "Accept all cookies", you agree Stack Exchange can store cookies on your device and disclose information in accordance with our [Cookie Policy](#).

Accept all cookies

Necessary cookies only

[Customize settings](#)

answered May 23, 2013 at 14:07



Tom-K

671 1 5 3

insert with codepage 65001: The
either way the data I need to import is
[r](#) Mar 26, 2015 at 16:18

, and SQL2014 SP2 – [Kristen](#) May 14,

[.com/en-gb/help/3136780/...](#) - i.e. this
[an](#) May 21, 2018 at 10:36



You can't. You should first use a N type data field, convert your file to UTF-16 and then import it. The database does not support UTF-8.

33



Share Improve this answer Follow

answered Jul 27, 2011 at 6:48



Michael-O

18.2k 6 55 122



+50



I was afraid of that. Thanks for confirmation – [Goran Jovic](#) Jul 27, 2011 at 17:02

- 2 I had some serious trouble while setting up a data warehouse with SQL Server 2008 and Analysis Services last year. I wanted to insert a huge CSV file into the database with bulk insert and after hours of trying, I realized that the database knows only [Unicode BMP](#) which is a subset of UTF-16. So my entire file had to be recoded with `iconv` in Unix first, then the import went smoothly. See [here](#) arguments => CODEPAGE, code_page and DATAFILETYPE, widenative – [Michael-O](#) Jul 27, 2011 at 17:19

That's pretty much my requirement. Thanks! You saved me a lot of time – [Goran Jovic](#) Jul 28, 2011 at 9:00

- 7 code page 65001 (UTF-8 encoding) is supported in SQL2016, and SQL2014 SP2 (also mentioned in other answers, but they are well down the page and might be missed) – [Kristen](#) May 14, 2017 at 13:45

I have sql 2016, I open the file in notepad ++ , in tab Encoding I see what's the encoding that was recognized. e.g. Windows-1255 so the code is: `WITH (CODEPAGE = '1255', FIELDTERMINATOR = ',', ROWTERMINATOR = '\n', FIRSTROW=2)`, works perfect! – [AJ AJ](#) Mar 17, 2021 at 10:55



Your privacy

By clicking "Accept all cookies", you agree Stack Exchange can store cookies on your device and disclose information in accordance with our [Cookie Policy](#).

Accept all cookies

Necessary cookies only

Customize settings



11



1. In excel save file as CSV(Comma delimited)
2. Open saved CSV file in notepad++
3. Encoding -> Convert to UCS-2 Big Endian
4. Save

```
BULK INSERT #tmpData
FROM 'C:\Book2.csv'
WITH
(
    FIRSTROW = 2,
    FIELDTERMINATOR = ';', --CSV field delimiter
    ROWTERMINATOR = '\n', --Use to shift the control to next row
    TABLOCK
);
```

Done.

Share Improve this answer

Follow

edited Mar 17 at 2:58



Delfino

979 4 21 46

answered Jan 7, 2015 at 8:51



Janis Rudovskis

193 3 9

It this case you should have all strings as **nvarchar** – [beloblotskiy](#) May 20, 2015 at 21:47

I don't think it has to be big endian, it can be little endian instead. So either of the utf-16 encodings are supported. Sidepoints- as you probably know, (though for the sake of any readers),you don't need TABLOCK and CSVs are normally ";" as field separator. – [barlop](#) Mar 29, 2018 at 10:51



Microsoft just added UTF-8 support to SQL Server 2014 SP2:



answered Jul 12, 2016 at 16:36



Jon Morisi

101 1 1

Your privacy

By clicking "Accept all cookies", you agree Stack Exchange can store cookies on your device and disclose information in accordance with our [Cookie Policy](#).

Accept all cookies

Necessary cookies only

[Customize settings](#)



You can re-encode the data file with UTF-16. That's what I did anyway.

4

Share Improve this answer Follow

answered Apr 19, 2011 at 11:16



sayap

6,169 2 36 40



3 Are you saying that it handles UTF-16 encoding properly but not UTF-8. If that's the only problem, it's cool. – Goran Jovic Apr 19, 2011 at 11:55



Use these options - DATAFILETYPE='char' and CODEPAGE = '1252'

2

Share Improve this answer

edited Jun 26, 2015 at 20:34

answered Jun 26, 2015 at 19:20

Follow



Atanu Roy

1,384 2 17 29



Heber

21 1



1 A brief explanation would improve this answer. – user3717023 Jun 27, 2015 at 2:16



Note that as of Microsoft SQL Server 2016, UTF-8 is supported by [bcp](#) , [BULK_INSERT](#) [\(as was part of the original question\)](#), and [OPENROWSET](#) .

2

answered Jul 19, 2016 at 21:58



Charles Burns

10.4k 7 64 83



Your privacy

By clicking "Accept all cookies", you agree Stack Exchange can store cookies on your device and disclose information in accordance with our [Cookie Policy](#).

Accept all cookies

Necessary cookies only

[Customize settings](#)

r the unicode data?

answered Mar 31, 2011 at 10:34



AdaTheDev

144k 28 208 200



Thought I would add my thoughts to this. We were trying to load data into SqlServer using bcp and had lots of trouble.

1



bcp does not, in most versions, support any type of UTF-8 files. We discovered that UTF-16 would work, but it is more complex than is shown in these posts.

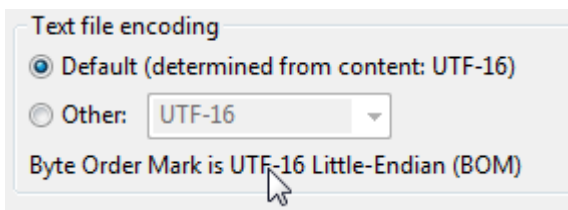


Using Java we wrote the file using this code:



```
PrintStream fileStream = new PrintStream(NEW_TABLE_DATA_FOLDER + fileName, "x-UTF-16LE-BOM");
```

This gave us the correct data to insert.



We tried using just UTF16 and kept getting EOF errors. This is because we were missing the BOM part of the file. From Wikipedia:

UTF-16, a BOM (U+FEFF) may be placed as the first character of a file or character stream to indicate the endianness (byte order) of all the 16-bit code units of the file or stream.

If these bytes are not present, the file won't work. So we have the file, but there is one more secret that needs to be addressed. When constructing your command line you must include -w to tell bcp what type of data it is. When using just English data, you can use -c



.txt -S
me -P password -w

Your privacy

By clicking "Accept all cookies", you agree Stack Exchange can store cookies on your device and disclose information in accordance with our [Cookie Policy](#).

Accept all cookies

Necessary cookies only

Customize settings

ZH

发动机机油滤清器堵塞减量



DA	Sikkerhedsstop pga. tilstoppet motoroliefilter
DE	Abschaltung Motorölfilterverstopfung

Share Improve this answer Follow

answered Mar 15, 2017 at 19:04



[markthegrea](#)

3,733 7 56 80

- 1 You language codes are wrong. As per ISO standard, language codes are always lowercase, but country codes are uppercase. – [Michael-O](#) Jun 12, 2017 at 8:40



1

Only for to share, I had a similar problem, I had portugues accents in a file and bcp imported garbage chars.(e.g. À became ¯) I tried -C with almost all codepages without success. After hours I found a hint on the bcp MS help page.



Format File codepages are having priority over the -C attribute



Means that in the format file I had to use "" like in LastName, once I changed the codepage, the attribute -C 65001 imported the UTF8 file without any problem

```
13.0
4
1      SQLCHAR          0      7      ", "      1      PersonID
""
2      SQLCHAR          0      25     ", "      2      FirstName
SQL_Latin1_General_CP1_CI_AS
3      SQLCHAR          0      30     ", "      3      LastName
""
4      SQLCHAR          0      11     "\r\n"    4      BirthDate
""
```

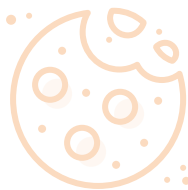
Share Improve this answer Follow

answered Nov 14, 2019 at 19:45



[Martin Lietz](#)

113 2 8



tion instead of OLEDB.

answered Jan 27, 2015 at 11:57



[JYatesDBA](#)

25 1 10

Your privacy

By clicking "Accept all cookies", you agree Stack Exchange can store cookies on your device and disclose information in accordance with our [Cookie Policy](#).

Accept all cookies

Necessary cookies only

[Customize settings](#)



My exported data are in TSV format from DB which has Latin-1 encoding.

0

This easy to check: `SELECT DATABASEPROPERTYEX('DB', 'Collation') SQLCollation;`



Extract file is in UTF-8 format.

BULK INSERT isn't working with UTF-8, so I convert UTF-8 to ISO-8859-1 (aka Latin-1) with simple Clojure script:

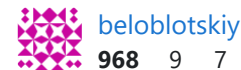


```
(spit ".\\dump\\file1.txt"
      (slurp ".\\dump\\file1_utf8.txt" :encoding "UTF-8")
      :encoding "ISO-8859-1")
```

To execute - correct paths and `java.exe -cp clojure-1.6.0.jar clojure.main utf8_to_Latin1.clj`

Share Improve this answer Follow

answered May 20, 2015 at 21:42



968 9 7



I have tested the bulk insertion with UTF -8 Format. It works fine in Sql Server 2012.

0

```
string bulkInsertQuery = @"DECLARE @BulkInsertQuery NVARCHAR(max) = 'bulk insert
[dbo].[temp_Lz_Post_Obj_Lvl_0]
FROM 'C:\\Users\\suryan\\Desktop\\SIFT
JOB\\New folder\\POSTdata_OBJ5.dat''
WITH ( FIELDTERMINATOR = '''+ CHAR(28) +
''', ROWTERMINATOR = '''+CHAR(10) + ''')'
EXEC SP_EXECUTESQL @BulkInsertQuery";
```



I was using * DAT file with FS as column separator



answered Aug 12, 2015 at 7:44



suryanarayan baliarsingh

1

Your privacy

By clicking "Accept all cookies", you agree Stack Exchange can store cookies on your device and disclose information in accordance with our [Cookie Policy](#).

Accept all cookies

Necessary cookies only

[Customize settings](#)

j. codepage 65001)? Or did you just
[oda](#) Feb 3, 2016 at 7:29