# Import and export data from SQL Server and Azure SQL Database

You can use a variety of methods to import data to, and export data from, SQL Server and Azure SQL Database. These methods include Transact-SQL statements, command-line tools, and wizards.

You can also import and export data in a variety of data formats. These formats include flat files, Excel, major relational databases, and various cloud services.

## Methods for importing and exporting data

### Use Transact-SQL statements

You can import data with the BULK INSERT or the OPENROWSET(BULK...) commands.

SQL Server and Microsoft Windows can be configured to enable an instance of SQL Server to connect to another instance of SQL Server by forwarding the credentials of an authenticated Windows user. This arrangement is known as impersonation or delegation. Understanding how SQL Server version handle security for user impersonation is important when you use BULK INSERT or OPENROWSET. User impersonation allows the data file to reside on a different computer than either the SQL Server process or the user. For example, if a user on **Computer\_A** has access to a data file on **Computer\_B**, and the delegation of credentials has been set appropriately, the user can connect to an instance of SQL Server that is running on **Computer\_C**, access the data file on **Computer\_B**, and bulk import data from that file into a table on **Computer\_C**.

When executing the BULK INSERT statement by using **sqlcmd** or **osql**, from one computer, inserting data into SQL Server on a second computer, and specifying a *data\_file* on third computer by using a UNC path, you may receive a 4861 error.

To resolve this error, use SQL Server Authentication and specify a SQL Server login that uses the security profile of the SQL Server process account, or configure Windows to enable security account delegation.

BULK INSERT [ database\_name . [ schema\_name ] . | schema\_name . ] [ table\_name | view\_name ]

FROM 'data\_file'

[ WITH ( [ [ , ] BATCHSIZE = batch\_size ]

[ [ , ] CHECK\_CONSTRAINTS ]

[ [ , ] CODEPAGE = { 'ACP' | 'OEM' | 'RAW' | 'code\_page' } ]

[ [ , ] DATAFILETYPE =

{ 'char' | 'native'| 'widechar' | 'widenative' } ]

[ [ , ] DATASOURCE = 'data\_source\_name' ]

[ [ , ] ERRORFILE = 'file\_name' ]

[ [ , ] ERRORFILE\_DATASOURCE = 'data\_source\_name' ]

[ [ , ] FIRSTROW = first\_row ]

[ [ , ] FIRE\_TRIGGERS ]

[ [ , ] FORMATFILE\_DATASOURCE = 'data\_source\_name' ]

[ [ , ] KEEPIDENTITY ]

[ [ , ] KEEPNULLS ]

[ [ , ] KILOBYTES\_PER\_BATCH = kilobytes\_per\_batch ]

[ [ , ] LASTROW = last\_row ]

[ [ , ] MAXERRORS = max\_errors ]

[ [ , ] ORDER ( { column [ ASC | DESC ] } [ ,...n ] ) ]

[ [ , ] ROWS\_PER\_BATCH = rows\_per\_batch ]

[ [ , ] ROWTERMINATOR = 'row\_terminator' ]

[ [ , ] TABLOCK ]

-- input file format options

[ [ , ] FORMAT = 'CSV' ]

[ [ , ] FIELDQUOTE = 'quote\_characters']

[ [ , ] FORMATFILE = 'format\_file\_path' ]

[ [ , ] FIELDTERMINATOR = 'field\_terminator' ]

[ [ , ] ROWTERMINATOR = 'row\_terminator' ]

)]

BULK INSERT SalesOrderDetail FROM '\\computer2\salesforce\dailyorders\neworders.txt';

The following format file uses the SQLFLT8 data type to map the second data field to the second column:

Copy

<?xml version="1.0"?>

<BCPFORMAT xmlns="http://schemas.microsoft.com/sqlserver/2004/bulkload/format" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">

<RECORD>

<FIELD ID="1" xsi:type="CharTerm" TERMINATOR="\t" MAX\_LENGTH="30"/>

<FIELD ID="2" xsi:type="CharTerm" TERMINATOR="\r\n" MAX\_LENGTH="30"/> </RECORD> <ROW>

<COLUMN SOURCE="1" NAME="c1" xsi:type="SQLFLT8"/>

<COLUMN SOURCE="2" NAME="c2" xsi:type="SQLFLT8"/> </ROW> </BCPFORMAT>

To use this format file (using the file name C:\t\_floatformat-c-xml.xml) to import the test data into the test table, issue the following Transact-SQL statement:

Copy

BULK INSERT bulktest..t\_float

FROM 'C:\t\_float-c.dat' WITH (FORMATFILE='C:\t\_floatformat-c-xml.xml');

GO

## Import a JSON document into a single column

SELECT BulkColumn

FROM OPENROWSET (BULK 'C:\JSON\Books\book.json', SINGLE\_CLOB) as j

## Import multiple JSON documents

DECLARE @i INT = 1

DECLARE @json AS NVARCHAR(MAX)

WHILE(@i < 10)

BEGIN

SET @file = 'C:\JSON\Books\book' + cast(@i AS VARCHAR(5)) + '.json';

SELECT @json = BulkColumn FROM OPENROWSET (BULK (@file), SINGLE\_CLOB) AS j

SELECT \* FROM OPENJSON(@json) AS json

-- Optionally, save the JSON text in a table.

SET @i = @i + 1 ;

END

By default, triggers are not fired. To fire triggers explicitly, use the FIRE\_TRIGGER option.

Disabling constraints is the default behavior. To check constraints explicitly, use the CHECK\_CONSTRAINTS option.

BULK INSERT SalesOrderDetail FROM 'f:\orders\lineitem.tbl'

WITH ( FIELDTERMINATOR =' |', ROWTERMINATOR = ':\n', FIRE\_TRIGGERS );

INSERT INTO T(XmlCol) SELECT \* FROM OPENROWSET(

BULK 'c:\SampleFolder\SampleData3.txt', SINGLE\_BLOB) AS x;

### Use BCP from the command prompt

bcp [[database\_name.](https://docs.microsoft.com/en-us/sql/tools/bcp-utility#db_name)] [schema](https://docs.microsoft.com/en-us/sql/tools/bcp-utility#schema).{[table\_name](https://docs.microsoft.com/en-us/sql/tools/bcp-utility#tbl_name) | [view\_name](https://docs.microsoft.com/en-us/sql/tools/bcp-utility#vw_name) | ["query"](https://docs.microsoft.com/en-us/sql/tools/bcp-utility#query)

{[in](https://docs.microsoft.com/en-us/sql/tools/bcp-utility#in) [data\_file](https://docs.microsoft.com/en-us/sql/tools/bcp-utility#data_file) | [out](https://docs.microsoft.com/en-us/sql/tools/bcp-utility#out) [data\_file](https://docs.microsoft.com/en-us/sql/tools/bcp-utility#data_file) | [queryout](https://docs.microsoft.com/en-us/sql/tools/bcp-utility#qry_out) [data\_file](https://docs.microsoft.com/en-us/sql/tools/bcp-utility#data_file) | [format](https://docs.microsoft.com/en-us/sql/tools/bcp-utility#format) [nul](https://docs.microsoft.com/en-us/sql/tools/bcp-utility#format)}

[[-a packet\_size](https://docs.microsoft.com/en-us/sql/tools/bcp-utility#a)]

[[-b batch\_size](https://docs.microsoft.com/en-us/sql/tools/bcp-utility#b)]

[[-c](https://docs.microsoft.com/en-us/sql/tools/bcp-utility#c)]

[[-C { ACP | OEM | RAW | code\_page }](https://docs.microsoft.com/en-us/sql/tools/bcp-utility#C) ]

[[-d database\_name](https://docs.microsoft.com/en-us/sql/tools/bcp-utility#d)]

[[-e err\_file](https://docs.microsoft.com/en-us/sql/tools/bcp-utility#e)]

[[-E](https://docs.microsoft.com/en-us/sql/tools/bcp-utility#E)]

[[-f format\_file](https://docs.microsoft.com/en-us/sql/tools/bcp-utility#f)]

[[-F first\_row](https://docs.microsoft.com/en-us/sql/tools/bcp-utility#F)]

[[-G Azure Active Directory Authentication](https://docs.microsoft.com/en-us/sql/tools/bcp-utility#G)]

[[-h"hint [,...n]"](https://docs.microsoft.com/en-us/sql/tools/bcp-utility#h)]

[[-i input\_file](https://docs.microsoft.com/en-us/sql/tools/bcp-utility#i)]

[[-k](https://docs.microsoft.com/en-us/sql/tools/bcp-utility#k)]

[[-K application\_intent](https://docs.microsoft.com/en-us/sql/tools/bcp-utility#K)]

[[-L last\_row](https://docs.microsoft.com/en-us/sql/tools/bcp-utility#L)]

[[-m max\_errors](https://docs.microsoft.com/en-us/sql/tools/bcp-utility#m)]

[[-n](https://docs.microsoft.com/en-us/sql/tools/bcp-utility#n)]

[[-N](https://docs.microsoft.com/en-us/sql/tools/bcp-utility#N)]

[[-o output\_file](https://docs.microsoft.com/en-us/sql/tools/bcp-utility#o)]

[[-P password](https://docs.microsoft.com/en-us/sql/tools/bcp-utility#P)]

[[-q](https://docs.microsoft.com/en-us/sql/tools/bcp-utility#q)]

[[-r row\_term](https://docs.microsoft.com/en-us/sql/tools/bcp-utility#r)]

[[-R](https://docs.microsoft.com/en-us/sql/tools/bcp-utility#R)]

[[-S [server\_name[\instance\_name]](https://docs.microsoft.com/en-us/sql/tools/bcp-utility#S)]

[[-t field\_term](https://docs.microsoft.com/en-us/sql/tools/bcp-utility#t)]

[[-T](https://docs.microsoft.com/en-us/sql/tools/bcp-utility#T)]

[[-U login\_id](https://docs.microsoft.com/en-us/sql/tools/bcp-utility#U)]

[[-v](https://docs.microsoft.com/en-us/sql/tools/bcp-utility#v)]

[[-V (80 | 90 | 100 | 110 | 120 | 130 )](https://docs.microsoft.com/en-us/sql/tools/bcp-utility#V) ]

[[-w](https://docs.microsoft.com/en-us/sql/tools/bcp-utility#w)]

[[-x](https://docs.microsoft.com/en-us/sql/tools/bcp-utility#x)]

#### Copying table rows into a data file (with a trusted connection)

bcp WideWorldImporters.Warehouse.StockItemTransactions out D:\BCP\StockItemTransactions\_character.bcp -c –T

bcp WideWorldImporters.Warehouse.StockItemTransactions OUT D:\BCP\StockItemTransactions\_native.bcp -m 1 -n -e D:\BCP\Error\_out.log -o D:\BCP\Output\_out.log -S –T

#### Copying table rows into a data file (with mixed-mode authentication)

bcp WideWorldImporters.Warehouse.StockItemTransactions out D:\BCP\StockItemTransactions\_character.bcp -c -U<login\_id> -S<server\_name\instance\_name>

#### Copying data from a file to a table

bcp WideWorldImporters.Warehouse.StockItemTransactions\_bcp IN D:\BCP\StockItemTransactions\_character.bcp -c –T

bcp WideWorldImporters.Warehouse.StockItemTransactions\_bcp IN D:\BCP\StockItemTransactions\_native.bcp -b 5000 -h "TABLOCK" -m 1 -n -e D:\BCP\Error\_in.log -o D:\BCP\Output\_in.log -S –T

#### Copying a specific row into a data file

bcp "SELECT \* from Application.People WHERE FullName = 'Amy Trefl'" queryout D:\BCP\Amy\_Trefl\_c.bcp -d WideWorldImporters -c –T

#### Copying data from a query to a data fil

bcp "SELECT FullName, PreferredName FROM WideWorldImporters.Application.People ORDER BY FullName" queryout D:\BCP\People.txt -t, -c –T

#### Creating format files

REM non-XML character format

bcp WideWorldImporters.Warehouse.StockItemTransactions format nul -f D:\BCP\StockItemTransactions\_c.fmt -c -T

REM non-XML native format

bcp WideWorldImporters.Warehouse.StockItemTransactions format nul -f D:\BCP\StockItemTransactions\_n.fmt -n -T

REM XML character format

bcp WideWorldImporters.Warehouse.StockItemTransactions format nul -f D:\BCP\StockItemTransactions\_c.xml -x -c -T

### Use the Import Flat File Wizard

### Feature in SQL 2017 from Database -> Task -> Import flat file

### Use the SQL Server Import and Export Wizard