

# Specify Field and Row Terminators (SQL Server)

Article • 04/11/2023

**Applies to:**  [SQL Server](#)  [Azure SQL Database](#)  [Azure SQL Managed Instance](#)  [Azure Synapse Analytics](#)  [Analytics Platform System \(PDW\)](#)

For character data fields, optional terminating characters allow you to mark the end of each field in a data file with a *field terminator* and the end of each row with a *row terminator*. Terminating characters are one way to indicate to programs that read the data file where one field or row ends and another field or row begins.

## Important

When you use native or Unicode native format, use length prefixes rather than field terminators. Native format data can conflict with terminators because a native-format data file is stored in the Microsoft SQL Server internal binary data format.

## Characters Supported As Terminators

The **bcp** command, BULK INSERT statement, and the OPENROWSET bulk rowset provider support a variety of characters as field or row terminators and always look for the first instance of each terminator. The following table lists the supported characters for terminators.

 Expand table

Terminating character	Indicated by
Tab	<code>\t</code>  This is the default field terminator.
Newline character	<code>\n</code>  This is the default row terminator.

Terminating character	Indicated by
Carriage return/line feed	\r
Backslash*	\\
Null terminator (nonvisible terminator)**	\0
Any printable character (control characters are not printable, except null, tab, newline, and carriage return)	(*, A, t, l, and so on)
String of up to 10 printable characters, including some or all of the terminators listed earlier	(**\t**, end, !!!!!!!!!, \t-\n, and so on)

\*Only the t, n, r, 0 and '\0' characters work with the backslash escape character to produce a control character.

\*\*Even though the null control character (\0) is not visible when printed, it is a distinct character in the data file. This means that using the null control character as a field or row terminator is different than having no field or row terminator at all.

### Important

If a terminator character occurs within the data, it is interpreted as a terminator, not as data, and the data after that character is interpreted as belonging to the next field or record. Therefore, choose your terminators carefully to make sure that they never appear in your data. For example, a low surrogate field terminator would not be a good choice for a field terminator if the data contains that low surrogate.

## Use Row Terminators

The row terminator can be the same character as the terminator for the last field. Generally, however, a distinct row terminator is useful. For example, to produce tabular output, terminate the last field in each row with the newline character (\n) and all other fields with the tab character (\t). To place each data record on its own line in the data file, specify the combination \r\n as the row terminator.

### Note

When you use **bcp** interactively and specify \n (newline) as the row terminator, **bcp** automatically prefixes it with a \r (carriage return) character, which results in

a row terminator of `\r\n`.

## Specify Terminators for Bulk Export

When you bulk export **char** or **nchar** data, and want to use a non-default terminator, you must specify the terminator to the **bcp** command. You can specify terminators in any of the following ways:

- With a format file that specifies the terminator on a field-by-field basis.

### ⓘ Note

For information about how to use format files, see [Format Files for Importing or Exporting Data \(SQL Server\)](#).

- Without a format file, the following alternatives exist:
  - Using the **-t** switch to specify the field terminator for all the fields except the last field in the row and using the **-r** switch to specify a row terminator.
  - Using a character-format switch (**-c** or **-w**) without the **-t** switch, which sets the field terminator to the tab character, `\t`. This is the same as specifying **-t\t**.

### ⓘ Note

If you specify the **-n** (native data) or **-N** (Unicode native) switch, terminators are not inserted.

- If an interactive **bcp** command contains the **in** or **out** option without either the format file switch (**-f**) or a data-format switch (**-n**, **-c**, **-w**, or **-N**), and you have chosen not to specify prefix length and field length, the command prompts for the field terminator of each field, with a default of none:

```
Enter field terminator [none]:
```

Generally, the default is a suitable choice. However, for **char** or **nchar** data fields, see the following subsection, "Guidelines for Using Terminators." For an example that shows this prompt in context, see [Specify Data Formats for Compatibility when Using bcp \(SQL Server\)](#).

### ⚠ Note

After you interactively specify all of the fields in a **bcp** command, the command prompts you save your responses for each field in a non-XML format file. For more information about non-XML format files, see [Non-XML Format Files \(SQL Server\)](#).

## Guidelines for Using Terminators

In some situations, a terminator is useful for a **char** or **nchar** data field. For example:

- For a data column that contains a null value in a data file that will be imported into a program that does not understand the prefix length information.

Any data column that contains a null value is considered variable length. In the absence of prefix lengths, a terminator is necessary to identify the end of a null field, making sure that the data is correctly interpreted.

- For a long fixed-length column whose space is only partially used by many rows.

In this situation, specifying a terminator can minimize storage space allowing the field to be treated as a variable-length field.

## Specify `\n` as a Row Terminator for Bulk Export

When you specify `\n` as a row terminator for bulk export, or implicitly use the default row terminator, bcp outputs a carriage return-line feed combination (CRLF) as the row terminator. If you want to output a line feed character only (LF) as the row terminator - as is typical on Unix and Linux computers - use hexadecimal notation to specify the LF row terminator. For example:

Windows Command Prompt

```
bcp -r '0x0A'
```

## Examples

This example bulk exports the data from the

`AdventureWorks.HumanResources.Department` table to the `Department-c-t.txt` data file

using character format, with a comma as a field terminator and the newline character (\n) as the row terminator.

The **bcp** command contains the following switches.

 Expand table

Switch	Description
-c	Specifies that the data fields be loaded as character data.
-t ,	Specifies a comma (,) as the field terminator.
-r \n	Specifies the row terminator as a newline character. This is the default row terminator, so specifying it is optional.
-T	Specifies that the <b>bcp</b> utility connects to SQL Server with a trusted connection using integrated security. If -T is not specified, you need to specify -U and -P to successfully log in.

For more information, see [bcp Utility](#).

At the Microsoft Windows command prompt enter:

Windows Command Prompt

```
bcp AdventureWorks.HumanResources.Department out C:\myDepartment-c-t.txt  
-c -t, -r \n -T
```

This creates `Department-c-t.txt`, which contains 16 records with four fields each. The fields are separated by a comma.

## Specify Terminators for Bulk Import

When you bulk import **char** or **nchar** data, the bulk-import command must recognize the terminators that are used in the data file. How terminators can be specified depends on the bulk-import command, as follows:

- **bcp**

Specifying terminators for an import operation uses the same syntax as for an export operation. For more information, see "Specifying Terminators for Bulk Export," earlier in this topic.

- **BULK INSERT**

Terminators can be specified for individual fields in a format file or for the whole data file by using the qualifiers shown in the following table.

 Expand table

Qualifier	Description
FIELDTERMINATOR = <i>'field_terminator'</i>	<p>Specifies the field terminator to be used for character and Unicode character data files.</p> <p>The default is \t (tab character).</p>
ROWTERMINATOR = <i>'row_terminator'</i>	<p>Specifies the row terminator to be used for character and Unicode character data files.</p> <p>The default is \n (newline character).</p>

For more information, see [BULK INSERT \(Transact-SQL\)](#).

- INSERT ... SELECT \* FROM OPENROWSET(BULK...)

For the OPENROWSET bulk rowset provider, terminators can be specified only in the format file (which is required except for large-object data types). If a character data file uses a non-default terminator, it must be defined in the format file. For more information, see [Create a Format File \(SQL Server\)](#) and [Use a Format File to Bulk Import Data \(SQL Server\)](#).

For more information about the OPENROWSET BULK clause, see [OPENROWSET \(Transact-SQL\)](#).

## Specify \n as a Row Terminator for Bulk Import

When you specify \n as a row terminator for bulk import, or implicitly use the default row terminator, bcp and the BULK INSERT statement expect a carriage return-line feed combination (CRLF) as the row terminator. If your source file uses a line feed character only (LF) as the row terminator - as is typical in files generated on Unix and Linux computers - use hexadecimal notation to specify the LF row terminator. For example, in a BULK INSERT statement:

SQL

ROWTERMINATOR = '0x0A'

## Examples

The examples in this section bulk import character data from the `Department-c-t.txt` data file created in the preceding example into the `myDepartment` table in the `AdventureWorks2022` sample database. Before you can run the examples, you must create this table. To create this table under the **dbo** schema, in SQL Server Management Studio Query Editor, execute the following code:

SQL

```
USE AdventureWorks2022;
GO
DROP TABLE myDepartment;
CREATE TABLE myDepartment
(DepartmentID smallint,
Name nvarchar(50),
GroupName nvarchar(50) NULL,
ModifiedDate datetime not NULL CONSTRAINT DF_AddressType_ModifiedDate
DEFAULT (GETDATE()))
);
GO
```

### A. Use bcp to interactively specify terminators

The following example bulk imports the `Department-c-t.txt` data file using a `bcp` command. This command uses the same command switches as the bulk export command. For more information, see "Specifying Terminators for Bulk Export," earlier in this topic.

At the Windows command prompt enter:

Windows Command Prompt

```
bcp AdventureWorks2022..myDepartment in C:\myDepartment-c-t.txt -c -t ,
-r \n -T
```

### B. Use BULK INSERT to interactively specify terminators

The following example bulk imports the `Department-c-t.txt` data file using a `BULK INSERT` statement that uses the qualifiers shown in the following table.

 Expand table

Option	Attribute
DATAFILETYPE = 'char'	Specifies that the data fields be loaded as character data.
FIELDTERMINATOR = ' , '	Specifies a comma ( , ) as the field terminator.
ROWTERMINATOR = ' \n '	Specifies the row terminator as a newline character.

In SQL Server Management Studio Query Editor, execute the following code:

SQL

```
USE AdventureWorks2022;
GO
BULK INSERT myDepartment FROM 'C:\myDepartment-c-t.txt'
WITH (
    DATAFILETYPE = 'char',
    FIELDTERMINATOR = ', ',
    ROWTERMINATOR = '\n'
);
GO
```

## Next steps

- [bcp Utility](#)
- [BULK INSERT \(Transact-SQL\)](#)
- [OPENROWSET \(Transact-SQL\)](#)
- [Specify Field Length by Using bcp \(SQL Server\)](#)
- [Specify Prefix Length in Data Files by Using bcp \(SQL Server\)](#)
- [Specify File Storage Type by Using bcp \(SQL Server\)](#)