

Use a format file to map table columns to data-file fields (SQL Server)

Article • 04/04/2023

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

A data file can contain fields arranged in a different order from the corresponding columns in the table. This topic presents both non-XML and XML format files that have been modified to accommodate a data file whose fields are arranged in a different order from the table columns. The modified format file maps the data fields to their corresponding table columns. Please review [Create a Format File \(SQL Server\)](#) for additional information.

Note

Either a non-XML or XML format file can be used to bulk import a data file into the table by using a **bcp utility** command, **BULK INSERT** statement, or **INSERT ... SELECT * FROM OPENROWSET(BULK...)** statement. For more information, see [Use a Format File to Bulk Import Data \(SQL Server\)](#).

Note

This syntax, including bulk insert, is not supported in Azure Synapse Analytics. In Azure Synapse Analytics and other cloud database platform integrations, accomplish data movement via [the COPY statement in Azure Data Factory](#), or by using [T-SQL statements such as COPY INTO and PolyBase](#).

Example test conditions

The examples of modified format files in this topic are based on the table and data file defined below.

Sample table

The script below creates a test database and a table named `myRemap`. Execute the following Transact-SQL in Microsoft SQL Server Management Studio (SSMS):

SQL

```
CREATE DATABASE TestDatabase;
GO

USE TestDatabase;
CREATE TABLE myRemap
(
    PersonID smallint,
    FirstName varchar(25),
    LastName varchar(30),
    Gender char(1)
);
```

Sample data file

The data below presents `FirstName` and `LastName` in the reverse order as presented in the table `myRemap`. Using Notepad, create an empty file `D:\BCP\myRemap.bcp` and insert the following data:

Output

```
1,Grosse,Anthony,M
2,Fatnowna,Alica,F
3,Rosenhain,Stella,F
```

Create the format files

To bulk import data from `myRemap.bcp` into the `myRemap` table, the format file must do the following:

- Map the first data field to the first column, `PersonID`.
- Map the second data field to the third column, `LastName`.
- Map the third data field to the second column, `FirstName`.
- Map the fourth data field to the fourth column, `Gender`.

The simplest method to create the format file is by using the [bcp utility](#). First, create a base format file from the existing table. Second, modify the base format file to reflect the actual data file.

Create a non-XML format file

Please review [Non-XML Format Files \(SQL Server\)](#) for detailed information. The following command will use the `bcp` utility to generate a non-xml format file, `myRemap.fmt`, based on the schema of `myRemap`. In addition, the qualifier `c` is used to specify character data, `t`, is used to specify a comma as a field terminator, and `T` is used to specify a trusted connection using integrated security. At a command prompt, enter the following command:

Windows Command Prompt

```
bcp TestDatabase.dbo.myRemap format nul -c -f D:\BCP\myRemap.fmt -t, -T
```

Modify the non-XML format file

See [Structure of Non-XML Format Files](#) for terminology. Open `D:\BCP\myRemap.fmt` in Notepad and perform the following modifications:

- 1. Re-arrange the order of the format-file rows so that the rows are in the same order as the data in `myRemap.bcp`.
- 2. Ensure the host file field order values are sequential.
- 3. Ensure there is a carriage return after the last format-file row.

Compare the changes:

Before

Output						
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
SQL_Latin1_General_CP1_CI_AS						
4	SQLCHAR	0	1	"\r\n"	4	Gender
SQL_Latin1_General_CP1_CI_AS						

After

Output						
13.0						
4						

1	SQLCHAR	0	7	", "	1	PersonID
2	SQLCHAR	0	30	", "	3	LastName
3	SQLCHAR	0	25	", "	2	FirstName
4	SQLCHAR	0	1	"\r\n"	4	Gender

The modified format file now reflects:

- The first data field in `myRemap.bcp` is mapped to the first column, `myRemap.. PersonID`
- The second data field in `myRemap.bcp` is mapped to the third column, `myRemap.. LastName`
- The third data field in `myRemap.bcp` is mapped to the second column, `myRemap.. FirstName`
- The fourth data field in `myRemap.bcp` is mapped to the fourth column, `myRemap.. Gender`

Create an XML format file

Please review [XML Format Files \(SQL Server\)](#) for detailed information. The following command will use the `bcp utility` to create an xml format file, `myRemap.xml`, based on the schema of `myRemap`. In addition, the qualifier `c` is used to specify character data, `t`, is used to specify a comma as a field terminator, and `T` is used to specify a trusted connection using integrated security. The `x` qualifier must be used to generate an XML-based format file. At a command prompt, enter the following command:

Windows Command Prompt

```
bcp TestDatabase.dbo.myRemap format nul -c -x -f D:\BCP\myRemap.xml -t, -T
```

Modify the XML format file

Review [Schema syntax for XML format files](#) for terminology. Open `D:\BCP\myRemap.xml` in Notepad and perform the following modifications:

1. The order in which the `<FIELD>` elements are declared in the format file is the

order in which those fields appear in the data file, thus reverse the order for the <FIELD> elements with ID attributes 2 and 3.

2. Ensure the <FIELD> ID attribute values are sequential.
3. The order of the <COLUMN> elements in the <ROW> element defines the order in which the bulk operation sends them to the target. The XML format file assigns each <COLUMN> element a local name that has no relationship to the column in the target table of a bulk import operation. The order of the <COLUMN> elements is independent of the order of <FIELD> elements in a <RECORD> definition. Each <COLUMN> element corresponds to a <FIELD> element (whose ID is specified in the SOURCE attribute of the <COLUMN> element). Thus, the values for <COLUMN> SOURCE are the only attributes that require revision. Reverse the order for <COLUMN> SOURCE attributes 2 and 3.

Compare the changes:

Before

XML

```
<?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="," MAX_LENGTH="7"/>
  <FIELD ID="2" xsi:type="CharTerm" TERMINATOR="," MAX_LENGTH="25"
COLLATION="SQL_Latin1_General_CP1_CI_AS"/>
  <FIELD ID="3" xsi:type="CharTerm" TERMINATOR="," MAX_LENGTH="30"
COLLATION="SQL_Latin1_General_CP1_CI_AS"/>
  <FIELD ID="4" xsi:type="CharTerm" TERMINATOR="\r\n" MAX_LENGTH="1"
COLLATION="SQL_Latin1_General_CP1_CI_AS"/>
</RECORD>
<ROW>
  <COLUMN SOURCE="1" NAME="PersonID" xsi:type="SQLSMALLINT"/>
  <COLUMN SOURCE="2" NAME="FirstName" xsi:type="SQLVARYCHAR"/>
  <COLUMN SOURCE="3" NAME="LastName" xsi:type="SQLVARYCHAR"/>
  <COLUMN SOURCE="4" NAME="Gender" xsi:type="SQLCHAR"/>
</ROW>
</BCPFORMAT>
```

After

XML

```
<?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="," MAX_LENGTH="7"/>
<FIELD ID="2" xsi:type="CharTerm" TERMINATOR="," MAX_LENGTH="30"
COLLATION="SQL_Latin1_General_CP1_CI_AS"/>
<FIELD ID="3" xsi:type="CharTerm" TERMINATOR="," MAX_LENGTH="25"
COLLATION="SQL_Latin1_General_CP1_CI_AS"/>
<FIELD ID="4" xsi:type="CharTerm" TERMINATOR="\r\n" MAX_LENGTH="1"
COLLATION="SQL_Latin1_General_CP1_CI_AS"/>
</RECORD>
<ROW>
<COLUMN SOURCE="1" NAME="PersonID" xsi:type="SQLSMALLINT"/>
<COLUMN SOURCE="3" NAME="FirstName" xsi:type="SQLVARYCHAR"/>
<COLUMN SOURCE="2" NAME="LastName" xsi:type="SQLVARYCHAR"/>
<COLUMN SOURCE="4" NAME="Gender" xsi:type="SQLCHAR"/>
</ROW>
</BCPFORMAT>
```

The modified format file now reflects:

- FIELD 1, which corresponds to COLUMN 1, is mapped to the first table column, myRemap.. PersonID
- FIELD 2, which corresponds to COLUMN 2, is re-mapped to the third table column, myRemap.. LastName
- FIELD 3, which corresponds to COLUMN 3, is re-mapped to the second table column, myRemap.. FirstName
- FIELD 4, which corresponds to COLUMN 4, is mapped to the fourth table column, myRemap.. Gender

Import data with a format file to map table columns to data-file field

The examples below use the database, datafile, and format files created above.

Use **bcp** and **non-XML format file**

At a command prompt, enter the following command:

Windows Command Prompt

```
bcp TestDatabase.dbo.myRemap IN D:\BCP\myRemap.bcp -f D:\BCP\myRemap.fmt
-T
```

Use **bcp** and **XML format file**

At a command prompt, enter the following command:

Windows Command Prompt

```
bcp TestDatabase.dbo.myRemap IN D:\BCP\myRemap.bcp -f D:\BCP\myRemap.xml  
-T
```

Use **BULK INSERT** and **non-XML format file**

Execute the following Transact-SQL in Microsoft SQL Server Management Studio (SSMS):

SQL

```
USE TestDatabase;  
GO  
  
TRUNCATE TABLE myRemap;  
BULK INSERT dbo.myRemap  
    FROM 'D:\BCP\myRemap.bcp'  
    WITH (FORMATFILE = 'D:\BCP\myRemap.fmt');  
GO  
  
-- review results  
SELECT * FROM TestDatabase.dbo.myRemap;
```

Use **BULK INSERT** and **XML format file**

Execute the following Transact-SQL in Microsoft SQL Server Management Studio (SSMS):

SQL

```
USE TestDatabase;  
GO  
  
TRUNCATE TABLE myRemap;  
BULK INSERT dbo.myRemap  
    FROM 'D:\BCP\myRemap.bcp'  
    WITH (FORMATFILE = 'D:\BCP\myRemap.xml');  
GO  
  
-- review results  
SELECT * FROM TestDatabase.dbo.myRemap;
```

Use **OPENROWSET(BULK...)** and **non-XML format file**

Execute the following Transact-SQL in Microsoft SQL Server Management Studio (SSMS):

SQL

```
USE TestDatabase;
GO

TRUNCATE TABLE myRemap;
INSERT INTO dbo.myRemap
    SELECT *
    FROM OPENROWSET (
        BULK 'D:\BCP\myRemap.bcp',
        FORMATFILE = 'D:\BCP\myRemap.fmt'
    ) AS t1;
GO

-- review results
SELECT * FROM TestDatabase.dbo.myRemap;
```

Use **OPENROWSET(BULK...)** and **XML format file**

Execute the following Transact-SQL in Microsoft SQL Server Management Studio (SSMS):

SQL

```
USE TestDatabase;
GO

TRUNCATE TABLE myRemap;
INSERT INTO dbo.myRemap
    SELECT *
    FROM OPENROWSET (
        BULK 'D:\BCP\myRemap.bcp',
        FORMATFILE = 'D:\BCP\myRemap.xml'
    ) AS t1;
GO

-- review results
SELECT * FROM TestDatabase.dbo.myRemap;
```

Next steps

- [bcp Utility](#)
- [Use a Format File to Skip a Table Column \(SQL Server\)](#)
- [Use a Format File to Skip a Data Field \(SQL Server\)](#)