# U-SQL Error: Unexpected number of columns in input record

Did you run into an error that said “Unexpected number of columns in input record at line <line number>.\nExpected <some number> columns” when you tried to submit your Azure Data Analytics job written in U-SQL? If yes, then read on.

If you ran into the error above, you possibly had U-SQL code in your script that looks similar to

﻿//0. Initialize some parameters/constants

DECLARE @INPUTDIR string = "/DemoData/Ambulance/";

DECLARE @INPUT\_DRIVERS string = @INPUTDIR + "Drivers.txt";

//1. Extract

@Drivers =

EXTRACT driver\_id int

, name string

, street string

, city string

, region string

, zipcode string

, country string

, phone\_numbers string

FROM @INPUT\_DRIVERS

USING Extractors.Tsv(encoding: Encoding.Unicode);

(and more code…)

You could be using Extractors.Csv() or Extractors.Text() as well in this case. To read more about our built-in extractors, please visit the [U-SQL Built-in Extractors documentation](https://msdn.microsoft.com/en-us/library/azure/mt621366.aspx). What you are doing here is using our native extractors that rely on certain delimiters for rows and columns to extract data from the file. E.g. Tsv by default uses tab as a column delimiter and carriage-return as a row delimiter. One requirement that we have is to ensure that the number of columns you specify in your EXTRACT statement match the number of columns (as defined by the delimiter) in one line of the file. In this article, we will suggest a number of ways to fix this behavior.

**Step 1: Perform basic validations on your data file**

* Are you using the right delimiter for the extraction?
* Does the number of delimited values in the file match the number of columns you have specified in your U-SQL script?
* Do you have blank spaces in your file? If yes, try to clean them up or use the silent:true option described in more detail in Step 2, Option 2.
* Do you have metadata in the first few lines of the file that describes the file? If yes, try to clean them up or use the silent:true option described in more detail in Step 2, Option 2.
* Do you have a header row that does not have all the specified columns? If yes, try to clean them up or use the silent:true option described in more detail in Step 2, Option 2.

**Step 2: Use one of the following options to fix the issue depending on your scenario**

**Option 1: Update the number of columns in your EXTRACT statement to match the number of columns in the file.**

This requires an understanding of the input you are processing. Understand the schema of your data by looking at the number of columns in your file and make sure they match the number of columns in your EXTRACT statement.

Even if you required only a subset of columns in the file for your processing, you need to extract all the columns. Also, the default extractor assumes that every row in this file has the specified number of columns and will throw an error. You can then use the SELECT statement to select a subset of columns.

*When to use this*: When the number of columns are manageable and you understand the schema of the data.

*When to not use this*: When you have a much larger number of columns or if you don’t know the full structure of your data and are interested in a small subset.

**Option 2: Use the silent option in your extractor to skip mismatched columns**

You can specify a silent parameter to your extractor that will skip rows in your file that have a mismatched number of columns. This is ideal for scenarios where you know the number of columns that are supposed to be in the file, but you don’t want one corrupt row to block the extraction of the rest of the data. Use this with caution – while you will get out of the syntax error, you might run into interesting semantic errors. The sample with the silent flag included looks like this :-

﻿//0. Initialize some parameters/constants

DECLARE @INPUTDIR string = "/DemoData/Ambulance/";

DECLARE @INPUT\_DRIVERS string = @INPUTDIR + "Drivers.txt";

//1. Extract

@Drivers =

EXTRACT driver\_id int

, name string

, street string

, city string

, region string

, zipcode string

, country string

, phone\_numbers string

FROM @INPUT\_DRIVERS

USING Extractors.Tsv(encoding: Encoding.Unicode, silent: true);

(and more code…)

*When to use this*: The majority of your data confirms with the columns you have specified and its OK to skip data that does not confirm with the standard.

*When to not use this*: You don’t know the structure of the data fully and you do not want data loss.

**Option 3 – Write a custom extractor or a user defined function in C#**

If you are comfortable with C#, you can go ahead and write your own extractor or user defined function in C# and use that to perform additional validations. Please note that the introducing custom code can slow down your processing (since they use managed assemblies).

For more information on how to write custom extractors and user defined functions, please click on [this article](https://azure.microsoft.com/en-us/documentation/articles/data-lake-analytics-u-sql-develop-user-defined-operators/) to learn more. An example U-SQL project that uses custom extractors can be found in this Github project (Check 3-Ambulance-User-Code).

﻿//0. Initialize some parameters/constants

DECLARE @INPUTDIR string = "/DemoData/Ambulance/";

DECLARE @INPUT\_DRIVERS string = @INPUTDIR + "Drivers.txt";

//1. Extract

@Drivers =

EXTRACT driver\_id int

, name string

, street string

, city string

, region string

, zipcode string

, country string

, phone\_numbers string

FROM @INPUT\_DRIVERS

USING AmbulanceDriver.DriverExtractor(col\_delim: ‘\t’, encoding: Encoding.Unicode);

*When to use this*: You are comfortable writing C# code, your performance requirements are flexible.

*When to not use this*: You have strict performance targets.

Hope this article helped you. If you have any questions/concerns/feedback, please reach out to us at [usql@microsoft.com](mailto:usql@microsoft.com).