



ORACLE

Oracle SQL*Loader Express Mode Loading

April, 2024, Version 2.0
Copyright © 2024, Oracle and/or its affiliates
Public

Table of contents

Introduction	3
Overview of CSV Files	3
Loading with Express Mode	4
An Example of Express Loading	4
Customizing SQL*Loader Express Mode	5
Changing the Field Terminator	5
Specifying the Order of the Fields in the Data File	5
Specifying a Data File Name that doesn't match Table Name	5
Using an Enclosure Character	6
Using an Enclosure Character for Fields Containing a New Line	6
Other Customizations	7
Using An Express Mode Log File To Develop A Control File	8
Conclusion	8



Introduction

Creating a SQL*Loader control file can be a challenge. SQL*Loader provides flexibility for loading different types of data files into Oracle Database. However, that flexibility can make SQL*Loader control files complicated. The more fields there are in the data file, the more room there is for errors.

Beginning with Oracle Database 12c, SQL*Loader has a feature called express mode that makes loading CSV files faster and easier. Most data files are in comma-separated values (CSV) format. With express mode, there is no need to write a control file for most CSV files that you load. Instead, you can load the CSV file with just a few parameters on the SQL*Loader command line. This technical brief explains how to get started with express mode.

Benefits of Using Express Mode

SQL*Loader express mode saves the time and effort of creating and testing a SQL*Loader control file. Instead, you specify a single SQL*Loader command with a few parameters to initiate the loading process.

SQL*Loader express mode is self-optimizing. It will use the fastest possible mechanism for loading data files. It will try to use external tables with parallel inserts and the append hint. Parallel inserts allow simultaneous loading from the data file by multiple processes, reducing the elapsed time for the load. If express mode finds that external tables cannot be used for a particular load operation it will choose the next fastest mechanism, a direct path load. The append hint on an insert statement causes Oracle Database to use direct path loading. Direct path loading eliminates much of the Oracle Database INSERT overhead by formatting Oracle data blocks and writing the data blocks directly to the database files.

Overview of CSV Files

Express mode can only load data from CSV files. CSV files have the following characteristics:

CSV files are text files where every record in the file contains data for one row in a table.

The fields in each record are in the same order in every record.

Each record ends with a newline.

On Unix-style operating systems, a newline is the linefeed character. On Microsoft Windows, a newline is a carriage return and a linefeed character.

A character, usually but not necessarily a comma, terminates the value for each field in the record.

The use of a comma is not a requirement. Other characters such as colon or vertical bar can also be used as terminators. For simplicity, we still call these files CSV files even if the terminator character is something other than a comma. The terminator after the last field in the record is optional.

CSV files enclose the field values with a double quote or another character.

The enclosure character is usually the double quote character, but it can be another character. For instance, enclosing a value allows you to have a newline or the field terminator character included as part of the value of the field.

Use of the enclosure character can be optional or mandatory.

CSV files that use the enclosure character can require them around every field or let the use of enclosures be optional. If the use of enclosure is optional, then they are typically used only for field values that contain a newline or the terminator character.

A field can contain the enclosure character if “stutter” syntax is used.

If a CSV file uses enclosure characters, and if a field value contains the enclosure character, then the enclosure character in the field value must be duplicated. This is called “stutter syntax”. It is a way to distinguish the occurrence of the enclosure character in the middle of a field from an enclosure character marking the end of the field.

Loading with Express Mode

The SQL*Loader TABLE parameter triggers express mode. The value of the TABLE parameter is the name of the table that SQL*Loader will load. If TABLE is the only parameter specified, then SQL* loader will do the following:

1. Looks for a data file in the current directory with the same name as the table being loaded that has an extension of .dat. The upper and lower case used in the name of the data file is the same as the case for the table name specified in the TABLE parameter
2. Assumes the order of the fields in the data file matches the order of the columns in the table
3. Assumes the fields are terminated by commas, but there is no enclosure character

An Example of Express Loading

Here is a simple example of a table, a data file and the SQL*Loader command that will load the table.

The table is created with the following command:

```
CREATE TABLE EMP
(EMPNO    number(4) not null,
 ENAME    varchar2(10),
 HIREDATE date,
 DEPTNO   number(2));
```

The data file name used to load the table is derived from the table name, emp, and is named emp.dat. The contents of the data file in this example are as follows:

```
7782,Clark,09-Jun-81,10
7839,King,17-Nov-81,12
```

Notice the order of the fields in the data file matches the order of the columns in the table. The following SQL*Loader command will load the table from the data file.

```
sqlldr userid=scott table=emp
```

Notice that no control file is used. After executing the SQL*Loader command, a SELECT from the table will return the following results:

EMPNO	ENAME	HIREDATE	DEPTNO
7782	Clark	09-JUN-81	10
7839	King	17-NOV-81	10

Customizing SQL*Loader Express Mode

While the SQL*Loader command in the previous example was simple, it used default settings that may not always be true. For example, a CSV file can use a character other than a comma to terminate fields. There may be multiple data files to load that do not use the name of the table in the file names. For these reasons, SQL*Loader Express Mode has additional command line parameters that allow you to customize the loading process.

Changing the Field Terminator

The `TERMINATED_BY` parameter for the SQL*Loader command line specifies the character that is used to terminate the fields. If the data file in the example used vertical bars for separators rather than commas, it would look like this:

```
7782|Clark|09-Jun-81|10
```

```
7839|King|17-Nov-81|12
```

The command used to load this data file is:

```
sqlldr userid=scott table=emp terminated_by='|'
```

Specifying the Order of the Fields in the Data File

When SQL*Loader Express Mode reads the fields from the record in the data file, by default, it expects the order of the fields in the data file is the same as the order of the columns in the table. However, this may not be the case. In this situation, you can use the `FIELD_NAMES` parameter. SQL*Loader will use the first record in the data file to determine the order of the fields in the data file.

The `FIELD_NAMES` parameter has two options. If there are multiple data files being loaded, `FIELD_NAMES=FIRST` tells SQL*Loader that only the first data file in the list has the field name list as its first record. Specifying `FIELD_NAMES=ALL` tells SQL*Loader that all data files have the field names list as the first record.

In this example, the field order in the data file `emp.dat` doesn't match the order of the columns in the table `EMP`. The first row defines the order of the fields in the data.

```
deptno,empno,ename,hiredate
```

```
10,7782,Clark,09-Jun-81
```

```
12,7839,King,17-Nov-81
```

The command used to load this data file is:

```
sqlldr userid=scott table=emp field_names=first
```

Specifying a Data File Name that doesn't match Table Name

The default name for the data file is the name of the table with a `.dat` extension. The `DATA` parameter is used to load a file with a different name or to load multiple files. The `DATA` parameter accepts a list of values, so you can specify multiple files as part of the load.

Beginning with Oracle Database 12c, SQL*Loader supports using the wildcard characters, `"*"` and `"?"` in file names. The `"*"` wild card character matches one or more characters in a file name. The `"?"` wild card character matches just one character in a file name. This simplifies the command string when loading dozens or hundreds of files with similar names.

For example, assume we have two files to load into table emp.

The first file is jan_2012_emp.dat:

```
7782,Clark,09-Jan-12,10
7839,King,17-Jan-12,12
```

The second file is feb_2012_emp.dat:

```
8109,Baby,12-Feb-12,10
8299,Lee,24-Feb-12,12
```

These files can be loaded with either of the two following commands:

```
sqlldr userid=scott table=emp data=jan_2012_emp.dat,feb_2012_emp.dat
sqlldr userid=scott table=emp data="*_2012_emp.dat"
```

The first command lists all the file names to be loaded. The second command uses wild cards in the filenames for the data parameter.

Using an Enclosure Character

A data file may need to use an enclosure character around fields because the fields might contain the terminator character or newline character as part of the value of the field. This is specified using the parameter `optionally_enclosed_by`.

For instance, consider the following table definition:

```
create table part_info
(part_number      varchar2(10),
 part_description varchar2(40)
);
```

The data file part_info.dat contains the following records:

```
10-1002,"size 12 widget, red"
10-1003,"size 12 widget, blue"
```

Table PART_INFO can be loaded with the following command:

```
sqlldr userid=scott table=part_info optionally_enclosed_by=\'\"'
```

This example used `OPTIONALLY_ENCLOSED_BY` because only some of the field values were enclosed by quotes. Use the `ENCLOSED_BY` parameter if every field is enclosed.

After the load, the table will have the following rows.

PART_NUMBER	PART_DESCRIPTION
10-1002	size 12 widget, red
10-1003	size 12 widget, blue

Using an Enclosure Character for Fields Containing a New Line

The enclosure character can be used to enclose fields that also contain newlines. For example, here is a table definition with a large text field.

```
CREATE TABLE EMP_RESUMES
(EMPNO      number(4) not null,
 EMP_RESUME varchar2(1000));
```

The data file contains two records, 7782 and 7839:

```
7782,"line 1 of resume for 7782
line 2 of resume
last line of resume"
7839,"line 1 of resume for 7839
line 2 of resume
last line of resume"
```

Notice that the EMP_RESUME field is enclosed by double quotes and has newlines embedded in the data. In this case, the SQL*Loader command must specify the CSV parameter with a value of WITH_EMBEDDED.

*Note: SQL*Loader does additional processing on files that contain newline characters inside of fields. This adds overhead and time to the loading process.*

The following SQL*Loader command loads the table EMP_RESUMES with the data file above:

```
sqlldr userid=scott table=emp_resumes csv=with_embedded
```

Selecting from the table EMP_RESUMES returns the data loaded from the file:

```
SQL> select * from emp_resumes;
      EMPNO
-----
EMP_RESUME
-----
          7782
line 1 of resume for 7782
line 2 of resume
last line of resume
          7839
line 1 of resume for 7839
line 2 of resume
last line of resume
```

Other Customizations

SQL*Loader Express Mode supports additional parameters to customize a load as noted below. The [Oracle Database Utilities Guide](#) documentation provides details.

- BAD – the file where SQL*Loader writes records that could not be loaded.
- CHARACTERSET – the name of the character set used to encode the data files.
- DATE_FORMAT – the format string to use when interpreting dates in the data file.

- `DEGREE_OF_PARALLELISM` – the degree of parallelism to use when loading with external tables.
- `DIRECT` – force the load to use direct path or conventional path load instead of external tables.
- `EXTERNAL_TABLE = EXECUTE` forces the load to use external tables to load the data file, while `GENERATE_ONLY` tells SQL*Loader to write the SQL statements used to load the external tables to the log file without executing those statements. The SQL statements can be edited, customized and run in SQL*Plus.
- `LOAD` – the number of records to load. The default is to load all records.
- `NULLIF` – the value for fields in the record that will cause SQL*Loader to insert a NULL for that field.
- `SILENT` – control what messages are written to the SQL*Loader log file.
- `TIMESTAMP_FORMAT` – the format string to use when interpreting timestamp fields in the data file.
- `TRIM` – how white space should be trimmed from the beginning and end of a field.

Using An Express Mode Log File To Develop A Control File

SQL*Loader express mode can generate a complex SQL*Loader control file for you, including the DDL for creating an external table. This can be helpful in situations where the express mode customization parameters are insufficient to load a particular file or set of files.

SQL*Loader Express Mode includes the contents of a regular SQL*Loader control file in its logfile. The logfile defaults to "tablename.log." You can save and use the control file information in the log file as is or as the basis for customizing a control file to load the data.

For example, the `DATE_FORMAT` parameter cannot be used if there are different data formats in various date fields in the data file. In this case, you can

1. Use SQL*Loader express mode to write the control file information to the log file
2. Create a new control file with that information, and
3. Modify the data format strings to the correct values for the date fields.

This makes SQL*Loader express mode a good way to learn how to write more complicated SQL*Loader control files.

Conclusion

SQL*Loader Express mode simplifies the effort needed to load CSV files, the most common file format. Express mode is self-optimizing, it will try to use the fastest mechanism for loading data files. Express mode includes optional parameters for customizing an express load and accelerates the development of more complex control files. Consider using SQL*Loader Express mode when you need to load data into Oracle Database from files.

Connect with us

Call **+1.800.ORACLE1** or visit **oracle.com**. Outside North America, find your local office at: **oracle.com/contact**.

 blogs.oracle.com

 facebook.com/oracle

 twitter.com/oracle

Copyright © 0000, Oracle and/or its affiliates. This document is provided for information purposes only, and the contents hereof are subject to change without notice. This document is not warranted to be error-free, nor subject to any other warranties or conditions, whether expressed orally or implied in law, including implied warranties and conditions of merchantability or fitness for a particular purpose. We specifically disclaim any liability with respect to this document, and no contractual obligations are formed either directly or indirectly by this document. This document may not be reproduced or transmitted in any form or by any means, electronic or mechanical, for any purpose, without our prior written permission.

Oracle, Java, MySQL, and NetSuite are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.