

# DB2 LOAD vs. Oracle SQL\*Loader

## A Technical Overview

### Part I - DB2 LOAD vis-à-vis Oracle SQL\*Loader

Many Oracle DBAs using DB2 for the first time naturally lean towards using the DB2 LOAD utility for converting their Oracle SQL\*Loader scripts in a migration of their databases. This article will help Oracle DBAs quickly come up to speed on using the DB2 LOAD utility. It will also provide a roadmap for converting Oracle SQL\*Loader scripts to DB2 LOAD scripts.

#### DB2 LOAD vs. DB2 IMPORT – which “path” do you use?

Actually, DB2 has two different utilities for putting data from an external source into a DB2 table: LOAD and IMPORT. LOAD puts data in at the page level, bypasses trigger firing and logging and puts off constraint checking and index building until after the data has been put into the DB2 table. IMPORT on the other hand basically performs INSERTS and obeys trigger firing, performs logging and does constraint checking and index building **as** it puts data into the table. There are many other differences between the options for these two utilities, however it is out of scope for this article to discuss them here.

**Note:** The DB2 IMPORT utility is not tied to reading a proprietary format as is the Oracle IMPORT utility, so these two utilities should not be compared to each other.

The Oracle SQL\*Loader utility on the other hand has two main modes, or paths, of operation: direct path and conventional path. Oracle DBAs use this one utility, but specify the “path”, which are used for similar reasons as the two DB2 utilities. SQL\*Loader “direct path” mode is similar in functionality to DB2 LOAD. SQL\*Loader “conventional path” mode is similar in functionality to DB2 IMPORT.

As DB2 migration experts, we have experienced over the years that most Oracle DBAs use the conventional path mode of SQL\*Loader most of the time and most of their expertise and scripts are conventional path oriented. In fact, some Oracle DBAs never use the direct path mode of the SQL\*Loader at all. However when they begin to learn DB2, they often opt for using the DB2 LOAD utility (probably because of its name) and struggle using it because it has many of the characteristics of SQL\*Loader direct path mode that they were not using before. So, we are clarifying here that even though most Oracle DBAs use conventional path load options most of the time, we will still demonstrate converting all SQL\*Loader scripts, regardless of the path, to DB2 LOAD utility scripts. We think this gives the maximum performance out of DB2 right off the bat. If for any reason a DBA wants to change these scripts to use IMPORT instead of LOAD, they can do so later if their situation warrants it.

**Note:** It is not practical to try to exhaustively compare each and every DB2 LOAD, and for that matter IMPORT, feature (keyword with its options) with each and every Oracle SQL\*Loader feature (keyword with its options.)

So, we will cover the major features that are typically encountered by the Oracle DBA who wants to quickly migrate their SQL\*Loader scripts to DB2 LOAD scripts.

# DB2 LOAD vs. Oracle SQL\*Loader

## A Technical Overview

### The SQL\*Loader command line – invoking SQLLDR

The Oracle SQL\*Loader utility is invoked with the binary SQLLDR and uses a command line syntax similar to that of the DB2 LOAD. The command line can contain many keywords that will indicate to the SQL\*Loader utility where messages will go to, where discarded records will go to and so on.

The SQLLDR command line also indicates the name of the “control file” which usually has the extension of .CTL. This control file can also indicate to the SQL\*Loader utility where messages will go to, where discarded records will go to and so on. A SQLLDR command line reference to the same keyword will override the control file reference of the same functionality, so you have to pay close attention to both the SQLLDR command line and the control file in order to know how the SQL\*Loader session will actually work. This was probably designed to provide flexibility and power for the SQL\*Loader utility, but for those of us migrating these scripts to DB2, they can be confusing to unravel if some of the same keywords are being used in both places and the use of these are not consistent from script to script.

The SQL\*Loader control file is the place where the heart and detail of the load functionality is described and so when we describe a comparison of SQL\*Loader vs. DB2 LOAD we are for the most part comparing the SQLLDR control file vs. the DB2 LOAD command line. First though, we will cover all the options of the SQLLDR command line and compare them to the DB2 LOAD command line to see similarities with these. Then we will cover the control file and its keywords and options to see how these also compare to the DB2 LOAD command line.

### Oracle **SQLLDR** Command Line Syntax Example:

```
SQLLDR CONTROL=sample.ctl,  
      DATA=sample.dat,  
      LOG=sample.log,  
      BAD=sample.bad,  
      DISCARD=sample.dsc,  
      USERID=scott/tiger,  
      ERRORS=999,  
      LOAD=2000,  
      DISCARDMAX=5
```

### Oracle **SQLLDR** Command Line Keyword comparison to DB2 LOAD Keywords

# DB2 LOAD vs. Oracle SQL\*Loader

## A Technical Overview

<i>Oracle SQL*Loader Keyword</i>	<i>Oracle SQL*Loader Keyword Description</i>	<i>DB2 LOAD Keyword</i>	<i>DB2 LOAD Keyword Description</i>
<b>IMPORTANT KEYWORDS</b>			
CONTROL=filename.ctl	File with detailed LOAD command options.	N/A	DB2 LOAD command options are not separately called from a control file. The DB2 LOAD command line contains all the keywords in one invocation.
1. DIRECT=true 2. DIRECT=false	1. Invoke DIRECT PATH mode of the Oracle SQL*Loader utility. 2. If not used or valued at "false" then the CONVENTIONAL PATH mode of the Oracle SQL*Loader utility is invoked.	1. LOAD 2. IMPORT	1. DB2 LOAD utility itself is the near equivalent to the Oracle Direct Path mode of the SQL*Loader utility. 2. DB2 IMPORT is the near equivalent to the Oracle Conventional Path mode of the SQL*Loader utility.
<b>OTHER KEYWORDS</b>			
BAD=filename.bad	Where to store rejected records.	MODIFIED BY DUMPFILE=filename	DB2 LOAD modifier used to determine where rejected records are written.
DATA=filename.dat	Input data source file.	FROM sourcename	DB2 LOAD sourcename can be a file, pipe, device or cursor.
DISCARD=filename.dsc	Exception records not loaded for a variety of reasons.	FOR EXCEPTION tablename	DB2 LOAD puts records that violate unique index rules (exceptions) into a previously created table.
1. DISCARDMAX=number 2. ERRORS=number	1. Defines maximum amount of discard records before SQL*Loader terminates. 2. Defines maximum amount of errors before SQL*Loader terminates.	1. WARNINGCOUNT=number 2. NOROWWARNINGS	1. DB2 LOAD terminates on this many warnings. Discards are just one type of warning. 2. Modifier NOROWWARNINGS can turn off row warnings, but still leave a warning for exception records.
LOAD=number	Number of records to be loaded. (ALL=default)	ROWCOUNT number	Specifies number of records to be loaded. When omitted, the default is all.
MULTITHREADING=true	Allows for stream building on the client side and stream loading on the server side.	1. CPU_PARALLELISM number 2. DISK_PARALLELISM number 3. FETCH_PARALLELISM yes	DB2 LOAD determines these for itself using autonomics to control the number or threads spawned for parsing, converting, formatting and writing records in file, device, pipe and cursor loads. These can be specified if desired with these three keywords.
ROWS=number	Rows per data save.	SAVECOUNT number	DB2 LOAD uses consistency points for recoverability of the load operation.

# DB2 LOAD vs. Oracle SQL\*Loader

## A Technical Overview

<b>Oracle SQL*Loader Keyword</b>	<b>Oracle SQL*Loader Keyword Description</b>	<b>DB2 LOAD Keyword</b>	<b>DB2 LOAD Keyword Description</b>
1. LOG=logfile 2. SILENT=options	1. LOG stores output from load operations. 2. SELECT=options can suppress message output for various portions of that operation.	1. MESSAGES messagefile 2. NOROWWARNINGS	1. DB2 puts messages to this message file. It suppresses messages if you do not specify a message file. 2. Modifier NOROWWARNINGS turns off portions of the load operation message output.
SKIP=number	Start load after n records. Usually used to restart a load operation that committed a partial load but did not complete.  Note: If using this feature, SQL*Loader requires the operators to determine this for themselves and picking the wrong number could mean lost or duplicate data.	1. RESTART 2. (REPLACE, INSERT, TERMINATE)	1. One of the modes that DB2 LOAD uses to pick up where it left off after the last consistency point it took before failure. DB2 LOAD determines for itself where to pick up and does not require operators to figure it out. 2. The other modes DB2 LOAD can execute under are REPLACE, INSERT and TERMINATE, but these do not correlate to the SKIP keyword in any way.
1. SKIP_INDEX_MAINTENANCE=true 2. SKIP_UNUSABLE_INDEX=true	1. Stops index maintenance and marks indexes as unusable. 2. Skips index maintenance on any indexes already marked unusable.	1. INDEXING MODE DEFERRED 2. INDEXING MODE REBUILD, INCREMENTAL, AUTOSELECT	1. DB2 LOAD can defer index refresh to occur later during access to the data or during database activation. 2. DB2 LOAD can also specify INDEXING MODE REBUILD, INCREMENTAL or AUTOSELECT which can specify how the LOAD will perform index maintenance.
READSIZE=number	Size of external data file read before a commit is required.	DATA BUFFER number	DB2 LOAD uses this many 4K pages to transfer data within the utility which is normally determined intelligently through autonomics, but can be specified if desired with this keyword.
USERID/PASSWORD	Userid that connects to the database.	CONNECT TO...	DB2 uses a connect command prior to any subsequent LOAD commands.

# DB2 LOAD vs. Oracle SQL\*Loader

## A Technical Overview

### The SQL\*Loader Control File – The Heart of SQL\*Loader Utility

Although the Oracle SQL\*Loader command line can contain many keywords to control how the utility functions, our experience is that most of these keywords are usually indicated through the control file instead of the command line. Let's explore an example of a typical SQL\*Loader Control File converted to a DB2 LOAD command line.

### DB2 LOAD Scripts

#### Converted From Typical Oracle SQL\*Loader Control Files

DB2 LOAD Command File Example INSERT Fixed Data	Oracle SQL*Loader Control File Example INSERT Fixed Data
<pre> (1) LOAD (2) FROM 'INPUT_FILE1.DAT' (3) OF ASC (4) MODIFIED BY DUMPFILE='INPUT_FILE1.BAD' (5) METHOD L (1 5, 6 15, 16 20) (6) INSERT INTO PROD.TB_TABLE1 (7) ( COL1,       COL2,       COL3 ) (8) FOR EXCEPTION PROD.TB_TABLE1_DSC ;</pre>	<pre> (1) LOAD DATA (2) INFILE 'INPUT_FILE1.DAT' (4) BADFILE 'INPUT_FILE1.BAD' (8) DISCARDFILE 'INPUT_FILE1.DSC' (6) APPEND INTO TABLE PROD.TB_TABLE1 (5) (7) ( COL1 POSITION(01:05),           COL2 POSITION(06:15),           COL3 POSITION(16:20) ) ;</pre>

DB2 LOAD Command File Example REPLACE Variable Data	Oracle SQL*Loader Control File Example REPLACE Variable Data
<pre> (1) LOAD (2) FROM 'INPUT_FILE2.DAT' (3) OF DEL (4) MODIFIED BY DUMPFILE='INPUT_FILE2.BAD'       (3) COLDEL        (3) CHARDEL" (5) METHOD P (1 5, 6 15, 16 20) (6) REPLACE INTO PROD.TB_TABLE2 (7) ( COL1,       COL2,       COL3 ) (8) FOR EXCEPTION PROD.TB_TABLE2_DSC ;</pre>	<pre> (1) LOAD DATA (2) INFILE 'INPUT_FILE2.DAT' (4) BADFILE 'INPUT_FILE2.BAD' (8) DISCARDFILE 'INPUT_FILE2.DSC' (6) REPLACE INTO TABLE PROD.TB_TABLE2 (3) FIELDS TERMINATED BY ' ' (3) OPTIONALLY ENCLOSED BY '"' (5) (7) ( COL1,           COL2,           COL3 ) ;</pre>

# DB2 LOAD vs. Oracle SQL\*Loader

## A Technical Overview

Here is a description of the load script sections as compared in the above examples:

### (1) **LOAD**

This invokes the LOAD utility in DB2 or you can use IMPORT to invoke that utility.

In Oracle, LOAD DATA is used to invoke the SQL\*Loader utility. To specify a direct path load you must say DIRECT=true. The default is DIRECT=false and is thus not shown in the example.

### (2) **FROM [inputfile\_name]**

This is the file name that contains the data to be loaded. DB2 LOAD can also load data from a pipe, cursor or device.

Oracle also names the input file or device, and is able to name data with a BEGIN ...END clause that functions as a cursor.

### (3) **OF ASC / DEL**

For DB2 LOAD, ASC means un-delimited ASCII data and that the data is positional. DEL means delimited ASCII and the data can be variable in length for each row. There are a number of modifiers for delimited data, but the two key ones are COLDEL which determines how columns are delimited from column to column, the default is a comma, and CHARDEL which determines how character data is delimited, the default is double quote.

Oracle has FIX (the default and thus not shown in the example) or VAR, but these keywords are rarely used. It is usually the use of other keywords and the insert column references that determine whether or not the data is fixed or variable. For example, keywords FIELDS TERMINATED BY and FIELDS ENCLOSED BY to function similarly as COLDEL and CHARDEL in DB2 LOAD, which are required for variable, delimited data.

### (4) **MODIFIED BY DUMPFILE=[dumpfile\_name]**

DB2 puts rejected records to this file.

Oracle uses the BADFILE keyword to accomplish the same thing.

### (5) **METHOD P (1,2,3)**

DB2 LOAD has three methods:

- METHOD L is for ASC data only, and this method tells the start and end of each column. It looks like this: METHOD L (start1 end1, start2 end2....)
- METHOD N is for IXF or cursor data and this names the columns in the source table that are being loaded. It looks like this: METHOD N (col1, col2, col4...)
- METHOD P is for DEL, IXF or cursor data and this give the positional number of the columns from the source data that are being loaded. It looks like this: METHOD P (1, 2, 4...)

SQL\*Loader can combine using a column name as well as position of the fields in one line as shown in the example.

# DB2 LOAD vs. Oracle SQL\*Loader

## A Technical Overview

### (6) INSERT / REPLACE INTO PROD.TABLEx

DB2 LOAD has four options here. The two that correspond to Oracle SQL\*Loader are INSERT and REPLACE. The other two DB2 LOAD options you can use are RESTART and TERMINATE. These are used when a DB2 LOAD does not complete for any reason.

SQL\*Loader also has INSERT, but this is only used for an empty table and since APPEND works like INSERT on an empty table, few Oracle DBAs use it. SQL\*Loader REPLACE works the same as DB2 LOAD REPLACE.

### (7) (COL1, COL2, COL3) Insert Column List

DB2 LOAD will use this list of columns to place the data into. If you omit the column list, DB2 LOAD will attempt to load the data from the first through the last column with the data as it is read and parsed from the first field through the last field.

SQL\*Loader can combine using a column name as well as position of the fields in one line as shown in the example. For variable length data, position is not given but instead is determined by delimiters.

### (8) FOR EXCEPTION [table\_name]

DB2 LOAD writes records that violate unique index rules (exceptions) to this previously created table.

SQL\*Loader uses DISCARDFILE for the same thing, only it is an OS file and not a DB2 table.

## SQLLDR Control File Keywords vs. DB2 LOAD Keywords

# DB2 LOAD vs. Oracle SQL\*Loader

## A Technical Overview

<b>Oracle Control File Keyword</b>	<b>Oracle Control File Keyword Description</b>	<b>DB2 LOAD Keyword</b>	<b>DB2 LOAD Keyword Description</b>
<b>OPTIONS CLAUSE</b>	<i>Invoked with "OPTIONS (KEYWORD=value)"</i>		
1. <code>DIRECT=true</code> 2. <code>DIRECT=false</code>	1. Invoke DIRECT PATH mode of the Oracle SQL*Loader utility. 2. If not used or valued at "false" then the CONVENTIONAL PATH mode of the Oracle SQL*Loader utility is invoked.	1. <code>LOAD</code> 2. <code>IMPORT</code>	1. DB2 LOAD utility itself is the near equivalent to the Oracle Direct Path mode of the SQL*Loader utility. 2. DB2 IMPORT is the near equivalent to the Oracle Conventional Path mode of the SQL*Loader utility.
<code>ERRORS=n</code>	Defines maximum amount of errors before SQLLDR terminates.	1. <code>WARNINGCOUNT=number</code> 2. <code>NOROWWARNINGS</code>	1. DB2 LOAD terminates on this many warnings. Discards are just one type of warning. 2. Modifier <code>NOROWWARNINGS</code> can turn off row warnings, but still leave a warning for exception records.
<code>LOAD=n</code>	Number of records to be loaded. (ALL=default)	<code>ROWCOUNT number</code>	Specifies number of records to be loaded. When omitted, the default is all.
<code>MULTITHREADING=true</code>	Allows for stream building on the client side and stream loading on the server side.	1. <code>CPU_PARALLELISM number</code> 2. <code>DISK_PARALLELISM number</code> 3. <code>FETCH_PARALLELISM yes</code>	DB2 LOAD determines these for itself using autonomics to control the number or threads spawned for parsing, converting, formatting and writing records in file, device, pipe and cursor loads. These can be specified if desired with these three keywords.
<code>READSIZE=n</code>	Size of external data file read before a commit is required.	<code>DATA BUFFER number</code>	DB2 LOAD uses this many 4K pages to transfer data within the utility which is normally determined intelligently through autonomics, but can be specified if desired with this keyword.
<code>ROWS=n</code>	Rows per data save.	<code>SAVECOUNT number</code>	DB2 LOAD uses consistency points for recoverability of the load operation.
<code>SILENT=[options]</code>	<code>SELECT=options</code> can suppress message output for various portions of that operation.	1. <code>MESSAGES messagefile</code> 2. <code>NOROWWARNINGS</code>	1. DB2 puts messages to this message file. It suppresses messages if you do not specify a message file. 2. Modifier <code>NOROWWARNINGS</code> turns off portions of the load operation message output.



# DB2 LOAD vs. Oracle SQL\*Loader

## A Technical Overview

Oracle Control File Keyword	Oracle Control File Keyword Description	DB2 LOAD Keyword	DB2 LOAD Keyword Description
SKIP=n	Start load after n records. Usually used to restart a load operation that committed a partial load but did not complete. Note: If using this feature, SQL*Loader requires the operators to determine this for themselves and picking the wrong number could mean lost or duplicate data.	1. RESTART 2. (REPLACE, INSERT, TERMINATE)	1. One of the modes that DB2 LOAD uses to pick up where it left off after the last consistency point it took before failure. DB2 LOAD determines for itself where to pick up and does not require operators to figure it out. 2. The other modes DB2 LOAD can execute under are REPLACE, INSERT and TERMINATE, but these do not correlate to the SKIP keyword in any way.
1. SKIP_INDEX_MAINTENANCE=true 2. SKIP_UNUSABLE_INDEX=true	1. Stops index maintenance and marks indexes as unusable. 2. Skips index maintenance on any indexes already marked unusable.	1. INDEXING MODE DEFERRED 2. INDEXING MODE REBUILD, INCREMENTAL, AUTOSELECT	1. DB2 LOAD can defer index refresh to occur later during access to the data or during database activation. 2. DB2 LOAD can also specify INDEXING MODE REBUILD, INCREMENTAL or AUTOSELECT which can specify how the LOAD will perform index maintenance.
CONTROL FILE MAIN KEYWORDS			
NOLOGGING	This option allows logging to be bypassed, but makes the table unrecoverable with a roll forward operation.	NONRECOVERABLE	With this option, table spaces are not put in backup pending state following the load operation, and a copy of the loaded data does not have to be made during the load operation.
CONTINUE_LOAD DATA	Starts a terminated load by automatically finding the proper place to continue. (direct path only)	RESTART	DB2 LOAD uses the last consistency point it took before failure to pick up where it left off.
LOAD DATA	Invoke SQLLDR binary to load data with any mode (or path).	1. LOAD 2. IMPORT	1. DB2 LOAD utility itself is the near equivalent to the Oracle Direct Path mode of the SQL*Loader utility. 2. DB2 IMPORT is the near equivalent to the Oracle Conventional Path mode of the SQL*Loader utility.
INFILE filename	Input data source file.	FROM sourcename	DB2 LOAD sourcename can be a file, pipe, device or cursor.
RECSIZE n	Size of fixed input record.	MODIFIED BY RECLLEN=x	Size of fixed input record.

# DB2 LOAD vs. Oracle SQL\*Loader

## A Technical Overview

<b>Oracle Control File Keyword</b>	<b>Oracle Control File Keyword Description</b>	<b>DB2 LOAD Keyword</b>	<b>DB2 LOAD Keyword Description</b>
BADFILE filename	Where to store rejected records.	MODIFIED BY DUMPFILE= filename	DB2 LOAD modifier used to determine where rejected records are written.
DISCARDFILE filename	Exception records not loaded for a variety of reasons.	FOR EXCEPTION tablename	DB2 LOAD puts records that violate unique index rules (exceptions) into a previously created table.
DISCARD	Exception records not loaded for a variety of reasons.	FOR EXCEPTION tablename	DB2 LOAD puts records that violate unique index rules (exceptions) into a previously created table.
DISCARDMAX	Defines maximum amount of discard records before SQLDR terminates.	1. WARNINGCOUNT= number 2. NOROWWARNINGS	1. DB2 LOAD terminates on this many warnings. Discards are just one type of warning. 2. Modifier NOROWWARNINGS can turn off row warnings, but still leave a warning for exception records.
1. VAR n 2. FIX n	Expects data to be variable or fixed in formatting. If VAR is used, then n is the number of bytes at the beginning of the row that declares how long each row is.	1. OF DEL 2. OF ASC	DB2 variable data is delimited and fixed data is ASCII.
1. INSERT or APPEND 2. REPLACE	1. INSERT and APPEND are similar in that they add to data already existing, except INSERT expects the table to be empty. 2. REPLACE truncates any data already in the table before adding the new data.	1. INSERT 2. REPLACE	1. DB2 LOAD INSERT adds to the data in the table, even an empty one. 2. REPLACE truncates any data already in the table before adding the new data.
INTO TABLE tablename	Table the data is going into.	INTO TABLE tablename	Table the data is going into.
TERMINATED BY string	Data is read until the first occurrence of this string.	MODIFIED BY COLDELx	All input data is delimited from column to column by this character. (Default is a comma.)
ENCLOSED BY string	Data is optionally enclosed by this string; usually this is for character data.	MODIFIED BY CHARDELx	Input character data is enclosed with this character. (Default is double quote.)
LOBFIL(filename)	This is designated at the INSERT column list and loads LOBs from external file name source.  The data itself contains the filename with a full path name other wise LOBs are searched in the same directory as the LOAD script.	1. LOBS FROM pathnames 2. MODIFIED BY LOBSINFILE	1. Pathnames where LOB files will be found. 2. This must be coded to activate the LOBS FROM clause.  The data itself contains the filename without a full path name as the LOBS FROM path will be searched.

# DB2 LOAD vs. Oracle SQL\*Loader

## A Technical Overview

### Part II –SQL\*Loader to DB2 LOAD PERL Script Converter

XXX

#### Setting up the example:

XXX

#### Conclusion:

It is relatively easy for Oracle DBAs to take their knowledge of the SQL\*Loader and apply it to learning the DB2 LOAD and IMPORT utilities. Our focus in this article was specifically on LOAD, but much of this information applies to IMPORT as well. We have shown you the main keywords to both SQL\*Loader and DB2 LOAD and how they compare and we have provided an easy to use PERL script that can convert most of the SQL\*Loader scripts you may have. This should make things much easier for you in your adoption of using DB2 in your organization.

#### About the authors:



Burt Vialpando, an IBM employee since 1998, is a certified IT specialist currently working for the SMPO team performing presales Oracle to DB2 migration support. With professional IT experience since 1984, he holds numerous DB2, Oracle, and other certifications as well as multiple IBM patents. He currently serves on the Certification Board, the Competency Team and the Migrations Committee.



Vikram Khatri works for IBM in the Sales and Distribution Division and is a member of DB2 Migration team. Vikram has 19 years of IT experience and specializes in migrating non-DB2 databases to DB2. Vikram supports the DB2 technical sales organization by assisting with complex database migration projects as well as with database performance benchmark testing.