

Introduction to MultiLoad

Using a single MultiLoad job, you can do a number of different import, update and delete tasks on RDBMS tables and views. Each MultiLoad import task can do multiple data insert, update and delete functions on up to five different tables or views. It processes a series of MultiLoad commands and Teradata SQL statements you enter, usually as a batch mode job script. The MultiLoad commands provide the session control and data handling specifications for the data load operations. The Teradata SQL statements perform the actual data load functions on the Teradata RDBMS tables and views. MultiLoad can work on Unix, Windows 95/2000/NT/XP where client systems are connected to Teradata RDBMS through network. MultiLoad runs in two modes : Interactive and Batch.

If your MultiLoad job just has only one import task, then you can use FastLoad which finish it much quicker. But FastLoad utility works only on empty tables. You can not use the FastLoad to

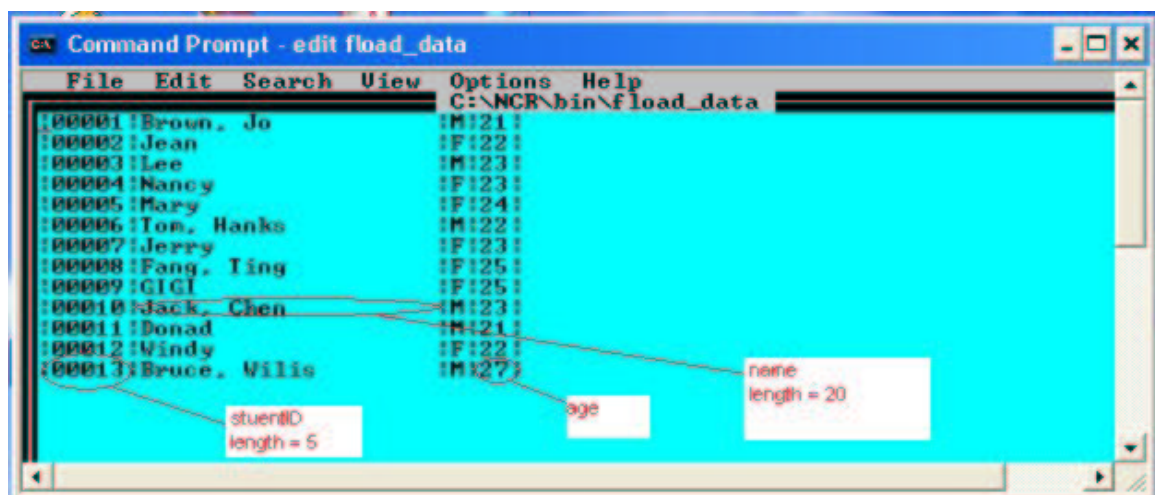
- Insert additional data rows into an existing table
- Update individual rows of existing tables
- Delete individual rows from existing tables
- Load data into multiple tables

MultiLoad Data Source

MultiLoad command IMPORT has an option FORMAT which is used to specify the format of extern data source. The format may be

- FASTLOAD
- BINARY
- TEXT
- UNFORMAT
- VARTEXT

The following example (also used as the example of FastLoad) shows an unformatted data source file created by MS DOS command **edit**.



Although the field age should be an integer data type, we just write it as ASCII string. The MultiLoad will convert it to appropriate data type according to the definition of the destination table. The MultiLoad can convert data as the following rules

- o Numeric – to – numeric
- o Character – to – numeric
- o Character – to – date
- o Date – to – character

Using MultiLoad

In interactive mode, MultiLoad use terminal screen and keyboard as the standard output and input streams. In batch mode, you can use > and < redirect the standard output / input streams. If you want to invoke MultiLoad in interactive mode, use command :

```
c:\ncr\bin\MultiLoad
```

You can invoke it in batch mode by using command

```
c:\ncr\bin\MultiLoad [options] < infile > outfile
```

Here, infile is a MultiLoad job script file which includes all MultiLoad commands and SQL statements. The outfile is the output stream file.

```
-b                /* only print the brief info which can be
                  used to determine success or failure. */

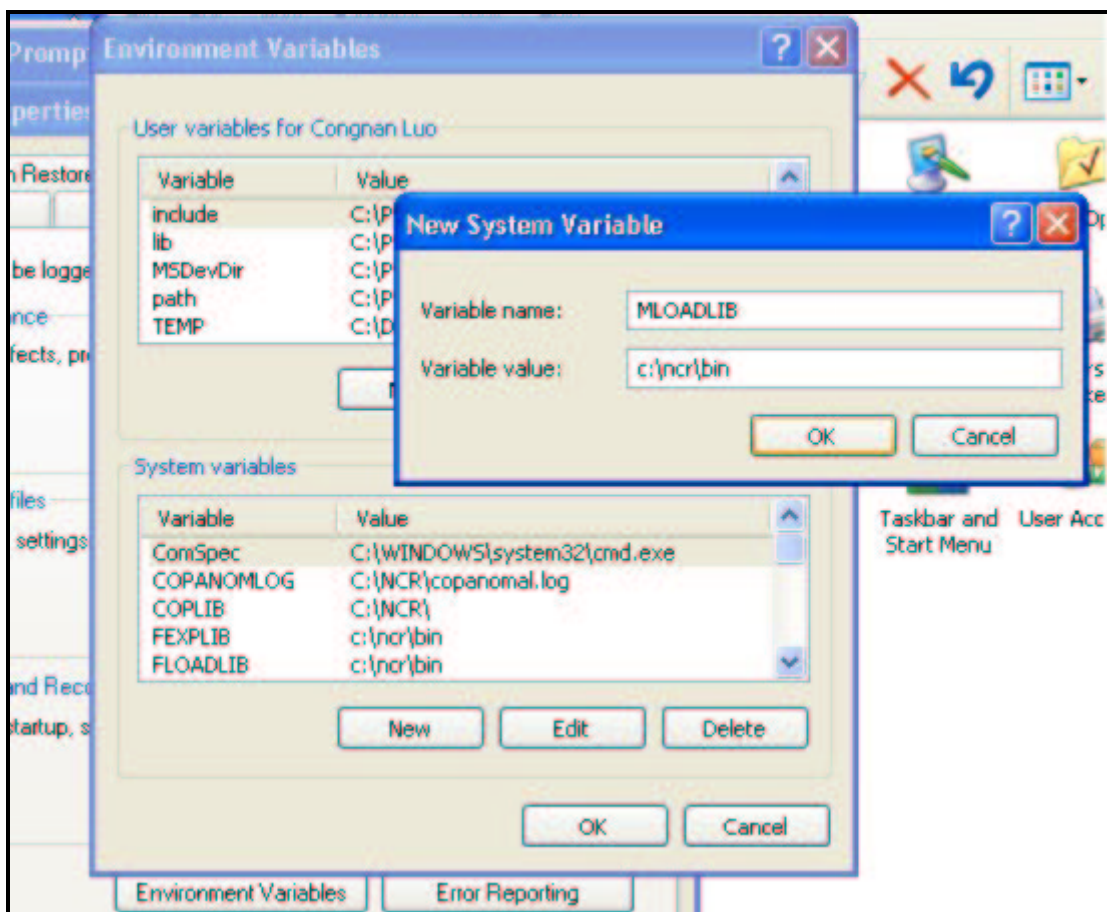
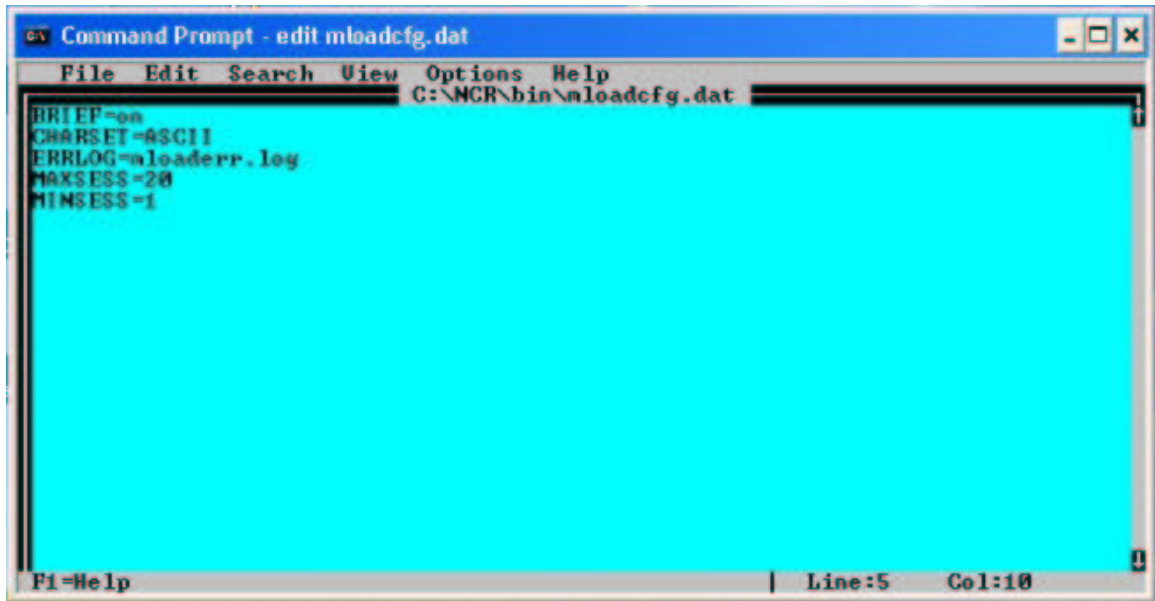
-c charset_name   /* the name can be ASCII ( 255 ) and
-c charset_code   KANJISJIS_0S ( 119 ), SCHGB2312_1T0
                  (121 ) */

-e filename       /* specifies a file for error messages */

-r 'MultiLoad Command' /* the MultiLoad Command may be RUN
                       FILE, so you can run a job script */

-M               /* maximal number of sessions */
-N               /* minimal number of sessions */
```

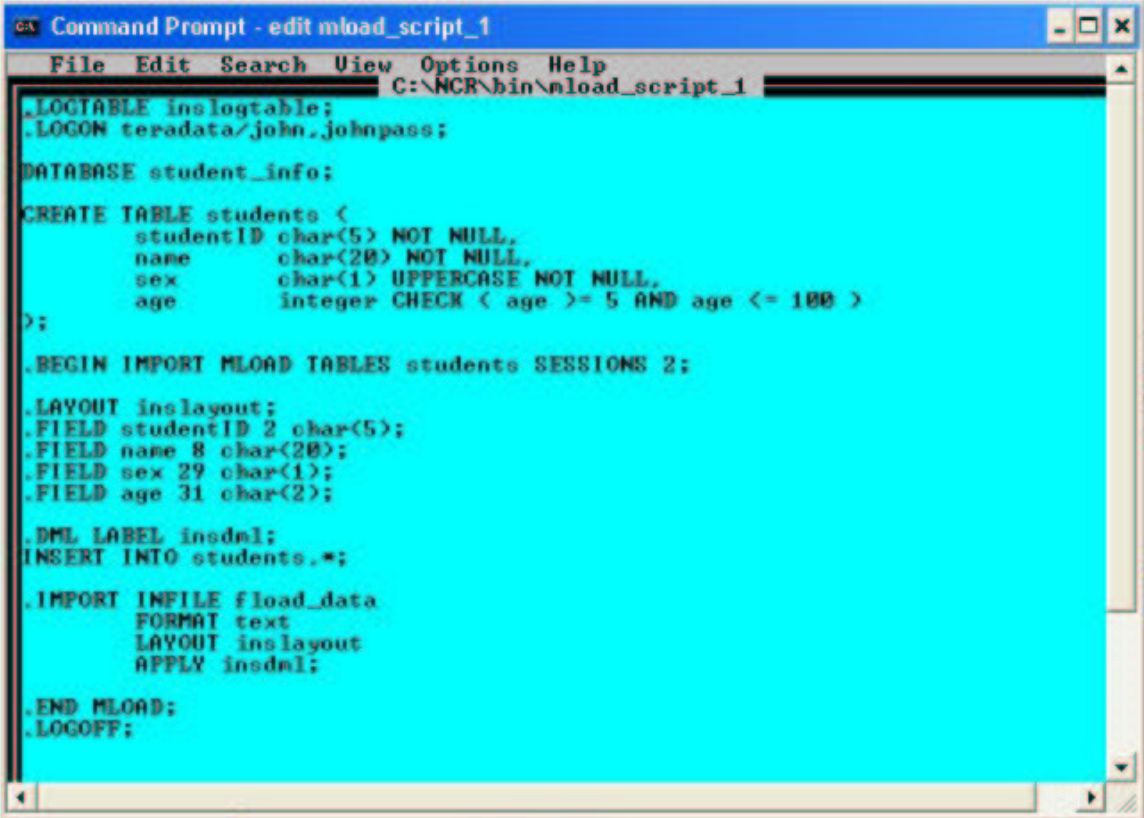
When MultiLoad is executed, it will check whether you specify the above options with **MultiLoad** command. If not, it will check whether some option parameters are specified in job scripts or the MultiLoad configuration file. If not, it will use the default setting. So you can write a configuration file to specify these options, the name of the file must be **mloadcfg.dat** and it must be placed under the current directory for your MultiLoad utility software. Or you need to use an environment variable **MLOADLIB** to specify the directory.



In network-attached environment, if you want to terminate the MultiLoad, please execute LOGOFF/QUIT command and press CTRL+C. When your MultiLoad job is aborted because of some errors, maybe the destination table and two error tables are already created in your database. If you want to re-execute the MultiLoad job, you must drop them first. If your MultiLoad job is finished successfully, then MultiLoad will drop two error tables automatically, you don't need to take care of them.

MultiLoad Example

In this example, we will load the data from the file, **fload_data**, into the table, **students**, in the database **student_info** whose owner is user **john**. We need to edit the MultiLoad job script file, **mload_script**, which is under the directory where MultiLoad works on. Before executing this MultiLoad job, we first drop the table **students** in database **student_info**. Then we invoke MultiLoad utility to finish this job.



```
Command Prompt - edit mload_script_1
File Edit Search View Options Help
C:\NCR\bin\mload_script_1
.LOGTABLE inslogtable;
.LOGON teradata/john.johnpass;
DATABASE student_info;
CREATE TABLE students (
  studentID char(5) NOT NULL,
  name      char(20) NOT NULL,
  sex       char(1) UPPERCASE NOT NULL,
  age       integer CHECK ( age >= 5 AND age <= 100 )
);
.BEGIN IMPORT MLOAD TABLES students SESSIONS 2;
.LAYOUT inslayout;
.FIELD studentID 2 char(5);
.FIELD name 8 char(20);
.FIELD sex 29 char(1);
.FIELD age 31 char(2);
.DML LABEL insdml;
INSERT INTO students.*;
.IMPORT INFILE fload_data
  FORMAT text
  LAYOUT inslayout
  APPLY insdml;
.END MLOAD;
.LOGOFF;
```

```
ca Command Prompt
C:\NCR\bin>mload mload_script_1
**** 14:13:41 UT12416 Config file parameters in effect are: BRIEF.
**** 14:13:41 UT12416 Config file parameters in effect are: ASCII.
**** 14:13:41 UT12416 Config file parameters in effect are: MAXSESS=20.
**** 14:13:41 UT12416 Config file parameters in effect are: MINSESS=1.
**** 14:13:41 UT12413 Error Logging is enabled: mloaderr.log
=====
=          MultiLoad Utility          Release MLOD.07.05.00          =
=          Platform WIN32              =
=====
**** 14:13:41 UT12411 Processing start date: SAI MAR 23, 2002
=====
=          Logon/Connection            =
=====
0001 .LOGTABLE inslogtable;
0002 .LOGON teradata/john.;
**** 14:13:42 UT18601 Character set specified: ASCII
**** 14:13:42 UT18400 Maximum supported buffer size: 64K
**** 14:13:47 UT16211 A successful connect was made to the RDBMS
**** 14:13:47 UT16217 Logtable 'JOHN.inslogtable' has been created.
0003 DATABASE student_info;
**** 14:13:48 UT1016 'DATABASE' request successful.
0004 CREATE TABLE students (
      studentID char(5) NOT NULL,
      name       char(20) NOT NULL,
      sex        char(1) UPPERCASE NOT NULL,
      age        integer CHECK ( age >= 5 AND age <= 100 )
);
**** 14:13:50 UT1016 'CREATE' request successful.
0005 .BEGIN IMPORT MLOAD TABLES students SESSIONS 2;
=====
=          Processing MultiLoad Statements          =
=====
```

From the above window, we can see the status information MultiLoad prints on the screen. MultiLoad will create a log file `mloaderr.log` under the work directory, and then it create a `john.inslogtable` as the checkpoint table. If the job is paused because of unexpected factors, maybe you want to completely re-do the job, then you must delete the log file, drop the log table `inslogtable`, the target table `students`, the table `WT_students`, two error table `ET_students` and `UV_students`. Some of them are created by MultiLoad automatically during each parse. If your job is completely finished by MultiLoad, then it will drop some work tables automatically.

```

**** 14:36:34 UTY0815 MLOAD session(s) connected: 2.
=====
MultiLoad Acquisition Phase
=====
**** 14:36:44 UTY0826 A checkpoint has been taken, recording that end of file has
been reached
**** 14:36:44 UTY1803 Import processing statistics
- IMPORT 1 Total thus far
=====
Candidate records considered:..... 13..... 13
Apply conditions satisfied:..... 13..... 13
Candidate records not applied:..... 0..... 0
Candidate records rejected:..... 0..... 0
=====
MultiLoad Application Phase
=====
**** 14:36:48 UTY0818 Statistics for table students:
Inserts: 13
Updates: 0
Deletes: 0
=====
MultiLoad Task Cleanup
=====
**** 14:36:53 UTY0821 Error table EI_students is EMPTY, dropping table.
**** 14:36:56 UTY0821 Error table UU_students is EMPTY, dropping table.
**** 14:37:00 UTY0825 Error table statistics for:

Target table 1: students
Number of Rows Error Table Name
=====
0 EI_students
0 UU_students
=====
**** 14:37:02 UTY0822 MultiLoad processing complete for this MultiLoad import ta
sk.
=====
MultiLoad Task Complete
=====
**** 14:37:03 UTY1024 Session modal request, 'DATABASE', re-executed.
0015 .LOGOFF;
=====
Logoff/Disconnect
=====
**** 14:37:09 UTY6216 The restart log table has been dropped.
**** 14:37:09 UTY6212 A successful disconnect was made from the RDBMS.
**** 14:37:09 UTY2410 Total processor time used = '0.046875 Seconds'
- Start : 14:36:23 - SAT MAR 23, 2002
- End : 14:37:09 - SAT MAR 23, 2002
- Highest return code encountered = '0'.
C:\NCR\bin>_

```

We can logon Teradata RDBMS through BTEQ to check the result.

```

c:\ Command Prompt - bteq
*** Query completed. 13 rows found. 4 columns returned.
*** Total elapsed time was 1 second.

studentID  name                sex    age
-----
00009     GIGI                F      25
00007     Jerry               F      23
00005     Mary                F      24
00013     Bruce, Wilis       M      27
00002     Jean                F      22
00003     Lee                 M      23
00011     Donad               M      21
00012     Windy               F      22
00001     Brown, Jo          M      21
00006     Tom, Hanks         M      22
00010     Jack, Chen         M      23
00008     Fang, Ting         F      25
00004     Nancy               F      23

BTEQ -- Enter your DBC/SQL request or BTEQ command:

```