Convert TLC Legacy to TLC32 Database

1st make sure the TLC Legacy data files are as error free as they can be. Run TLCFIX on each of the Data files. The following is an excerpt of TLCFIX from the Utility Section of the TLC Manual:
TLCFIX - calls a program utility named filefix. The basic command formats are shown below.
Note: When running this and other TLC utilities against different TLC Databases it will help to insure the TLCDATA environment is set for your data directory. Example: if the Data Files you are checking/loading/unloading are in directory h:\mfg\tools\cribdata then the command SET TLCDATA=h:\mfg\tools\cribdata should be in effect. If you are not sure, run the command SET from the DOS prompt to see the environment settings. If you are using TLC.CFG under Revision 2.7.2 change the TLCDATA environment variable in the TLC.CFG file.
Purpose: To check and repair TLC database files. It should be used whenever a system crash occurs, i.e., a power failure, or when you receive faulty information or suspect the integrity of the TLC database files has been compromised. This utility compares the *.idx* file to the *.dat* file for consistency. If an inconsistency exists the utility will ask you if you wish to delete or rebuild the corrupted indexes.

Basic Command Format: filefix [-n] [-y] [filename]

Command Format Explanation:

-n if present, automatically answers NO to all questions.

-y if present, automatically answers YES to all questions. Do not run TLCFIX with the -y option if checking the file for the first time. If you do not use -n or -y, TLCFIX will be interactive and will wait for your response to each error condition it finds.

filename is the name of the TLC database file you wish to check.

Use the following chart to specify the DATA filename.

TOOLID	KITID	KITDEF	GAGEID	ONORDID
VENDID	USERID	MACHID	JOBID	REFFID
ALIAS	ACTIVINV	REWRKINV	RWRKHIST	
SCRAPINV	RESRV	GAGECH	PRESET2	

EXAMPLE: filefix -n \data\toolid
Will check the toolid .idx and .dat files for inconsistencies and answer NO to all questions.

UNIX EXAMPLE: (using UNIX/XENIX command style)

bcheck -n /usr/data/toolid

Helpful hint: Always write any system report errors down for reference in case correction techniques do not work as expected and communication of errors is needed by ITC personnel.

TLCFIX asks the questions necessary to execute the filefix command string.

2nd dump all the files from the TLC Legacy database to text files using the TLCDUMP Utility. The following is an excerpt of TLCDUMP from the Utility Section of the TLC Manual:

TLC DUMP calls a utility program named **FILEDUMP** and asks the questions necessary to complete the command string to run the utility Filedump.

Program File Name(s): fileload - Unix

FILEDUMP.EXE - Windows or DOS

TLCDUMP.EXE - Windows or DOS Executable sends command format to FILEDUMP

Purpose: To allow an ASCII file to be created from any C-ISAM file contained in the TLC System Database. The ASCII file can be used as a conduit for passing information to other software application programs or transmission to other computer systems. Once the ASCII file is built, its data integrity becomes the customer's responsibility.

Basic Command Format: filedump [-n] [-q] [-d?] [-fn] [filename]

Command Format Explanation:

-n if present, no quote characters (") are placed around alpha fields; this is the default condition.

-q if present, quotes (") are placed around all alpha fields (ala dBase like files), but not around numeric fields.

-r if present, the output replaces the specified file; the default is to append the output to the specified file.

-0 (zero) if present, the numeric fields in the output will have leading zeros so that all fields will have fixed length.

-d? if present, ? is the character used as a delimiter between fields; the vertical bar (|) is the default delimiter.

-u if present, shared access to the file - UNIX only.

-fn will specify the TLC Database file to dump from. Use the following chart for specifying **n** (the TLC \DATA file).

```
1 = TOOLID    2 = KITID    3 = KITDEF    4 = GAGEID   5 = ONORDID
6 = VENDID    7 = USERID   8 = MACHID    9 = JOBID    a = REFFID
b = ALIAS     c = ACTIVINV d = REWRKINV  e = RWRKHIST
f = SCRAPINV  g = RESRV    h = GAGECH    i = PRESET2
```

filename is the name of the ASCII text file to output the records to; default is standard output (so that the output can be used in a pipeline and can be the input to another command).

MS-DOS EXAMPLE: filedump -q -d, -f2 kitid.txt

Dumps the KITID file, in ASCII, to a file named "kitid.txt" with "marks around ALPHA's and comma's as field delimiters.

This example will dump the On Order file (5), pipe the output into a sort command, which uses the "|" as a field delimiter and sorts on field number 3 (vendor) and pipe the output into a grep command which will discard all records with "D" in the first character and redirect the resulting output to a file called "vendord.txt".

Sample Output of FILEDUMP:

MS-DOS EXAMPLE: filedump -q -d, -f2 kitid.txt

"10030099 ","NO.39 .099D DRILL TOOL HOLDER ASSY ","STANDARD NO. 39 .099D DRILL ASSEMBLY .USED ON 7-1098-4 OPER. 50 .S4000 F4 OFFSET XX .GLZ STANDARD STANDARD EXTENSION LENGTH 1.55 MIN FL",5,
"1234567890-10 ","BEARING MOUNTING BRACKET OP 10 REV.C","THIS KIT CALL KITS THAT CALL KITS (MULTILEVELNEST).IT ALSO CALL FIXTURES AND TAPES NOT DEFINED TO THE.TOOL DATABASE A NICE FEATURE OF TLC'S KITTING .NOTE:RUN AVAILABITY 1ST OR CALL J. DOAKS XT. 6659 ",9,
"18011000 ","1.0" INSERT DRILL TOOL ASSEMBLY ","STANDARD ASSEMBLY USED ON 7- 1098-4 OPER. 50 .S3000 F6 OFFSET XX GLZ 9.000 5.000 .EXTENSION FROM HOLDER ",5,
"19010047 ","NO.1 CENTER DRILL TOOL HOLDER ASSY ","STANDARD NO. 1 CENTER DRILL ASSEMBLY .USED ON 7-1098-4 OPER. 50 .S1000 F10 OFFSET XX GLZ 8.250 4.000 .EXTENSION FROM HOLDER 1.5 CLEARANCE ",5,

3rd after dumping all the TLC Legacy data files from the TLC database you should view them with a text viewer to quickly verify the file integrity. If you find blocks of data that do not look correct you can edit/delete information here. Be sure to maintain the data format and sequence defined in the TLC DATA FILE FORMATS section of the Utilities Manual.
Another option is to export the text file for each data file to a spreadsheet program, e.g., Excel, and edit delete your tool information in the spreadsheet. If you export to another file format you must convert back to the text format before loading to the new TLC Database.

4th the TLC 32 database should be empty. If it is not and you wish to create an empty database 1) copy the empty database from the TLC 32 CD-ROM or 2) use the undocumented utility called REINIT.EXE. You can find it on the CD-ROM that came with your system. It will reinitialize a database's TLC files to zero length files. Again be sure your TLCDATA environment variable is SET to the correct directory before running this utility. It will ask you twice if you wish to reset the data file. The other option to running this utility is to copy the data directory that came with your original system. It should be an empty data directory with zero length CISAM files.
You must zero the data files or start with an **empty** data file before loading the new database. If you are working on one database you may want to have it copied or backed up before running this utility. If you are creating another database you may want to copy or rename an existing database and REINIT it. In the end the data in the data files of the database will be destroyed.

5th load all the new text files into the new database using the TLC 32 TLCDUMP Utility (it does both the Dump & Load functions). The following is an excerpt of TLCLOAD from the Utility Section of the TLC Manual:
TLC LOAD calls a utility program named **FILELOAD** and asks the questions necessary to complete the command string to run the utility Fileload.

Program File Name(s): fileload - Unix
FILELOAD.EXE - Windows or DOS
TLCLOAD.EXE - Windows or DOS Executable sends command format to FILELOAD
Purpose: To allow a compatible ASCII formatted file to be loaded into any C-ISAM file contained in the TLC System Database. The ASCII file can be used as a conduit for passing information from other software application programs or transmission from other computer systems. Formatting of the ASCII file and its data integrity is the customer's responsibility.

Basic Command Format:
fileload [-n] [-q] [-x] [-r] [-d?] [-fn] [filename]

Command Format Explanation:

-n if present, no quote characters (") are expected around alpha fields; this is the default condition.

-q if present, quotes (") are expected around all alpha fields (ala dBase like files), but not around numeric fields.

-x if present, the file is not loaded, it is displayed for testing the format before actual loading.

-r if present, the records read from the input text file will replace any existing record in the tool database with the same primary index; the default condition is not to replace matching records, but to flag the records as duplicates.

-d? if present, ? is the character used as a delimiter between fields; the vertical bar (|) is the default delimiter.

-fn will specify the TLC \DATA file to dump from. Use the fol lowing chart for specifying n (the TLC \DATA file).

```
1 = TOOLID    2 = KITID    3 = KITDEF    4 = GAGEID   5 = ONORDID
6 = VENDID    7 = USERID   8 = MACHID    9 = JOBID    a = REFFID
b = ALIAS     c = ACTIVINV d = REWRKINV  e = RWRKHIST
f = SCRAPINV  g = RESRV    h = GAGECH    i = PRESET2
```

filename is the name of the ASCII text file to load or read the data records from; default is the standard input (thus, fileload can be used at the end of a pipeline and read the output of another command as its input)., ,

Sample Output:
Ready to load \data\activinv from activ.dbs - press ENTER to begin

```
Record:      25 ID: CST031037-006     Duplicate!
Record:      32 ID: TD8.6MM           Duplicate!
32 Records read, 30 Records added to file \data\activinv
```

There should not be any duplicate records in the loading of a new file in the database. This condition usually presents itself when editing and reloading the existing database. However, if combing 2 or more databases, duplicate records can exist, especially in the TOOLID file. You will have decide which text file is to take precedence over the other and select the correct switch for *replace or do not replace duplicates* when loading the additional TOOLID data.
TLC does not allow two tools with the same TOOLID. So when combining databases the information of two tools with the same name from two or more databases must be combined before the FILELOAD or you will have to edit the information after the load. Predominately the locations, quantities, prices will be the biggest considerations.
All other data files will probably have enough difference in the record that duplicates will not be a problem. For example, the ACTIVINV file although being combined from two or more databases will probably have multiples of the same tool number issued but because the tool cribs are in different areas serving different machines and users the records will be sufficient different to co-exist in the same database file.

6th repeat the FILEFIX (1st step) again. This will verify the data was loaded correctly and there aren't any serious problems.

7th test the database with the system. Perform a few withdraws returns, etc. to see if everything works with the new data files. Run a few reports to see that everything lines up.

8th run the utility named INUSEFIX.

Purpose: To check the validity of the in-use count of a tool in the *toolid* file to the number of occurrences of the tool in the *activinv* file. If the utility finds count errors it will attempt to fix the count in the *toolid* file.

Notes:
1. If combining multiple databases into one database repeat steps 1, 2, 3 for each database and save the text file under an identifiable name for that database set. Run step 5 for each of the text files from the databases.
2. If errors happen while trying to run the utilities check that the TLCDATA Directory is available.
3. Be sure to maintain the data format and sequence defined in the TLC DATA FILE FORMATS section of the Utilities Manual.