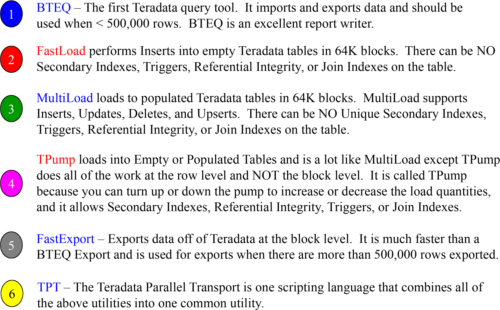
Chapter 20 - Teradata Load Utilities Introduction

“I don’t know who my grandfather was. I am more interested in who his grandson will become.”

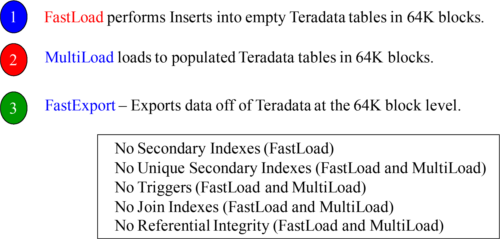
– Abraham Lincoln, President of the United States

TheTeradata Utilities



Above, are the utilities and their specialties.

BlockLevel Utilities

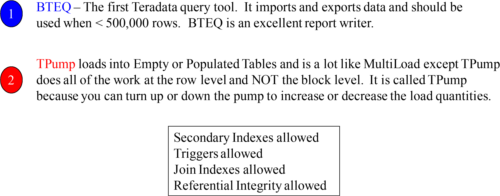


Only 30 FastLoad and MultiLoad combinations can run at one time.

Only 60 FastExport jobs can run simultaneously!

The block utilities have limits that prevent too many of them running at the same time. This feature is actually controlled by a new DBS Control parameter named MaxLoadAWT, which controls AMP Worker Tasks (AWT). When MaxLoadAWT is set to zero, then it is like going back in time to pre-V2R6.0 where only 15 FastLoad, MultiLoad, and FastExport jobs can run max.

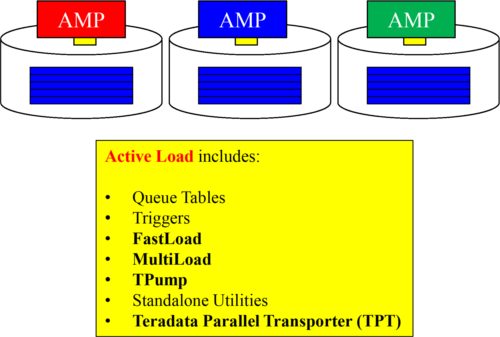
RowLevel Utilities



You can have an unlimited amount of TPump and BTEQ utilities running simultaneously!

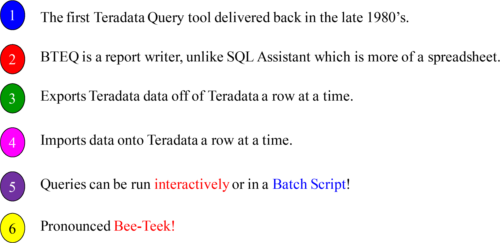
Row Level Utilities are not as fast as block level utilities, but they are really flexible and easy to implement. There are almost no restrictions.

TheActive Load Concept



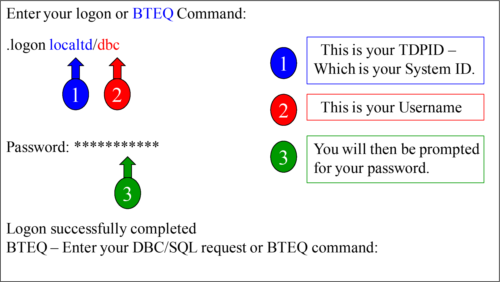
Active Load is delivered through methods that support continuous data loading. This was the original idea behind TPump. Now included are other streaming methods such as Queue tables, more frequent batch updates, and moving data from other platforms like SQL Server or Oracle directly to Teradata.

BTEQ– Batch Teradata Query Tool



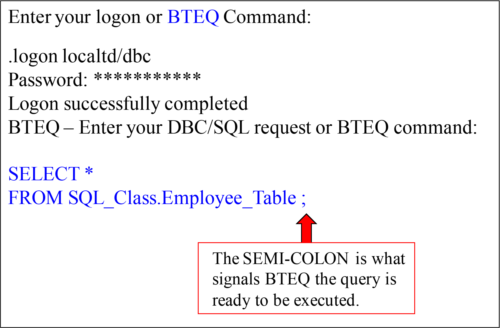
Why is BTEQ available on every Teradata system ever built? Because the Batch TEradata Query (BTEQ) tool was the original way that SQL was submitted to Teradata as a means of getting an answer set in a desired format. Here is what is excellent about BTEQ: It is a cool report writer and can do things SQL Assistant cannot do. It is also great at Export and Import of data at a row level. Although somewhat outdated, BTEQ can be a valuable asset (on a rainy day)!

Howto Logon to BTEQ in Interactive Mode



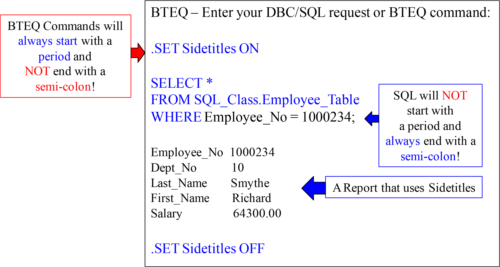
When logging onto BTEQ in interactive mode, you will type .LOGON followed by your TDP-ID and then a forward slash. The TDP-ID identifies your system. You could have multiple Teradata systems, such as a production system and a test system. The TDP is the name of the system you are logging onto. Then, you enter your User-ID. You will then be prompted for your password which is hidden from view as you type it in.

RunningQueries in BTEQ in Interactive Mode



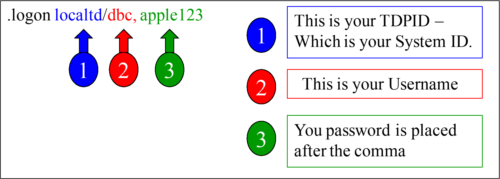
Remember that BTEQ commands begin with a period (.) and do not require a semi-colon (;) to end the statement. SQL commands do not ever start with a period, and they must always be terminated with a semi-colon. Remember, BTEQ Interactive Mode is **NOT** part of the **TPT utility**.

BTEQCommands vs BTEQ SQL Statements



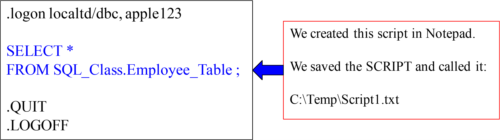
Remember that BTEQ commands begin with a period (.) and do not require a semi-colon (;) to end the statement. SQL commands do not ever start with a period, and they must always be terminated with a semi-colon.

Howto Logon to BTEQ in a SCRIPT



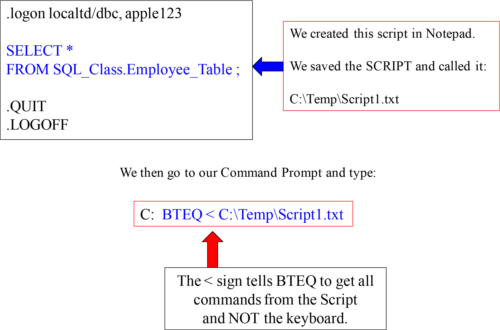
When logging onto BTEQ in a script, you will type .LOGON followed by your TDP-ID and then a forward slash. The TDP-ID identifies your system. You could have multiple Teradata systems such as a production system and a test system. The TDP is the name of the system you are logging onto. Then, you enter your User-ID. Then, you separate your password from your User-ID by a comma.

RunningQueries in BTEQ through a Batch Script



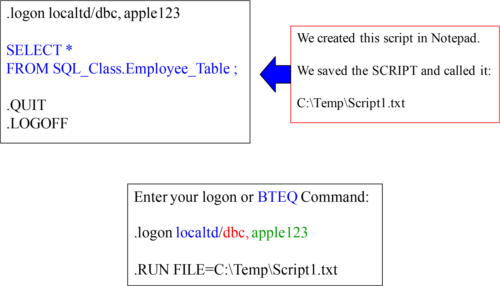
When you want to run your SQL or Import or Export jobs in a script, you create the script in something like notepad and save it. We saved the above script that we named Script1.txt in our C:\Temp directory on our local PC hard drive.

Runninga BTEQ Batch Script through the Command Prompt



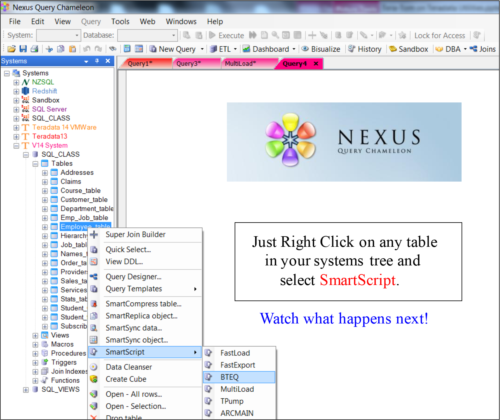
Once our script is created and saved, we can go to the command prompt. There, we invoke BTEQ, place a less than sign, and give it the script name using the full path.

Runninga BTEQ Batch Script through the Run Command



Once our script is created and saved, we can go logon to BTEQ. There, we put in the .RUN FILE command and give it the script name using the full path.

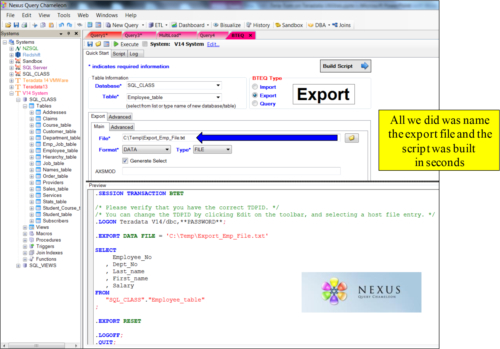
UsingNexus to Build Your BTEQ Scripts



The Nexus can build all of your traditional Teradata load utilities for you in seconds including BTEQ, FastLoad, MultiLoad, TPump, FastExport, and ARCMain.

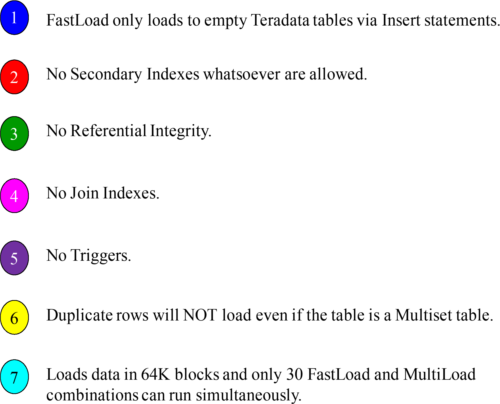
.

UsingNexus to Build Your BTEQ Scripts



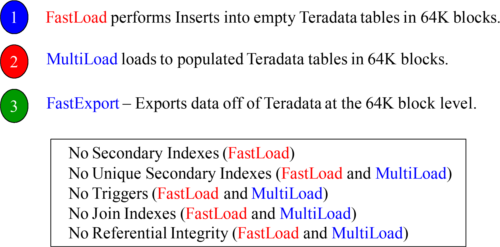
We pressed the EXPORT button, named our output file and hit build script. This took seconds. We can execute the script from Nexus or schedule it to run later or save it.

FastLoad



Above, is the scoop on FastLoad in a nutshell.

BlockLevel Utility Limits

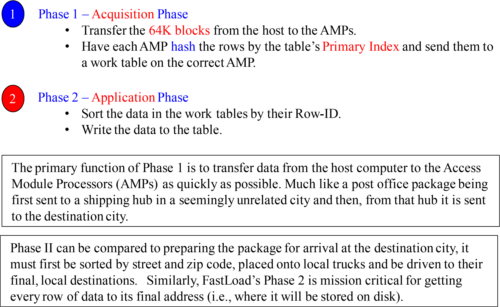


Only 30 FastLoad and MultiLoad combinations can run at one time.

Only 60 FastExport jobs can run simultaneously!

The block utilities have limits that prevent too many of them running at the same time. This feature is actually controlled by a new DBS Control parameter named MaxLoadAWT which controls AMP Worker Tasks (AWT). When MaxLoadAWT is set to zero, then it is like going back in time to pre-V2R6.0 where only 15 FastLoad, MultiLoad and FastExport jobs can run max.

FastLoadhas Two Phases



FastLoad is a very simple utility that only knows the word Insert. It only has two phases and those are to get the data to the AMPs, and then to sort and write the table.

FastLoadPhase 1



•FastLoad uses one SQL session to define AMP steps and another SQL session for log table restart operations.

•FastLoad sends a block of records to each AMP which has a “Load” session.

•The deblocking task within the AMP hashes each record and redistributes the rows.

•Each and every AMP has a receiving task which collects the rows and writes a block of unsorted data rows to disk.

•At the end of Phase 1, each AMP has the rows it should have, but the rows are not in row hash sequence.

Above, are the fine details of what happens in Phase 1 of a FastLoad.

FastLoadPhase 2

“The man who has no imagination has no wings.”

–Muhammad Ali



•When the FastLoad job receives the END LOADING statement, only then does the FastLoad start the End Loading (sort phase).

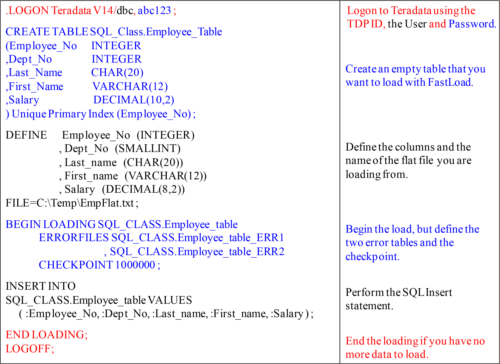
•Each AMP simultaneously sorts the Target table, puts the rows into their blocks, and then writes the blocks to the AMPs assigned disk.

•If the table has Fallback the rows are generated and written.

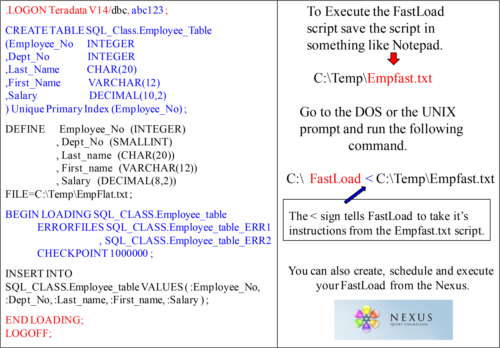
•Table data is now available after the Application Phase completes.

Above, are the fine details of what happens in Phase 2 of a FastLoad because FastLoad stings like a butterfly and floats like the sea. It’s vast and doesn’t hurt!

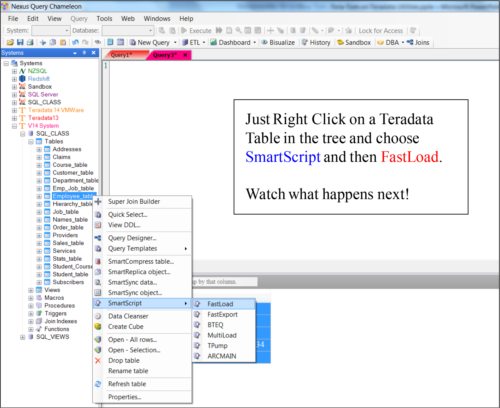
ASample FastLoad Script Created by Nexus SmartScript



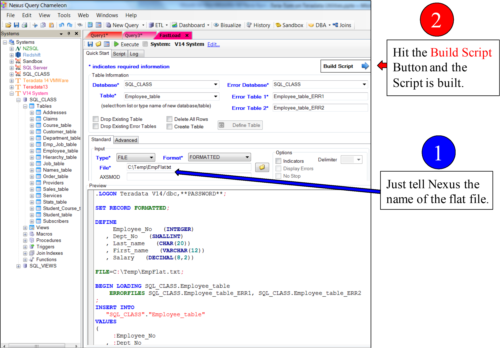
Executingthe FastLoad Script



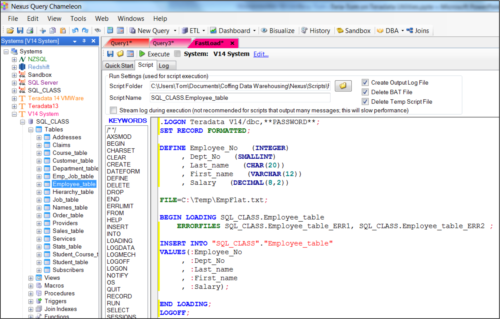
TheNexus SmartScript Easily Builds Your Utilities



TheNexus SmartScript FastLoad Builder

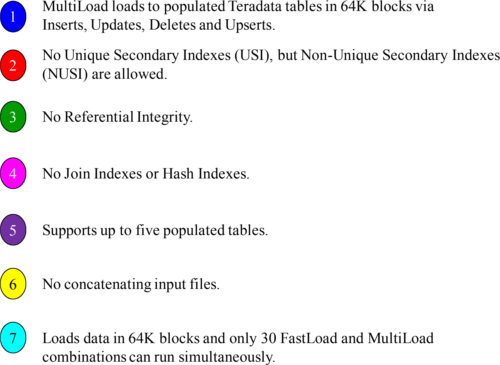


Createand Execute Your FastLoad Scripts with Nexus



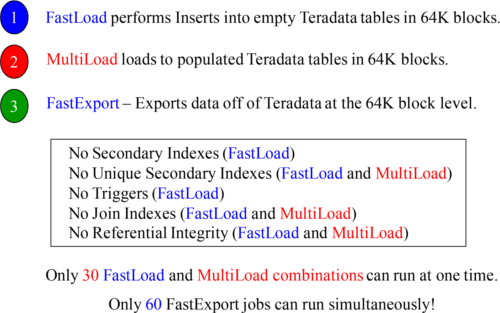
The Nexus SmartScript can create a FastLoad script in under 1 minute. With a few clicks of the mouse, the Nexus will build the script. You can schedule it or run it directly from Nexus. The Nexus even hides your password inside the script!

MultiLoad



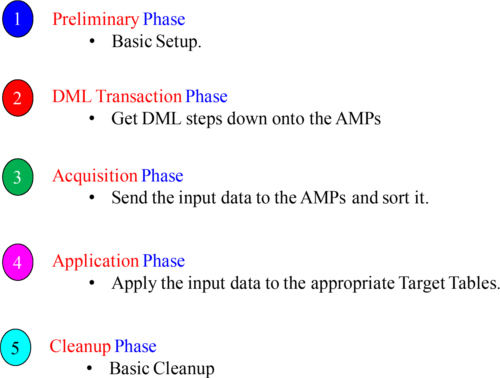
MultiLoad does it all and is used for maintenance on populated tables.

BlockLevel Utility Limits



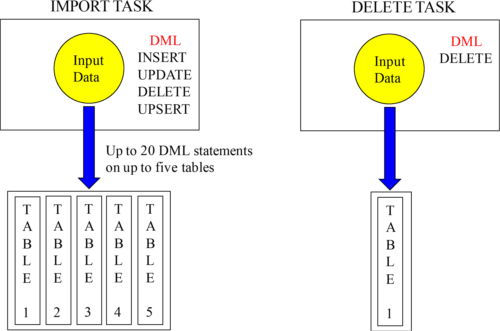
The block utilities have limits that prevent too many of them running at the same time. This feature is actually controlled by a new DBS Control parameter named MaxLoadAWT, which controls AMP Worker Tasks (AWT). When MaxLoadAWT is set to zero, then it is like going back in time to pre-V2R6.0 where only 15 FastLoad, MultiLoad and FastExport jobs can run max, otherwise up to 30 MultiLoads can run.

MultiLoadhas Five Phases



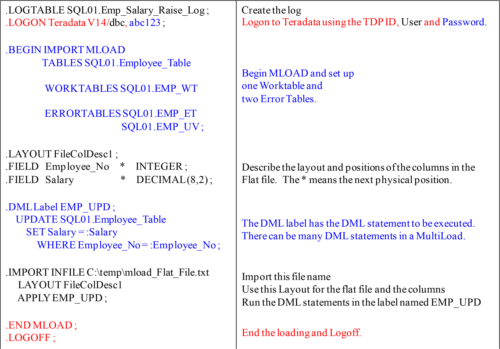
Above, are the five phases of MultiLoad. It is not important that you memorize these, but if a MultiLoad fails it tells you what phase it failed in. Based on which phase it is in during a failure will dictate what actions you need to take to recover or fix the problem.

MultiLoadhas IMPORT and DELETE Tasks

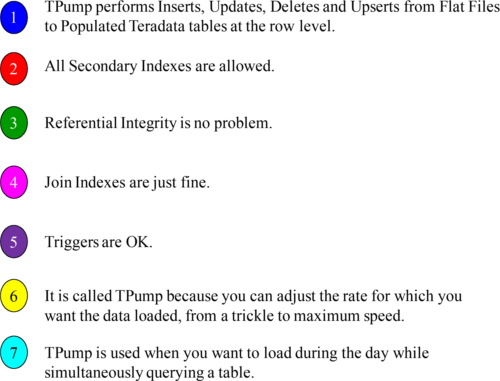


There are two separate tasks in MultiLoad. They are the IMPORT Task and the DELETE Task. MultiLoad can work on up to five tables in the IMPORT task and perform a variety of Inserts, Updates, Deletes, and Upserts.

ASample MultiLoad Script Created by Nexus SmartScript

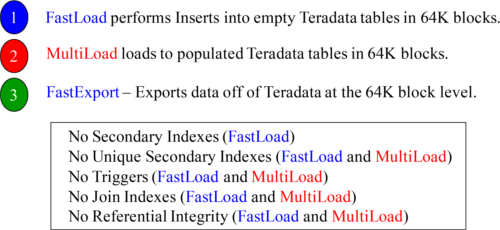


TPump



TPump does it all and is used for maintenance on populated table’s one row at a time. Remember, that TPump is the only load utility that can have its **statement rate dynamically changed** to accommodate peak and latent times?

TPumpis NOT a Block Level Utility and has No Limits

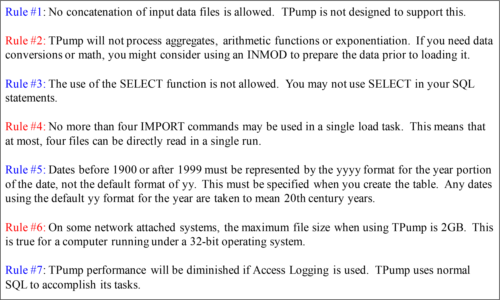


Only 30 FastLoad and MultiLoad combinations can run at one time.

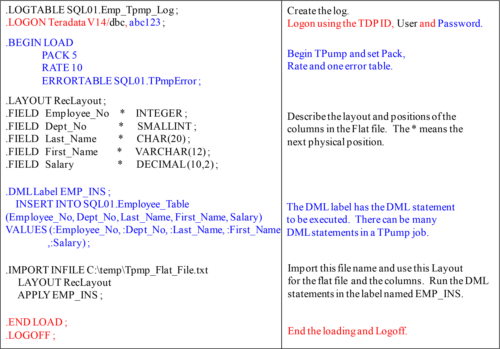
Only 60 FastExport jobs can run simultaneously!

TPump has none of the limits above, so it is perfect when you don’t need to load massive amounts of data or you want to simultaneously load and query a table. It is often used if the tables you’re loading to have secondary indexes, join indexes, triggers, or referential integrity. It is also great when you need hundreds of load jobs to run simultaneously. It is not as fast as FastLoad or MultiLoad, but it is pretty darn fast.

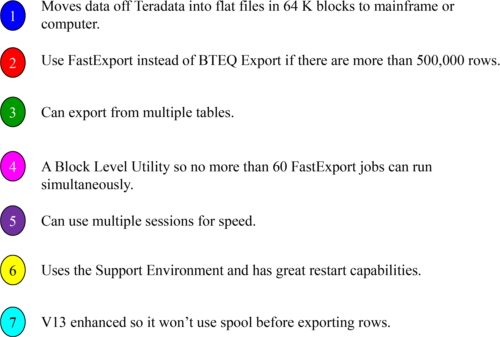
Limitationsof TPump



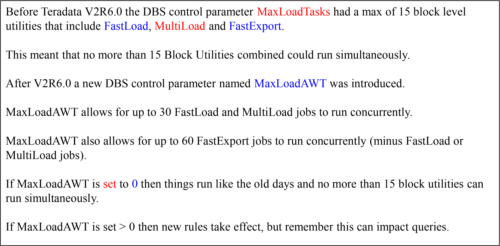
ASample TPump Script Created by Nexus SmartScript



FastExport

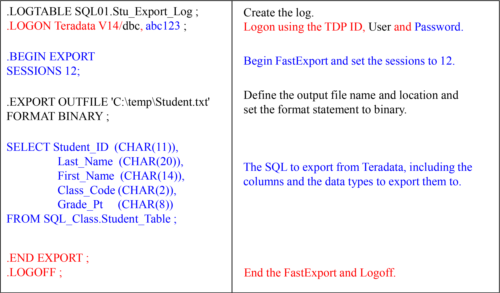


NewRules for Block Utilities

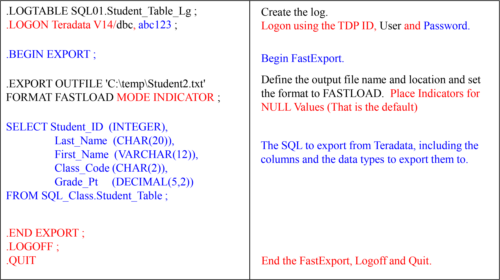


The new rules allow for much more flexibility when dealing with block utilities.

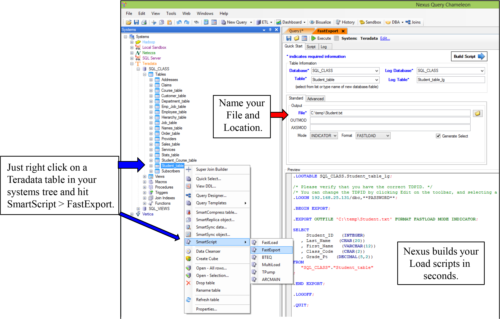
ASample FastExport Script Created by Nexus SmartScript



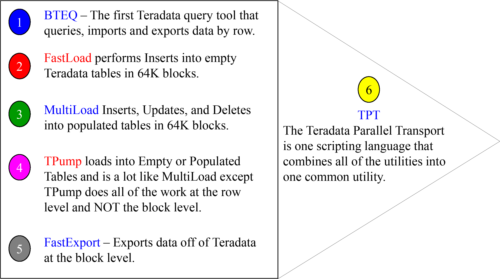
FastExportby Default places Null Indicators in Output



ASample FastExport Script Created by Nexus SmartScript

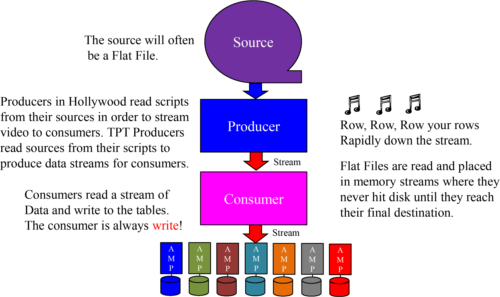


Whatis TPT?



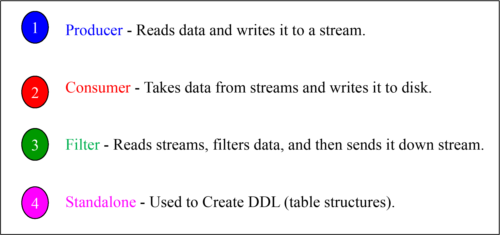
The Teradata Parallel Transport (TPT) utility combines BTEQ, FastLoad, MultiLoad, TPump, and FastExport utilities into one comprehensive language utility. This allows TPT to insert data to tables, export data from tables, and update tables.

TPTProducers Create Streams and Consumers Write Them



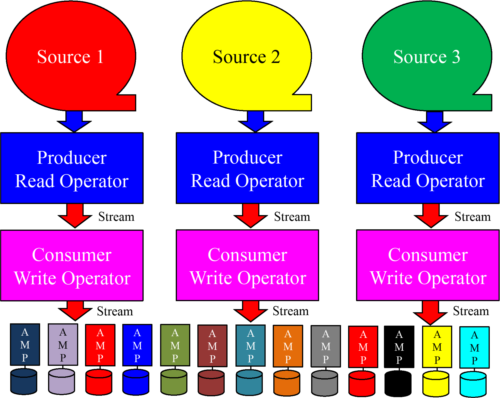
TPT works around the concept of Operators and Data Streams. There will be an Operator to read Source data, pass the contents of that Source to a data stream where another operator will be responsible for taking the Data Stream, and loading it to disk. A Producer Operator, designed to read input, will move the data to a Data Stream. The Consume Operator, designed to write data to a Teradata table, will then Load the data.

TheFour Major Operators of TPT



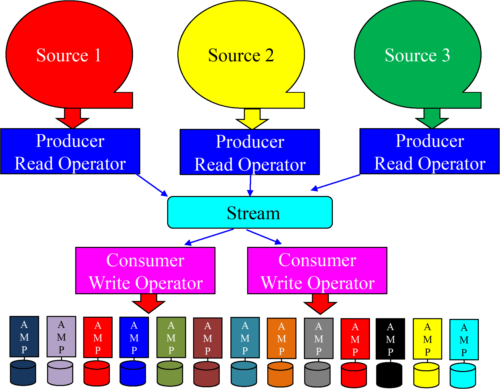
TPT is designed to read data from disk and then place that data in a memory data stream, thus eliminating unnecessary I/O. The Operators above work together to Create Table structures, read data from flat files and send it down the stream, filter data from the stream and then send it further down the stream, and to finally read the stream and write it to its final disk destination.

TPTcan read from multiple source files in Parallel



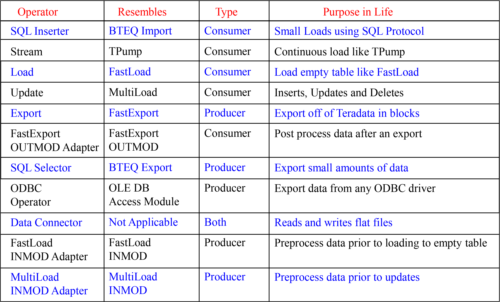
One of the clever concepts behind TPT is that you can take multiple sources (multiple flat files for example), and utilize multiple Producer Operators to reach each source and then move multiple data streams to multiple Consumer Operators. These operators will then write the multiple sources to one Teradata table. This couldn't be done before with FastLoad or MultiLoad.

TPTcan have more Operators than Consumers

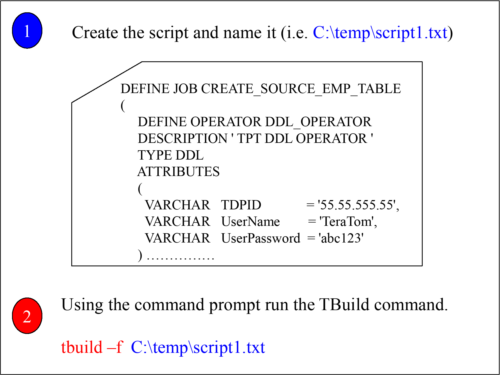


Each source can be handled by a different instance of the producer operator which merges these files into a single data stream.

TPTOperators and their Equivalent Load Utility



Howto Run a TPT Script



The easiest way to run a TPT script is to use the TBuild utility. You first create your script and then run TBuild; passing TBuild the script name to run.