Frequently Asked Questions:
Teradata Parallel Transporter In MicroStrategy

1) What is TPTAPI in general?

TPT API stands for Teradata Parallel Transporter API and is an alternative means to load/unload data between a Teradata Database Server and Client application apart from the traditional ODBC. Teradata, over the years has built a number of utilities to exchange data with it, these include “FastLoad”, “MultiLoad”, “TPump” and “FastExport”. TPT API essentially combines all these utilities in a way that they are accessible via a single scripting language utility. In addition, client-side processing has been parallelized by implementing the capability of spawning multiple “instances”, in addition to multiple sessions to more effectively move any bottleneck away from the client platform. By invoking these various utilities from within TPT API, user will be able add, delete, update data in Teradata tables and export data out of Teradata systems. You can learn more about TPTAPI in Chapter 1 of the Teradata Parallel Transporter Application Programmer Interface as well as other Teradata Parallel Transporter documentation that is available for download on the Teradata website (http://www.info.teradata.com).

2) How does MicroStrategy use TPTAPI ?

In MicroStrategy’s case we have specifically implemented invoking the “FastExport” protocol utilizing the Export Operator from TPTAPI and be able to export data quickly out of Teradata into MicroStrategy Cubes. The “FastExport” protocol is capable of exporting data out of Teradata utilizing parallel sessions and therefore has a higher throughput rate than a single session traditional ODBC. TPTAPI further optimizes throughput by enabling multiple “instances”.

3) When Should I use TPTAPI ?

A common question that people ask is that given that TPTAPI is retrieving data in parallel sessions wouldn’t I want to always use it rather than using ODBC. The simple answer is no and there are three reasons for it.

1) The main reason is that TPTAPI Export was designed for bulk export of data from Teradata and there is a certain time overhead for these export processes and the requisite parallelization that happens. This implies that TPTAPI Export delivers a benefit only when the data volume goes over a certain limit and in our experiments we found that a MicroStrategy cube with ½ GB of data is a good rule of thumb in terms when users start seeing data retrieval performance benefits from TPTAPI from MicroStrategy.

2) It is important to understand that TPTAPI Export uses a "Loader Slot" on the Teradata system. These "Loader Slots" are specially designed to move data both into and out of Teradata consuming resources to achieve higher throughput, and there is a limit to the number of "Loader Slots" that can be run concurrently at any point of time. DBAs manage this via the Workload Management Product (Viewpoint) and like to control these tasks. Given the limited number of loader slots, TPTAPI export is only appropriate for the biggest MicroStrategy Cubes.

3) Finally, the least important reason that rarely comes into play is that there are certain SQLs that cannot be run on TPTAPI. For example, in the TPT Application Programmer Guide 15.0 the SQL Request restrictions specify that TPTAPI's Select statement cannot use the 'USING' modifier or utilize variable substitution among others. Please refer to the TPT Application Programmer Guide to go through all the 'Select Request Restrictions' that apply for the select clause that is used within the Fast Export utility.

4) In MicroStrategy Developer, Can TPTAPI be used for both ROLAP report and OLAP Cubes ?

The short answer is "Yes". The long answer is that given the understanding that TPTAPI is appropriate when the data that is brought back is more than 500 MB, it is natural that OLAP Cube reports are natural places to use TPTAPI. Most ROLAP reports don't retrieve a large amount of data and it is therefore not the intended application point for TPTAPI.

Having said that, ROLAP reports are great way to try out and check if a particular report is retrieving the appropriate data via TPTAPI. This can be done by adding additional filters that restrict the amount of data that is retrieved and once TPTAPI is retrieving the data correctly in the ROLAP report, the user can make into a Cube report and remove the extra filters that were applied. All in all, it is great to have the TPTAPI functionality in ROLAP reports for testing purposes.

5) In MicroStrategy Web, Can TPTAPI be used for both Direct Access and In-memory Cubes ?

The short answer is "Yes". The longer answer is that TPTAPI is appropriate when the data is greater than 500 MB (as referenced above) and it is used in conjunction with In-memory Cubes. Most Direct Access cubes are not intended for retrieving a large amount of data and it is therefore not the intended application point for TPTAPI. Similar to the MicroStrategy Developer case, one could use the DDA mode to test out various scenarios for retrieving data from TPTAPI.

6) What setup is required for TPTAPI ?

In order to be able to connect via TPTAPI the user must install certain Teradata software on the Intelligence Server so that the Intelligence Server can connect via TPTAPI. Please note, TPTAPI is a Command Line Interface (CLI) application (vs. ODBC), therefore in addition to ODBC, you will need to install minimally Teradata CLIv2, and TPTAPI (which is currently included in the TPTBASE package). Currently MicroStrategy will utilize the EXPORT Operator only. Therefore, at a minimum, the following 64 bit versions are needed:

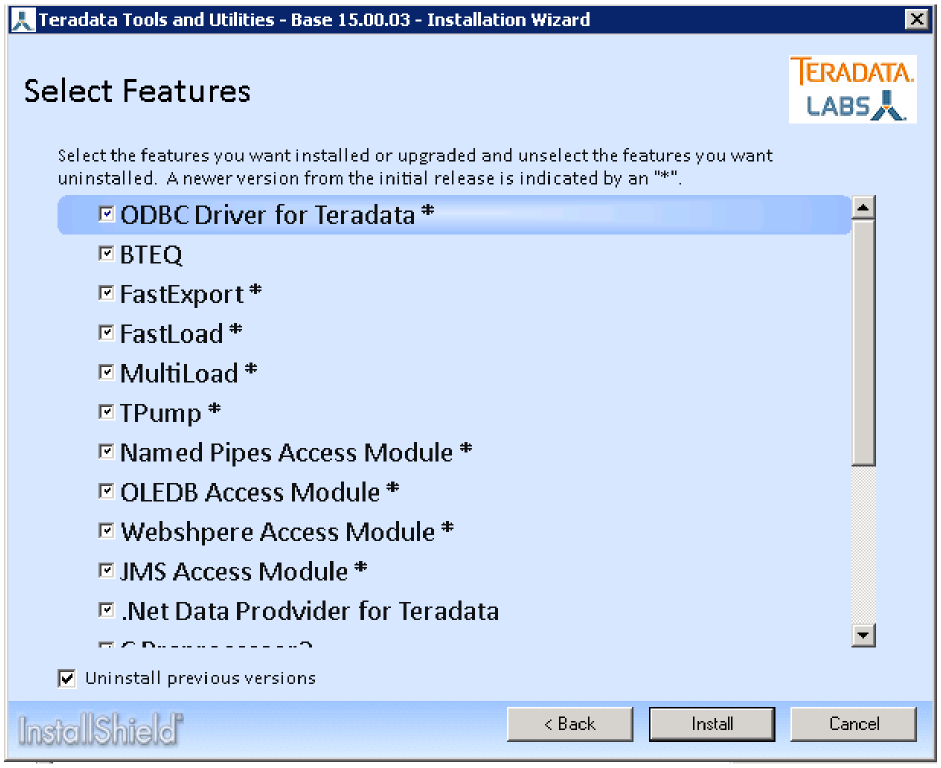
- Shared ICU Libraries for Teradata (TDICU)
- Teradata GSS Client (TERAGSS)
- ODBC Driver for Teradata
- Teradata CLIv2 (Cliv2)
- Teradata Parallel Transporter Base (TPT Base)

Teradata recommends installing the TTU components initially from the TTU installation media to ensure packages are installed in the appropriate order.

TPTAPI support is available beginning with MicroStrategy 10.1. Currently supported I-Server Operating Systems for TPTAPI are Windows and Linux only and the supported Teradata Tools and Utilities version is 15.0/15.1.

The installation details are as follows:

a) Windows Environment: The user has to install Teradata Tools and Utilities (TTU) as shown below. In our tests we have utilized the 15.00.03 version and installed all the available modules.



Teradata Tools and Utilities - Base 15.00.03 - Installation Wizard

Select Features

Select the features you want installed or upgraded and unselect the features you want uninstalled. A newer version from the initial release is indicated by an "*".

- ☑ ODBC Driver for Teradata *
- ☑ BTEQ
- ☑ FastExport *
- ☑ FastLoad *
- ☑ MultiLoad *
- ☑ TPump *
- ☑ Named Pipes Access Module *
- ☑ OLEDB Access Module *
- ☑ Webshpere Access Module *
- ☑ JMS Access Module *
- ☑ .Net Data Prodvider for Teradata

☑ Uninstall previous versions

b) For Linux environment

Please install the Teradata TTU (Tool and utilities) and some adjustment might be needed for the environment variables.

Teradata Version: 14.10, 15.00

1. Csh Script

```
setenv NLSPATH /opt/teradata/client/15.00/odbc_64/msg/%N #Added by ODBC14.10 Package
setenv COPLIB /opt/teradata/client/15.00/lib64 # Added by cliv2 14.10 package
setenv COPERR /opt/teradata/client/15.00/lib # Added by cliv2 14.10 package
setenv TWB_ROOT /opt/teradata/client/15.00/tbuild

setenv PATH $TWB_ROOT/bin64:$PATH
setenv LD_LIBRARY_PATH $TWB_ROOT/lib64:/usr/lib
setenv NLSPATH $TWB_ROOT/msg/%N:$NLSPATH
setenv NLSPATH $TWB_ROOT/msg/opermsgs.cat:$NLSPATH

setenv CLASSPATH /opt/teradata/client/15.00/jmsaxsmod/jmsam.jar:$CLASSPATH
setenv                                                          LD_LIBRARY_PATH
/usr/odbc/lib:$LD_LIBRARY_PATH/:/usr/odbc/drivers:/opt/teradata/client/15.00/lib64

setenv THREADONOFF= 1
```

2. Bash Script

```
# The Green Highlights are the changes that we have made in the profile after installing TTU

#
# End of /etc/profile
#
TD_ICU_DATA=/opt/teradata/client/15.00/tdicu/lib; export TD_ICU_DATA   # Added by tdicu 15.00 package
COPLIB=/opt/teradata/client/15.00/lib64; export COPLIB # Added by cliv2 15.00 package
COPERR=/opt/teradata/client/15.00/lib; export COPERR # Added by cliv2 15.00 package

if [ "${PATH}" = "" ]; then             # Modified by TPTBASE1500 package
  PATH=/opt/teradata/client/15.00/tbuild/bin64 # Modified by TPTBASE1500 package
else                                  # Modified by TPTBASE1500 package
  PATH=/opt/teradata/client/15.00/tbuild/bin64:${PATH} # Modified by TPTBASE1500 package
fi                                    # Modified by TPTBASE1500 package
export PATH                           # Modified by TPTBASE1500 package

if [ -z "$CLASSPATH" ]; then                     # Added by jmsaxsmod 15.00 package
   CLASSPATH=/opt/teradata/client/15.00/jmsaxsmod/jmsam.jar;  export  CLASSPATH      #  Added  by  jmsaxsmod
15.00 package
else                                          # Added by jmsaxsmod 15.00 package
   CLASSPATH=/opt/teradata/client/15.00/jmsaxsmod/jmsam.jar:$CLASSPATH;  export  CLASSPATH      #  Added  by
jmsaxsmod 15.00 package
fi                                            # Added by jmsaxsmod 15.00 package


if [ -z "$LD_LIBRARY_PATH" ]; then                 # Added by jmsaxsmod 15.00 package
  LD_LIBRARY_PATH=/opt/teradata/client/15.00/lib64; export LD_LIBRARY_PATH      # Added by jmsaxsmod 15.00
package
else                                          # Added by jmsaxsmod 15.00 package
  LD_LIBRARY_PATH=/opt/teradata/client/15.00/lib64:$LD_LIBRARY_PATH; export LD_LIBRARY_PATH      # Added
by jmsaxsmod 15.00 package
fi                                            # Added by jmsaxsmod 15.00 package

MANPATH=/opt/teradata/client/15.00/odbc_64/help/man:$MANPATH; export MANPATH #Added by ODBC15.00
Package
NLSPATH=/opt/teradata/client/15.00/odbc_64/msg/%N; export NLSPATH #Added by ODBC15.00 Package

export TWB_ROOT /opt/teradata/client/15.00/tbuild
export
LD_LIBRARY_PATH=/opt/teradata/client/15.00/tbuild/lib64:/usr/lib:/usr/odbc/lib:/usr/odbc/drivers:$LD_LIBRARY
_PATH
#export LD_LIBRARY_PATH=/opt/teradata/client/15.00/tbuild/msg/opermsgs.cat:$LD_LIBRARY_PATH
NLSPATH=/opt/teradata/client/15.00/tbuild/msg/%N:/opt/teradata/client/15.00/tbuild/msg/opermsgs.cat:$NLSP
ATH; export NLSPATH
export THREADONOFF=1
```

Teradata Version: 15.10

1. Csh Script

```
setenv NLSPATH /opt/teradata/client/15.10/odbc_64/msg/%N
setenv COPLIB /opt/teradata/client/15.10/lib64
setenv COPERR /opt/teradata/client/15.10/lib
setenv TWB_ROOT /opt/teradata/client/15.10/tbuild

setenv PATH $TWB_ROOT/bin64:$PATH
setenv LD_LIBRARY_PATH $TWB_ROOT/lib64:/usr/lib
setenv NLSPATH $TWB_ROOT/msg/%N:$NLSPATH
setenv NLSPATH $TWB_ROOT/msg/opermsgs.cat:$NLSPATH

setenv CLASSPATH /opt/teradata/client/15.10/jmsaxsmod/jmsam.jar:$CLASSPATH
setenv                                                             LD_LIBRARY_PATH
/usr/odbc/lib:$LD_LIBRARY_PATH/:/usr/odbc/drivers:/opt/teradata/client/15.10/lib64

setenv THREADONOFF= 1
```

2. Bash Script

```
# The Green Highlights are the changes that we have made in the profile after installing TTU

#
# End of /etc/profile
#
TTU_PATH=/opt/teradata/client/15.10
if [ "${PATH}" = "" ]; then
  PATH="${TTU_PATH}/bin"
else
  PATH="${TTU_PATH}/bin:${PATH}"
fi
export TTU_PATH PATH

#add libpath to LD_LIBRARY_PATH
if [ -d "${TTU_PATH}/lib64" ]
then
  libpath="${TTU_PATH}/lib64:${TTU_PATH}/lib"
else
  libpath="${TTU_PATH}/lib"
fi
if [ "${LD_LIBRARY_PATH}" = "" ]; then
  LD_LIBRARY_PATH="${libpath}"
else
  LD_LIBRARY_PATH="${libpath}:${LD_LIBRARY_PATH}"
fi
export LD_LIBRARY_PATH

COPLIB="${TTU_PATH}/lib64"; export COPLIB
COPERR="${TTU_PATH}/lib"; export COPERR
```

Actually this part is from Teradata folder: /opt/teradata/client/15.10/etc/ttu_1510_bash.env

```
if [ -z "$CLASSPATH" ]; then
  CLASSPATH=/opt/teradata/client/15.10/jmsaxsmod/jmsam.jar;
else
 CLASSPATH=/opt/teradata/client/15.10/jmsaxsmod/jmsam.jar:$CLASSPATH;
fi
export CLASSPATH

export TWB_ROOT=/opt/teradata/client/15.10/tbuild
export NLSPATH=/opt/teradata/client/15.10/odbc_64/msg/%N
export
NLSPATH=/opt/teradata/client/15.10/tbuild/msg/%N:/opt/teradata/client/15.10/msg/opermsgs.cat:$NLSPATH
export THREADONOFF=1
```

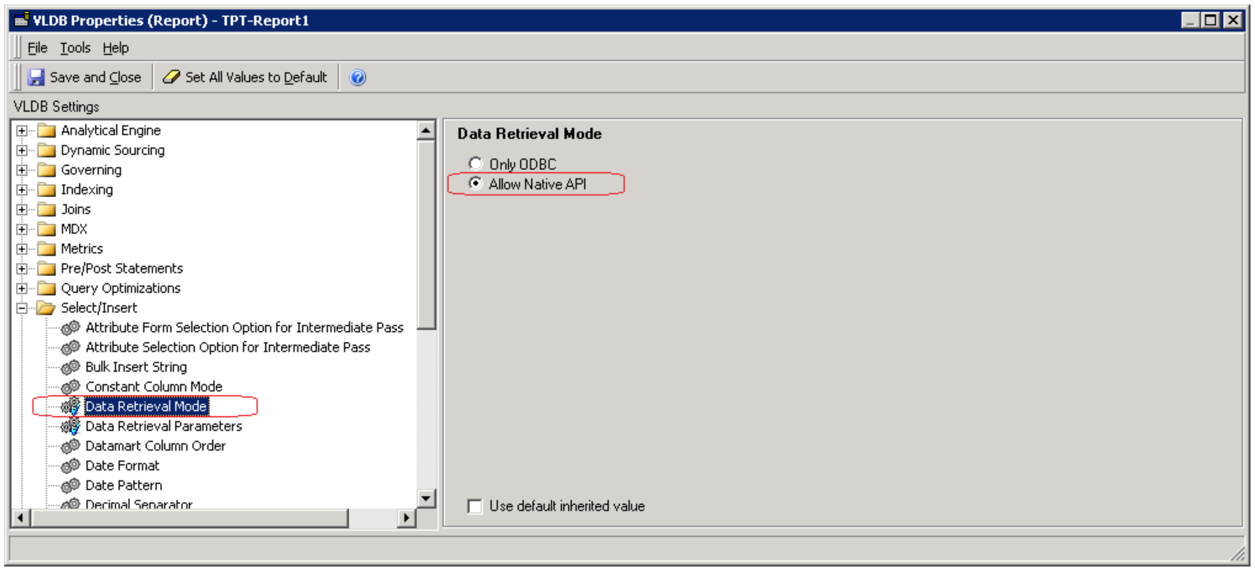
You may need to restart your machine to bring the environment variables into effect.

It is highly recommended to perform a smoke test to validate the TPTAPI installation and all pre-requisites have been satisfied by running the pre-compiled TPTAPI sample programs included with the TPTAPI installations BEFORE attempting to use with MicroStrategy. An intitial test with BTEQ (SELECT * from DBC.DBCINFO) will provide feedback if basic connectivity is established successfully – then proceed to test TPTAPI. Please refer to the Teradata TPTAPI Programmer Guide (available from Teradata Documentation site http://www.info.teradata.com). TPTAPI 15.0 documentation includes this information in Appendix B: Code Samples. Pre-compiled samples are in the /runsamp directory within the tptapi installation folders. See Teradata TPTAPI Programmer Guide Appendix B: for detailed usage instructions. Once the TPTAPI installation has been validated, usage with MicroStrategy can proceed.

7) How to run a report with TPTAPI setup in MSTR Developer ?

Step 1) Firstly, you should have MicroStrategy setup so that you are able to run the report and retrieve the requisite data via ODBC like usual.

Step 2) At the report level – set the 'Data Retrieval Mode' to the setting "Allow Native API".

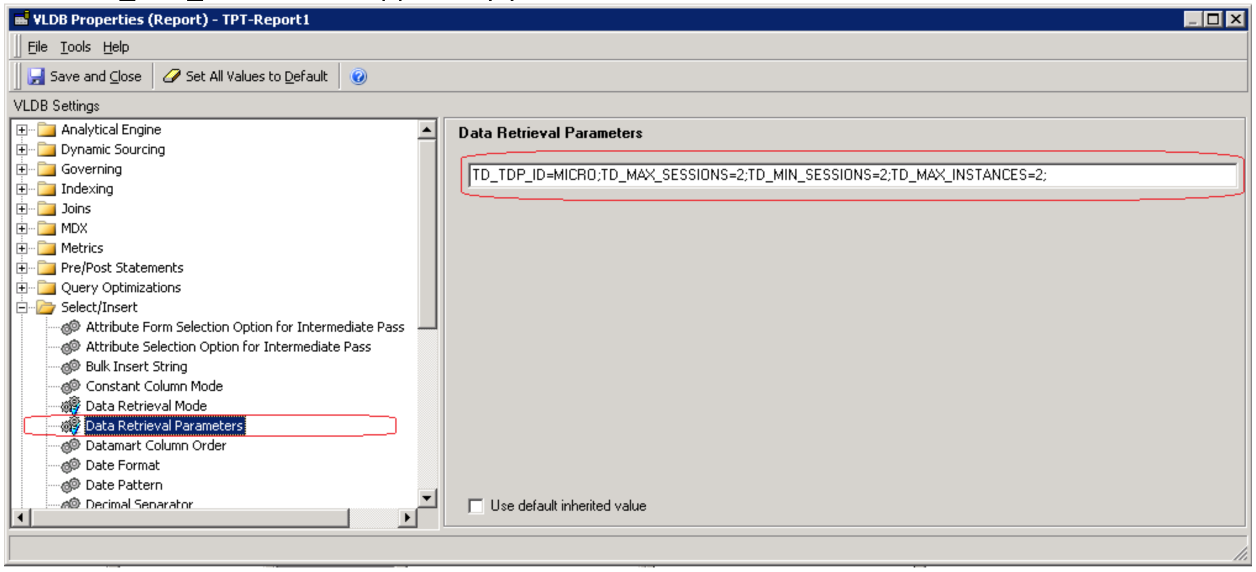

VLDB Properties (Report) - TPT-Report1

Data Retrieval Mode

Only ODBC [✓ Allow Native API]

☐ Use default inherited value

Step 3) At the report level – set the 'Data Retrieval Parameters' to the following setting
Note: TD_TDP_ID must be supplied by your Teradata database administrator


VLDB Properties (Report) - TPT-Report1

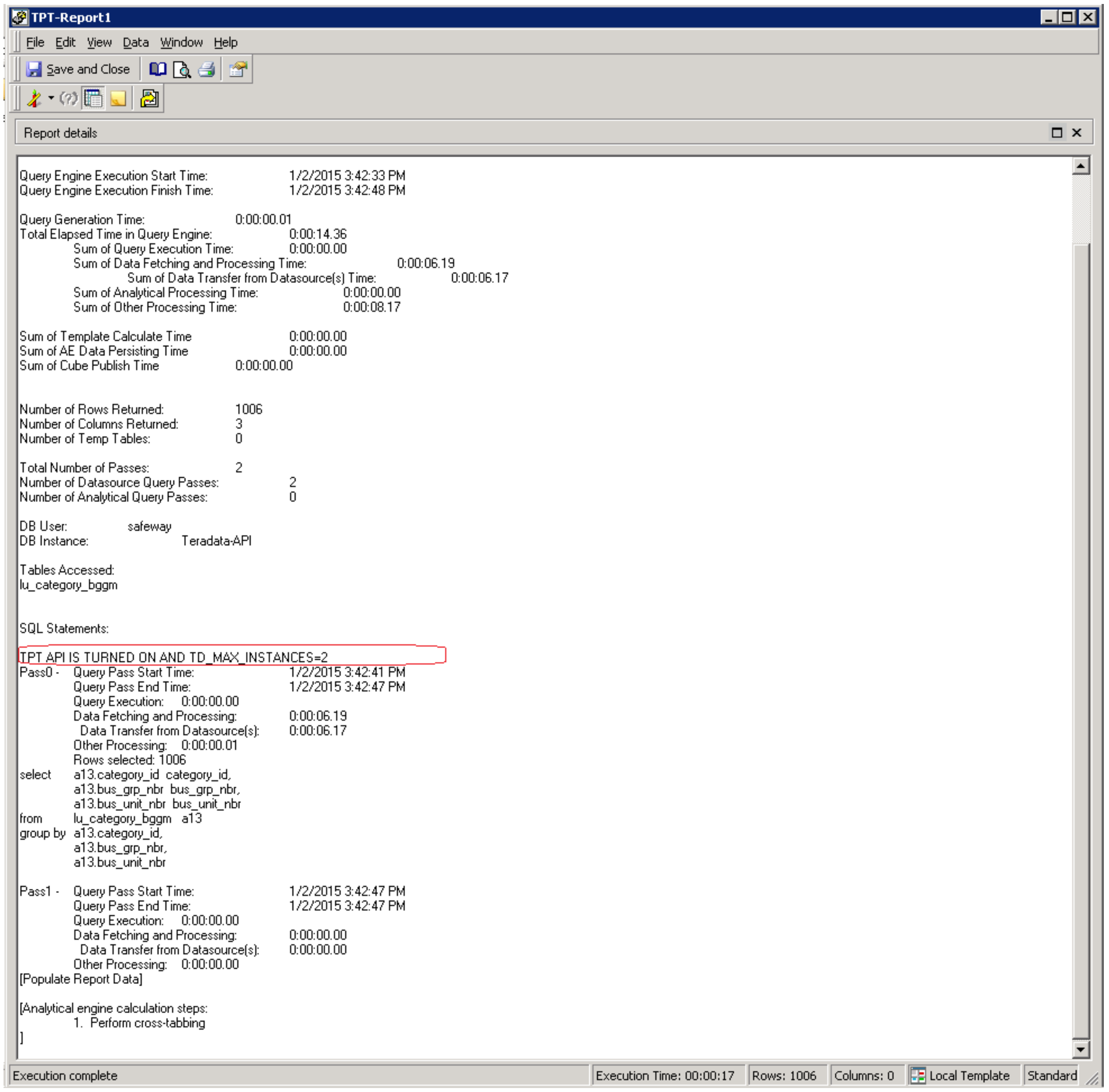
Data Retrieval Parameters

```
TD_TDP_ID=MICRO;TD_MAX_SESSIONS=2;TD_MIN_SESSIONS=2;TD_MAX_INSTANCES=2;
```

☐ Use default inherited value

Note: Even though the VLDB settings Data Retrieval Mode and Data Retrieval Properties are available at both Report level and database instance level. For the case of MicroStrategy Developer, it is recommended that you set these VLDB settings at the report level for specific cube reports on a case by case manner if they are retrieving large amounts of data from a warehouse.

Step 4) Once these parameters are set you should be able to retrieve the data via TPTAPI. Upon running the report you will notice the below message "TPT API IS TURNED ON AND TD_MAX_INSTANCES=*" on the top of all the select passes that have been retrieved by TPTAPI.



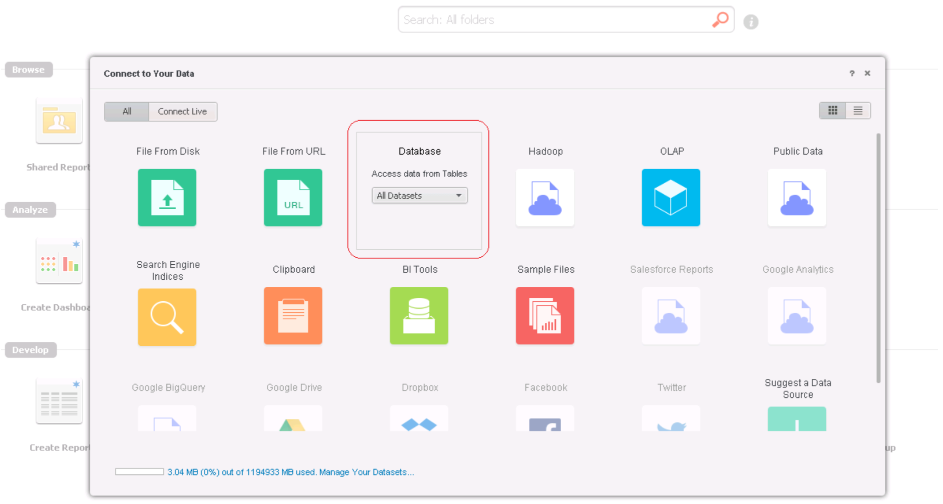
```
Query Engine Execution Start Time:          1/2/2015 3:42:33 PM
Query Engine Execution Finish Time:         1/2/2015 3:42:48 PM

Query Generation Time:                 0:00:00.01
Total Elapsed Time in Query Engine:         0:00:14.36
        Sum of Query Execution Time:        0:00:00.00
        Sum of Data Fetching and Processing Time:        0:00:06.19
                Sum of Data Transfer from Datasource(s) Time:        0:00:06.17
        Sum of Analytical Processing Time:        0:00:00.00
        Sum of Other Processing Time:             0:00:08.17

Sum of Template Calculate Time              0:00:00.00
Sum of AE Data Persisting Time              0:00:00.00
Sum of Cube Publish Time               0:00:00.00


Number of Rows Returned:               1006
Number of Columns Returned:            3
Number of Temp Tables:                 0

Total Number of Passes:                2
Number of Datasource Query Passes:          2
Number of Analytical Query Passes:          0

DB User:           safeway
DB Instance:                Teradata-API

Tables Accessed:
lu_category_bggm


SQL Statements:

TPT API IS TURNED ON AND TD_MAX_INSTANCES=2
Pass0 -  Query Pass Start Time:             1/2/2015 3:42:41 PM
         Query Pass End Time:               1/2/2015 3:42:47 PM
         Query Execution:    0:00:00.00
         Data Fetching and Processing:      0:00:06.19
          Data Transfer from Datasource(s):  0:00:06.17
         Other Processing:   0:00:00.01
         Rows selected: 1006
select   a13.category_id  category_id,
         a13.bus_grp_nbr  bus_grp_nbr,
         a13.bus_unit_nbr  bus_unit_nbr
from     lu_category_bggm  a13
group by  a13.category_id,
         a13.bus_grp_nbr,
         a13.bus_unit_nbr

Pass1 -  Query Pass Start Time:             1/2/2015 3:42:47 PM
         Query Pass End Time:               1/2/2015 3:42:47 PM
         Query Execution:    0:00:00.00
         Data Fetching and Processing:      0:00:00.00
          Data Transfer from Datasource(s):  0:00:00.00
         Other Processing:   0:00:00.00
[Populate Report Data]

[Analytical engine calculation steps:
         1.  Perform cross-tabbing
]
```

Execution complete | Execution Time: 00:00:17 | Rows: 1006 | Columns: 0 | Local Template | Standard

Note: If the SQL that is being generated is Multi-Pass SQL then please refer to Question 11.

8) How to build a cube with TPTAPI setup in MSTR Web?

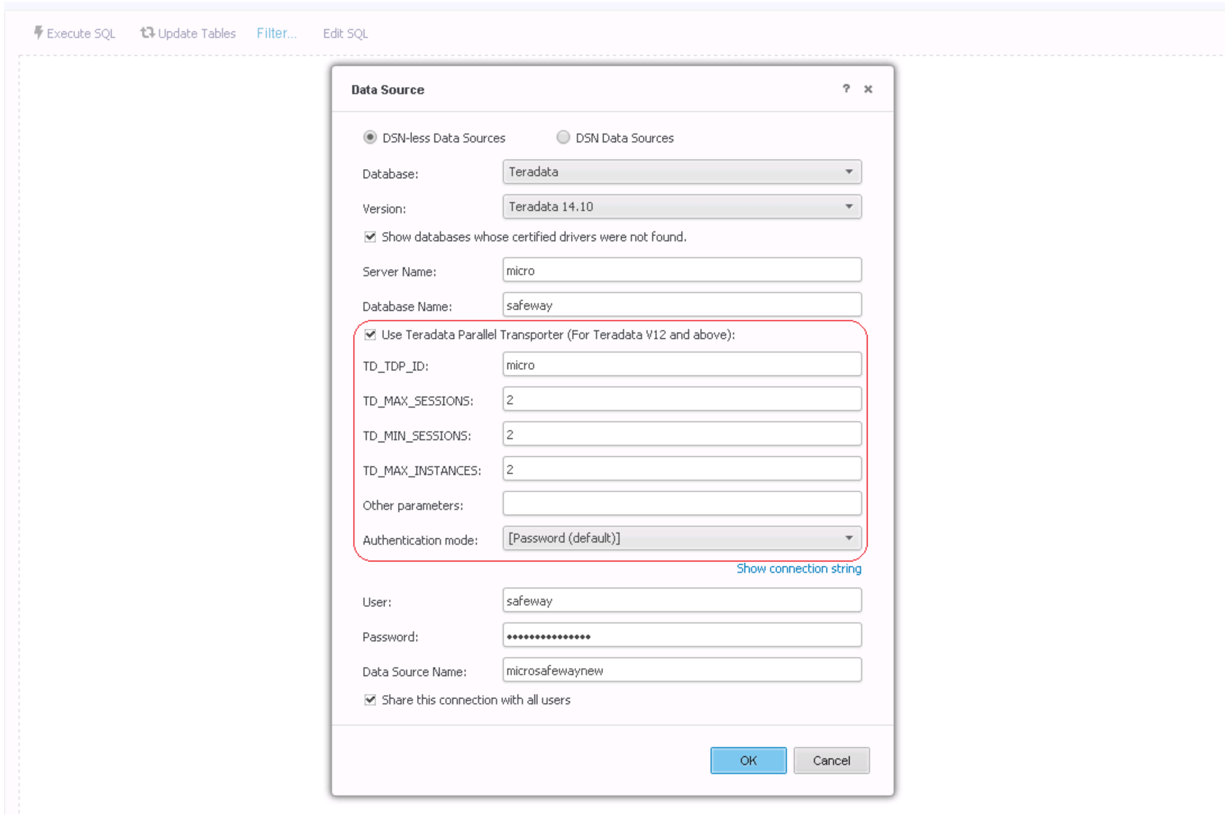
Step 0) Open the 'Build your Query' interface in MicroStrategy Web



Step 1) Create a database instance as shown below and note that apart from the usual 'Server Name', 'Database Name', 'User', 'Password', 'Data Source Name' for ODBC connections, the following changes must be made.

1) The user must set the checkbox 'Use Teradata Parallel Transporter' (this check box sets the 'Data Retrieval Mode' VLDB setting at the database connection level)

2) The user must set the parameters 'TD_TDP_ID', 'TD_MAX_SESSIONS', 'TD_MIN_SESSIONS', 'TD_MAX_INSTANCES' (these textboxes set the 'Data Retrieval Mode' VLDB setting at the database connection level)



(● DSN-less Data Sources) ○ DSN Data Sources

Database:	Teradata
Version:	Teradata 14.10

☑ Show databases whose certified drivers were not found.

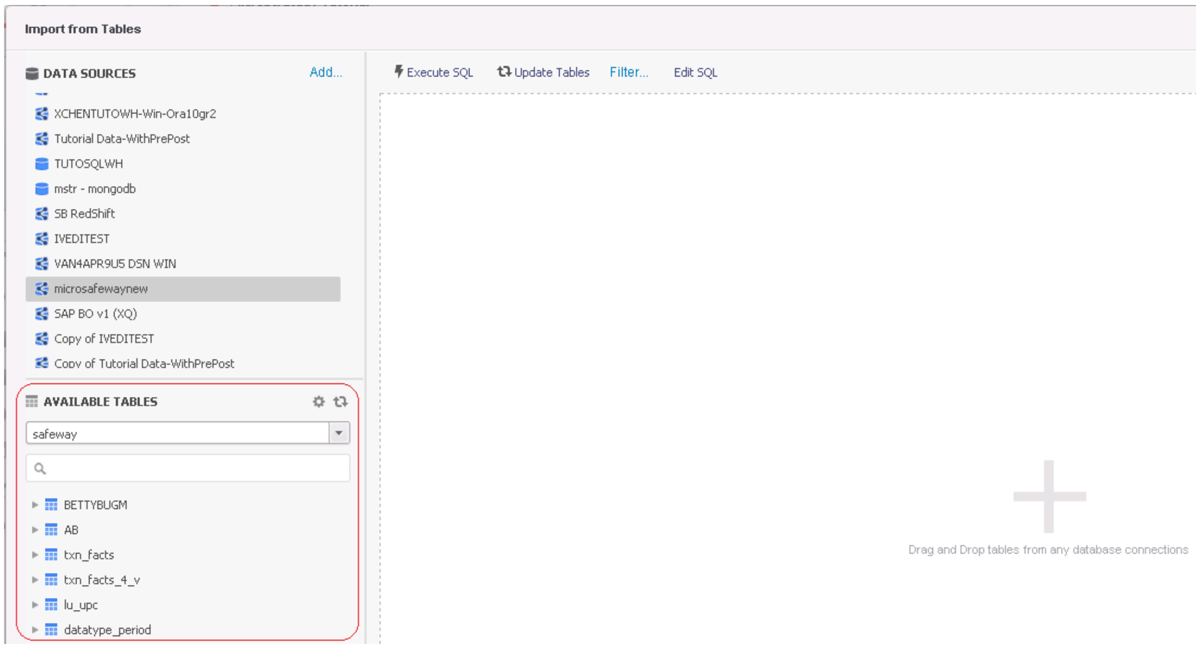
Server Name:	micro
Database Name:	safeway

☑ Use Teradata Parallel Transporter (For Teradata V12 and above):

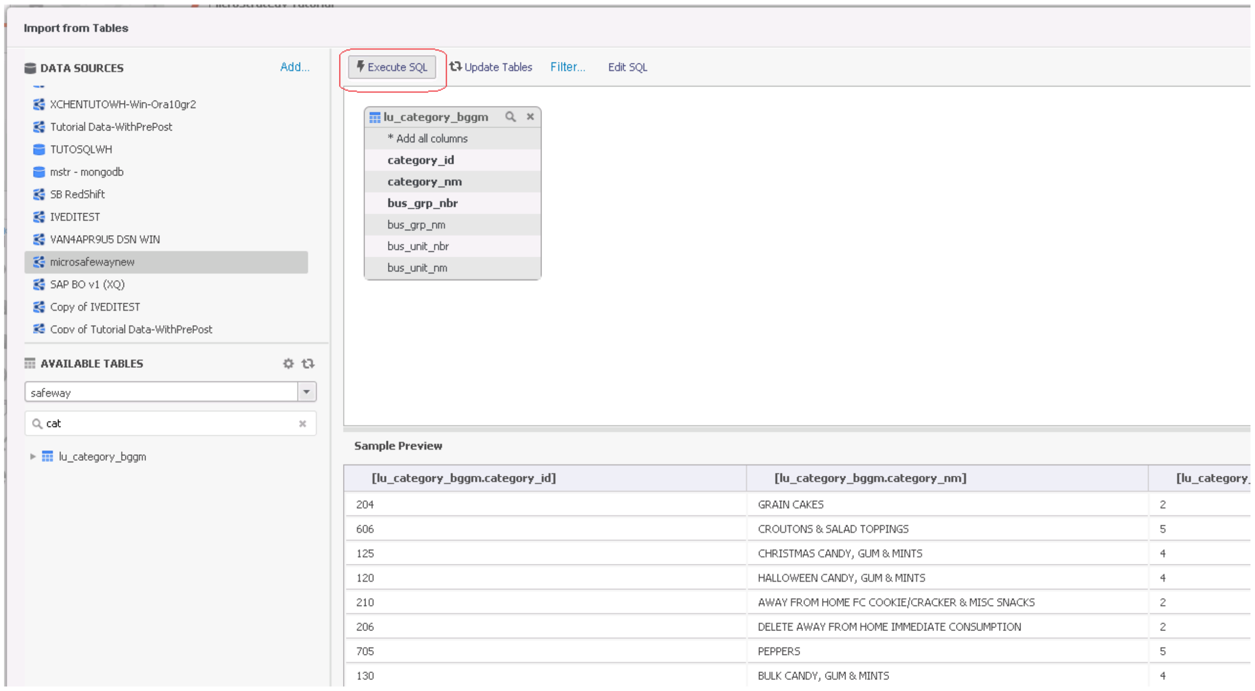
TD_TDP_ID:	micro
TD_MAX_SESSIONS:	2
TD_MIN_SESSIONS:	2
TD_MAX_INSTANCES:	2
Other parameters:	
Authentication mode:	[Password (default)]
User:	safeway
Password:	•••••••••••••••
Data Source Name:	microsafewaynew

☑ Share this connection with all users

Step 2) Upon connecting creating the database instance, click on the particular database instance and enter the relevant namespace. The calls for retrieving the tables are made through ODBC.

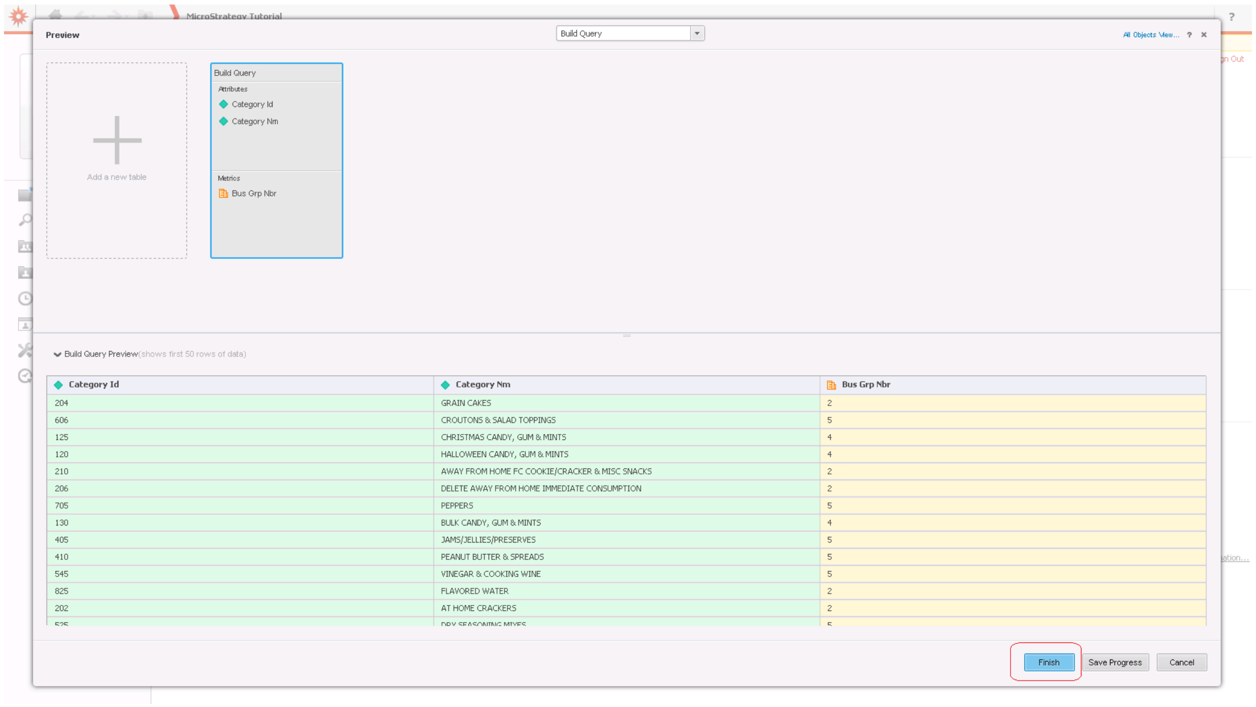


Step 3) Bring in the appropriate tables and create a query. Note that the sample preview data is retrieved via ODBC.



[lu_category_bggm.category_id]	[lu_category_bggm.category_nm]	[lu_category
204	GRAIN CAKES	2
606	CROUTONS & SALAD TOPPINGS	5
125	CHRISTMAS CANDY, GUM & MINTS	4
120	HALLOWEEN CANDY, GUM & MINTS	4
210	AWAY FROM HOME FC COOKIE/CRACKER & MISC SNACKS	2
206	DELETE AWAY FROM HOME IMMEDIATE CONSUMPTION	2
705	PEPPERS	5
130	BULK CANDY, GUM & MINTS	4

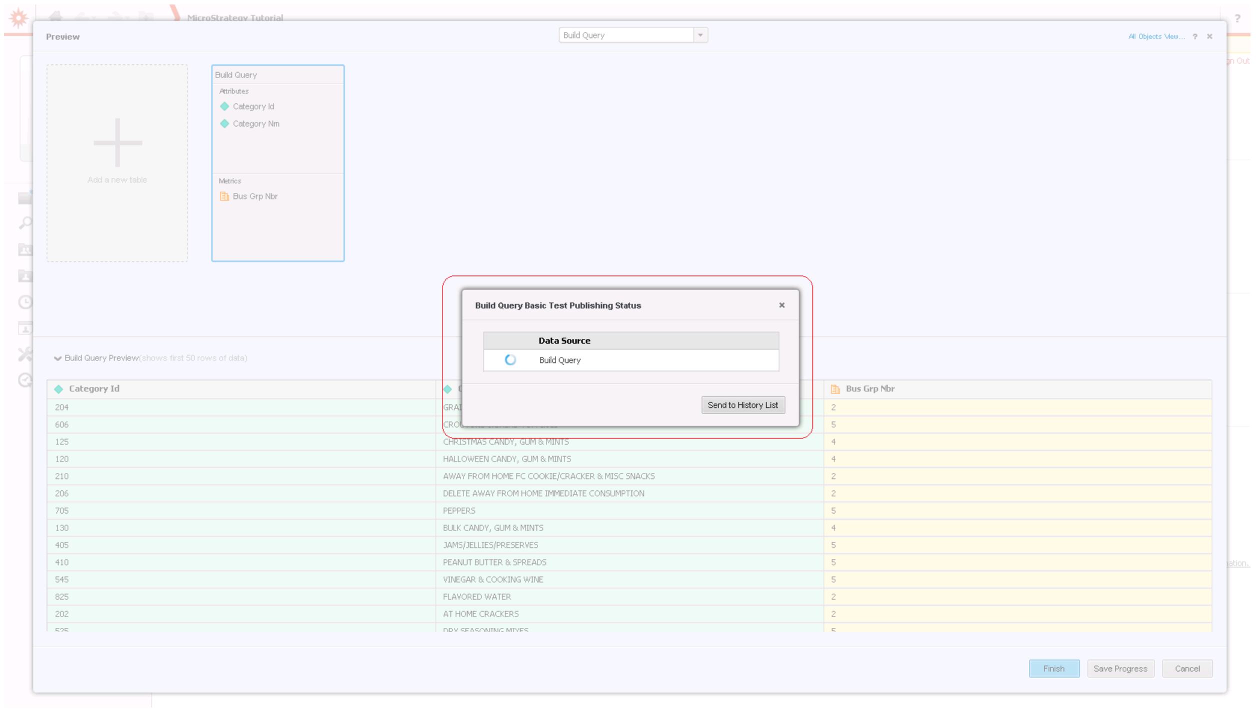
Step 4) Click on 'Finish' at this point (note that the data shown in the preview is still retrieved via ODBC).



Build Query Preview(shows first 50 rows of data)

Category Id	Category Nm	Bus Grp Nbr
204	GRAIN CAKES	2
606	CROUTONS & SALAD TOPPINGS	5
125	CHRISTMAS CANDY, GUM & MINTS	4
120	HALLOWEEN CANDY, GUM & MINTS	4
210	AWAY FROM HOME FC COOKIE/CRACKER & MISC SNACKS	2
206	DELETE AWAY FROM HOME IMMEDIATE CONSUMPTION	2
705	PEPPERS	5
130	BULK CANDY, GUM & MINTS	4
405	JAMS/JELLIES/PRESERVES	5
410	PEANUT BUTTER & SPREADS	5
545	VINEGAR & COOKING WINE	5
825	FLAVORED WATER	2
202	AT HOME CRACKERS	2

Step 5) Upon clicking Finish , the user is shown the publish dialog and it is at this point the data is retrieved via TPT-API.



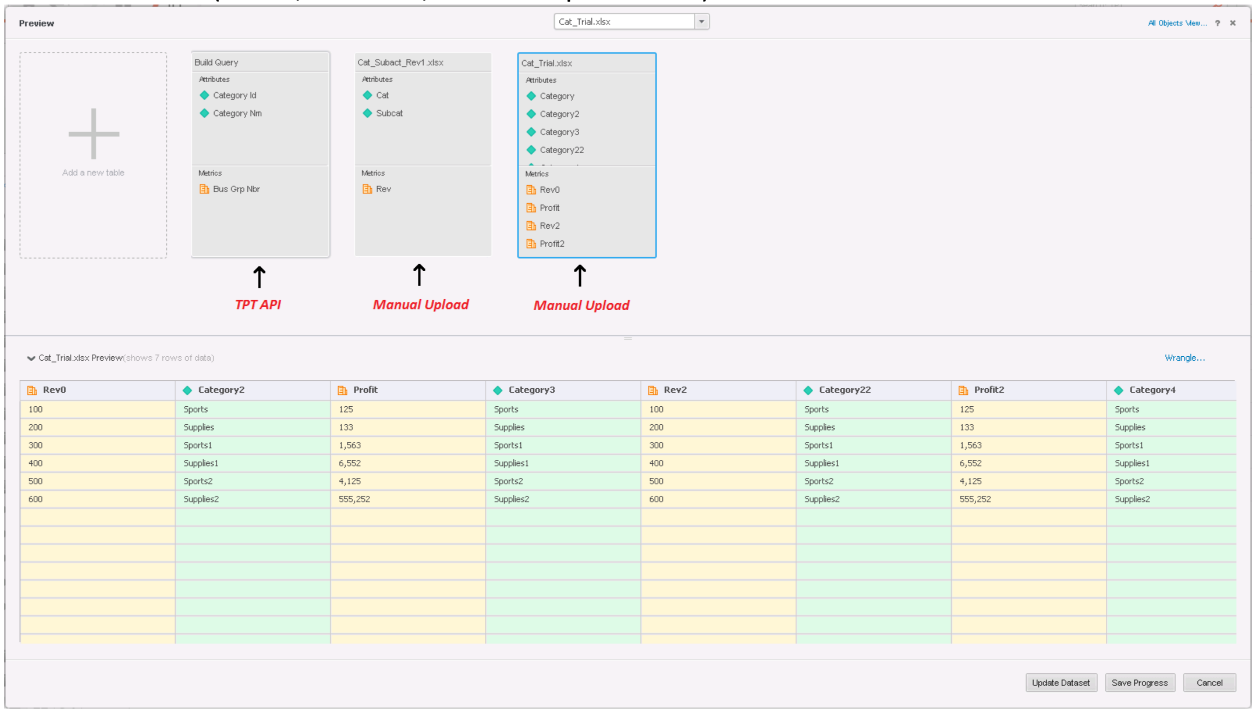
Build Query Basic Test Publishing Status

Data Source
Build Query

Build Query Preview(shows first 50 rows of data)

Category Id		Bus Grp Nbr
204	GRAI	2
606	CROI	5
125	CHRISTMAS CANDY, GUM & MINTS	4
120	HALLOWEEN CANDY, GUM & MINTS	4
210	AWAY FROM HOME FC COOKIE/CRACKER & MISC SNACKS	2
206	DELETE AWAY FROM HOME IMMEDIATE CONSUMPTION	2
705	PEPPERS	5
130	BULK CANDY, GUM & MINTS	4
405	JAMS/JELLIES/PRESERVES	5
410	PEANUT BUTTER & SPREADS	5
545	VINEGAR & COOKING WINE	5
825	FLAVORED WATER	2
202	AT HOME CRACKERS	2

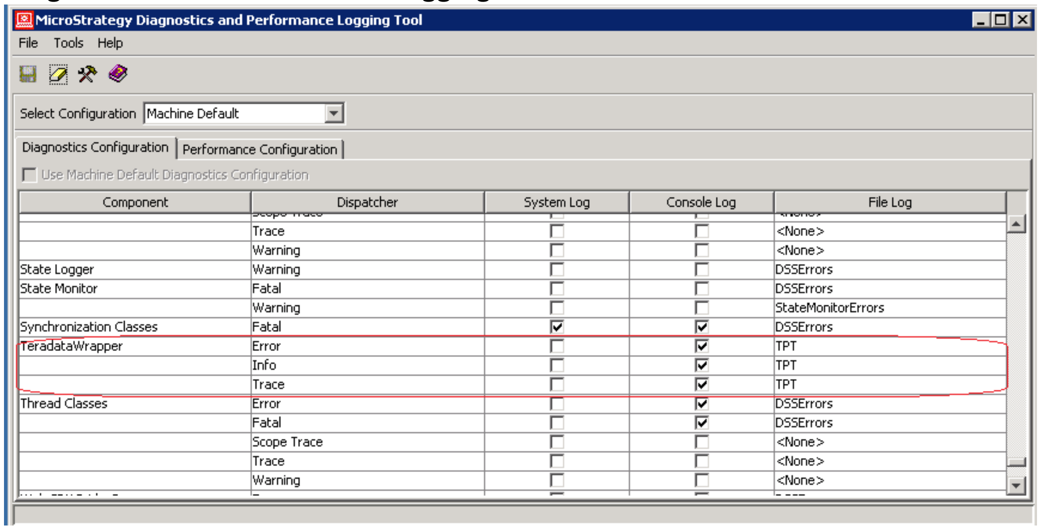
Note: In case the user is bringing in Multiple tables via the MTDI Interface then all the tables that are brought back from database instances that have TPT API setup will be brought back by TPTAPI while the rest will be brought back through the usual means as applicable to the individual table (ODBC, Web API, Manual Upload etc.)



Rev0	Category2	Profit	Category3	Rev2	Category22	Profit2	Category4
100	Sports	125	Sports	100	Sports	125	Sports
200	Supplies	133	Supplies	200	Supplies	133	Supplies
300	Sports1	1,563	Sports1	300	Sports1	1,563	Sports1
400	Supplies1	6,552	Supplies1	400	Supplies1	6,552	Supplies1
500	Sports2	4,125	Sports2	500	Sports2	4,125	Sports2
600	Supplies2	555,252	Supplies2	600	Supplies2	555,252	Supplies2

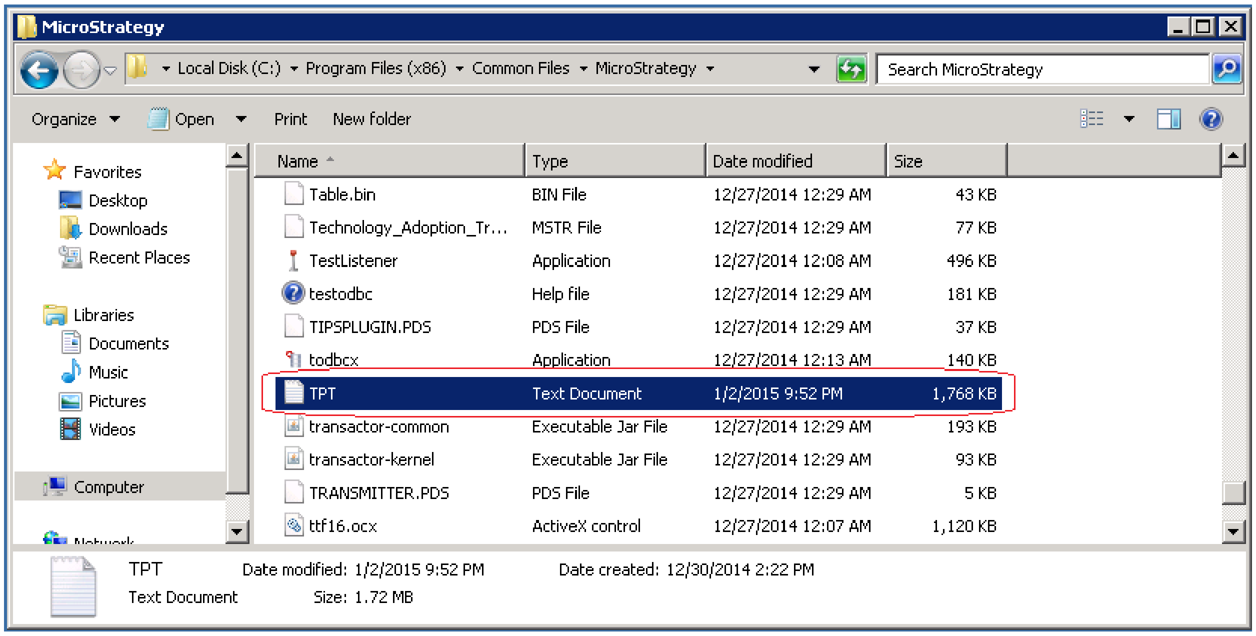
9) How do I generate logs for the TPTAPI Calls that are being made ?

The first step in the process is to turn on the flags for Teradata inside the MicroStrategy Diagnostics and Performance Logging tool.


MicroStrategy Diagnostics and Performance Logging Tool
File
Tools
Help
Select Configuration
Machine Default
Diagnostics Configuration
Performance Configuration
Use Machine Default Diagnostics Configuration

Component	Dispatcher	System Log	Console Log	File Log
	Trace	☐	☐	<None>
	Warning	☐	☐	<None>
State Logger	Warning	☐	☐	DSSErrors
State Monitor	Fatal	☐	☐	DSSErrors
	Warning	☐	☐	StateMonitorErrors
Synchronization Classes	Fatal	✓	✓	DSSErrors
TeradataWrapper	Error	☐	✓	TPT
	Info	☐	✓	TPT
	Trace	☐	✓	TPT
Thread Classes	Error	☐	✓	DSSErrors
	Fatal	☐	✓	DSSErrors
	Scope Trace	☐	☐	<None>
	Trace	☐	☐	<None>
	Warning	☐	☐	<None>

This setting would create a log file called TPT in the “C:\Program Files (x86)\Common Files\MicroStrategy” file location



Name	Type	Date modified	Size
Table.bin	BIN File	12/27/2014 12:29 AM	43 KB
Technology_Adoption_Tr...	MSTR File	12/27/2014 12:29 AM	77 KB
TestListener	Application	12/27/2014 12:08 AM	496 KB
testodbc	Help file	12/27/2014 12:29 AM	181 KB
TIPSPLUGIN.PDS	PDS File	12/27/2014 12:29 AM	37 KB
todbcx	Application	12/27/2014 12:13 AM	140 KB
TPT	Text Document	1/2/2015 9:52 PM	1,768 KB
transactor-common	Executable Jar File	12/27/2014 12:29 AM	193 KB
transactor-kernel	Executable Jar File	12/27/2014 12:29 AM	93 KB
TRANSMITTER.PDS	PDS File	12/27/2014 12:29 AM	5 KB
ttf16.ocx	ActiveX control	12/27/2014 12:07 AM	1,120 KB

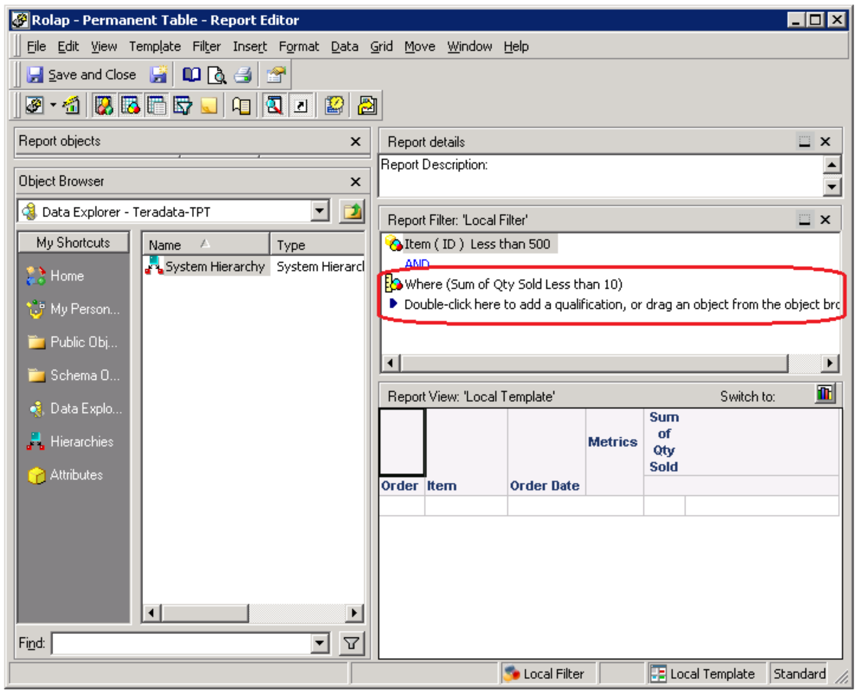
TPT Date modified: 1/2/2015 9:52 PM Date created: 12/30/2014 2:22 PM
Text Document Size: 1.72 MB

10) Since TPTAPI allows only a single select pass, What about Multi-Pass SQL ?

It is true that only a single select call can be made via TPTAPI when retrieving data from it. In case the MicroStrategy engine generates multi-pass SQL and TPTAPI flag is setup, then MicroStrategy handles this scenario by making all the non-select passes via ODBC and making the final select passes via TPTAPI.

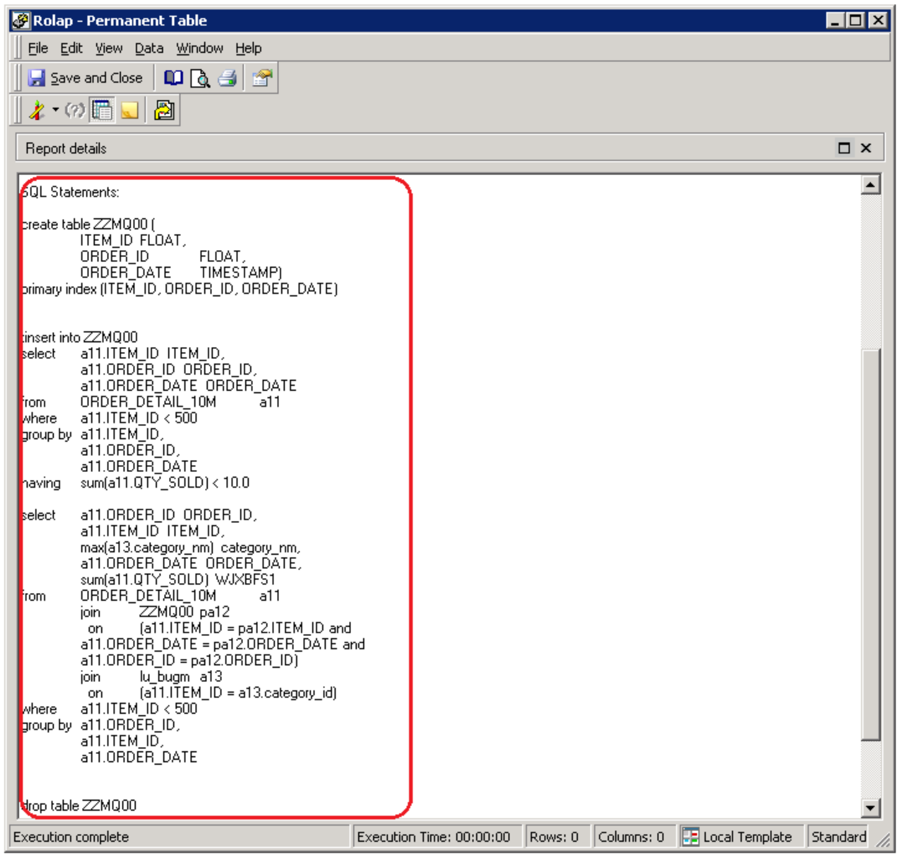
Here is an example:

1) Create a report with a metric qualification filter and this will result in Multi-pass SQL



Item (ID) Less than 500
AND
Where (Sum of Qty Sold Less than 10)
Double-click here to add a qualification, or drag an object from the object bro

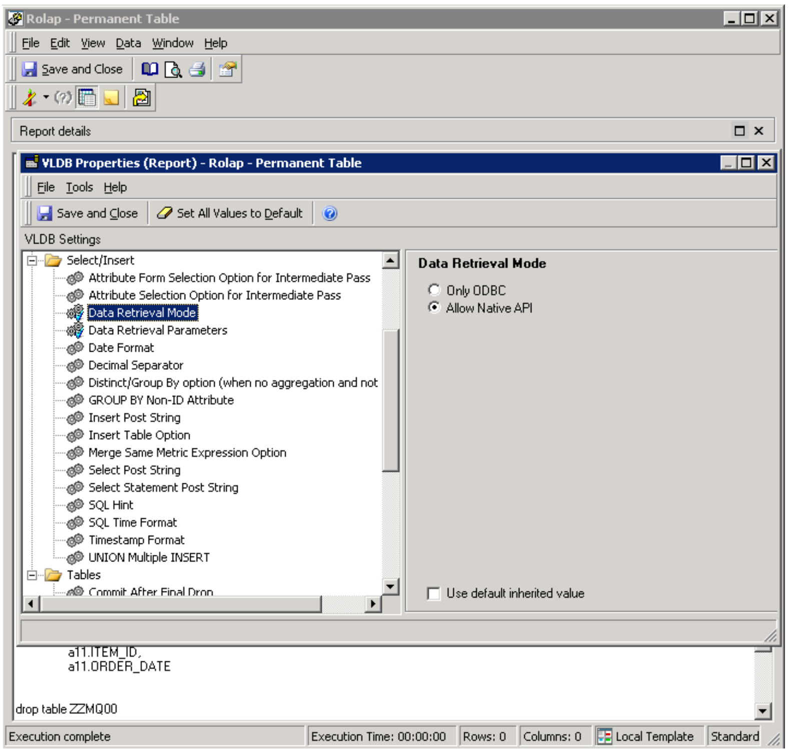
			Metrics	Sum of Qty Sold
Order	Item	Order Date		



```
SQL Statements:

create table ZZMQ00 (
        ITEM_ID FLOAT,
        ORDER_ID        FLOAT,
        ORDER_DATE      TIMESTAMP)
primary index (ITEM_ID, ORDER_ID, ORDER_DATE)


insert into ZZMQ00
select  a11.ITEM_ID  ITEM_ID,
        a11.ORDER_ID  ORDER_ID,
        a11.ORDER_DATE  ORDER_DATE
from    ORDER_DETAIL_10M        a11
where   a11.ITEM_ID < 500
group by a11.ITEM_ID,
        a11.ORDER_ID,
        a11.ORDER_DATE
having  sum(a11.QTY_SOLD) < 10.0

select  a11.ORDER_ID  ORDER_ID,
        a11.ITEM_ID  ITEM_ID,
        max(a13.category_nm)  category_nm,
        a11.ORDER_DATE  ORDER_DATE,
        sum(a11.QTY_SOLD)  WJXBFS1
from    ORDER_DETAIL_10M        a11
        join    ZZMQ00 pa12
          on    (a11.ITEM_ID = pa12.ITEM_ID and
        a11.ORDER_DATE = pa12.ORDER_DATE and
        a11.ORDER_ID = pa12.ORDER_ID)
        join    lu_bugm  a13
          on    (a11.ITEM_ID = a13.category_id)
where   a11.ITEM_ID < 500
group by a11.ORDER_ID,
        a11.ITEM_ID,
        a11.ORDER_DATE


drop table ZZMQ00
```

Execution complete | Execution Time: 00:00:00 | Rows: 0 | Columns: 0 | Local Template | Standard

2) Then turn on TPTAPI for the particular report and also set the intermediate table type to permanent table so that the intermediate tables created via ODBC are available for TPTAPI.


Rolap - Permanent Table
File Edit View Data Window Help
Save and Close
Report details
VLDB Properties (Report) - Rolap - Permanent Table
File Tools Help
Save and Close
Set All Values to Default
VLDB Settings
Select/Insert
Attribute Form Selection Option for Intermediate Pass
Attribute Selection Option for Intermediate Pass
Data Retrieval Mode
Data Retrieval Parameters
Date Format
Decimal Separator
Distinct/Group By option (when no aggregation and not
GROUP BY Non-ID Attribute
Insert Post String
Insert Table Option
Merge Same Metric Expression Option
Select Post String
Select Statement Post String
SQL Hint
SQL Time Format
Timestamp Format
UNION Multiple INSERT
Tables
Commit After Final Drop

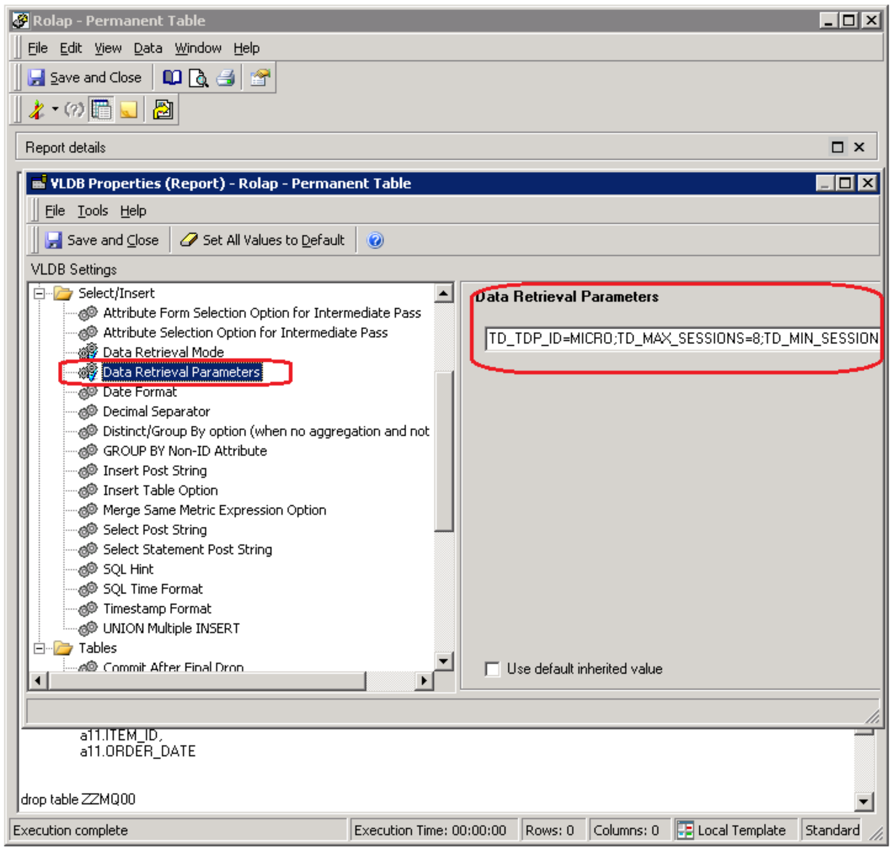
Data Retrieval Mode

Only ODBC [✓ Allow Native API]

☐ Use default inherited value

```
        a11.ITEM_ID,
        a11.ORDER_DATE

drop table ZZMQ00
```

Execution complete | Execution Time: 00:00:00 | Rows: 0 | Columns: 0 | Local Template | Standard



Rolap - Permanent Table

VLDB Properties (Report) - Rolap - Permanent Table

VLDB Settings

Select/Insert
- Attribute Form Selection Option for Intermediate Pass
- Attribute Selection Option for Intermediate Pass
- Data Retrieval Mode
- Data Retrieval Parameters
- Date Format
- Decimal Separator
- Distinct/Group By option (when no aggregation and not
- GROUP BY Non-ID Attribute
- Insert Post String
- Insert Table Option
- Merge Same Metric Expression Option
- Select Post String
- Select Statement Post String
- SQL Hint
- SQL Time Format
- Timestamp Format
- UNION Multiple INSERT

Tables
- Commit After Final Drop

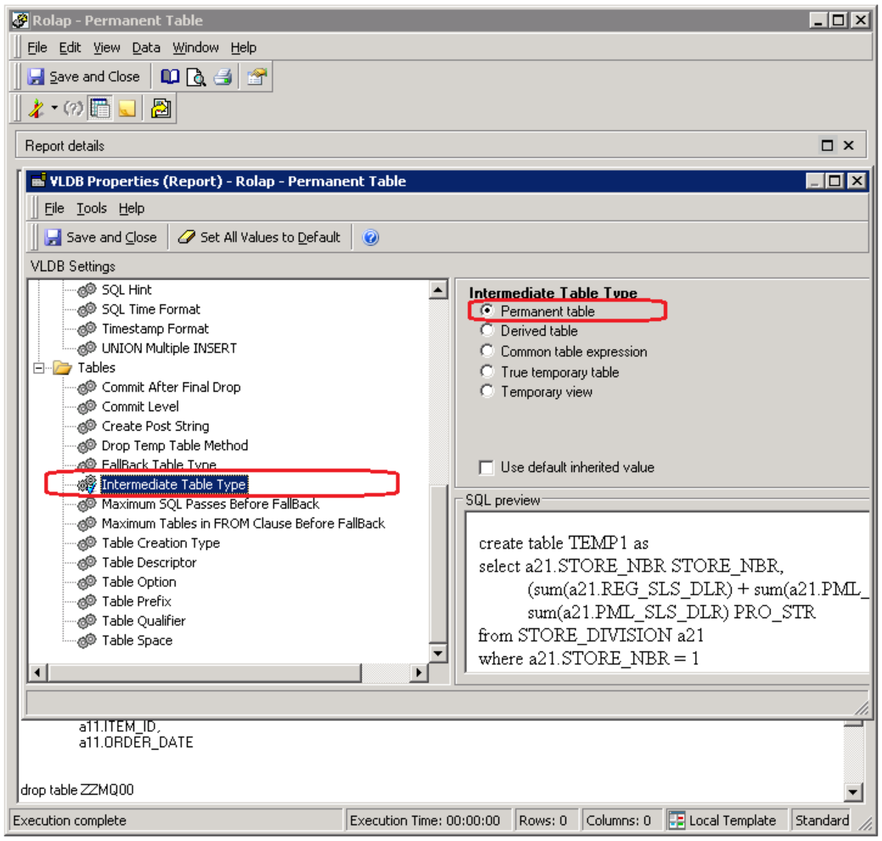
Data Retrieval Parameters

```
TD_TDP_ID=MICRO;TD_MAX_SESSIONS=8;TD_MIN_SESSION
```

Use default inherited value [ ]

```
a11.ITEM_ID,
a11.ORDER_DATE

drop table ZZMQ00
```

Execution complete | Execution Time: 00:00:00 | Rows: 0 | Columns: 0 | Local Template | Standard



Rolap - Permanent Table

VLDB Properties (Report) - Rolap - Permanent Table

VLDB Settings

- SQL Hint
- SQL Time Format
- Timestamp Format
- UNION Multiple INSERT

Tables
- Commit After Final Drop
- Commit Level
- Create Post String
- Drop Temp Table Method
- FallBack Table Type
- Intermediate Table Type
- Maximum SQL Passes Before FallBack
- Maximum Tables in FROM Clause Before FallBack
- Table Creation Type
- Table Descriptor
- Table Option
- Table Prefix
- Table Qualifier
- Table Space

Intermediate Table Type

- ● Permanent table
- ○ Derived table
- ○ Common table expression
- ○ True temporary table
- ○ Temporary view

Use default inherited value [ ]

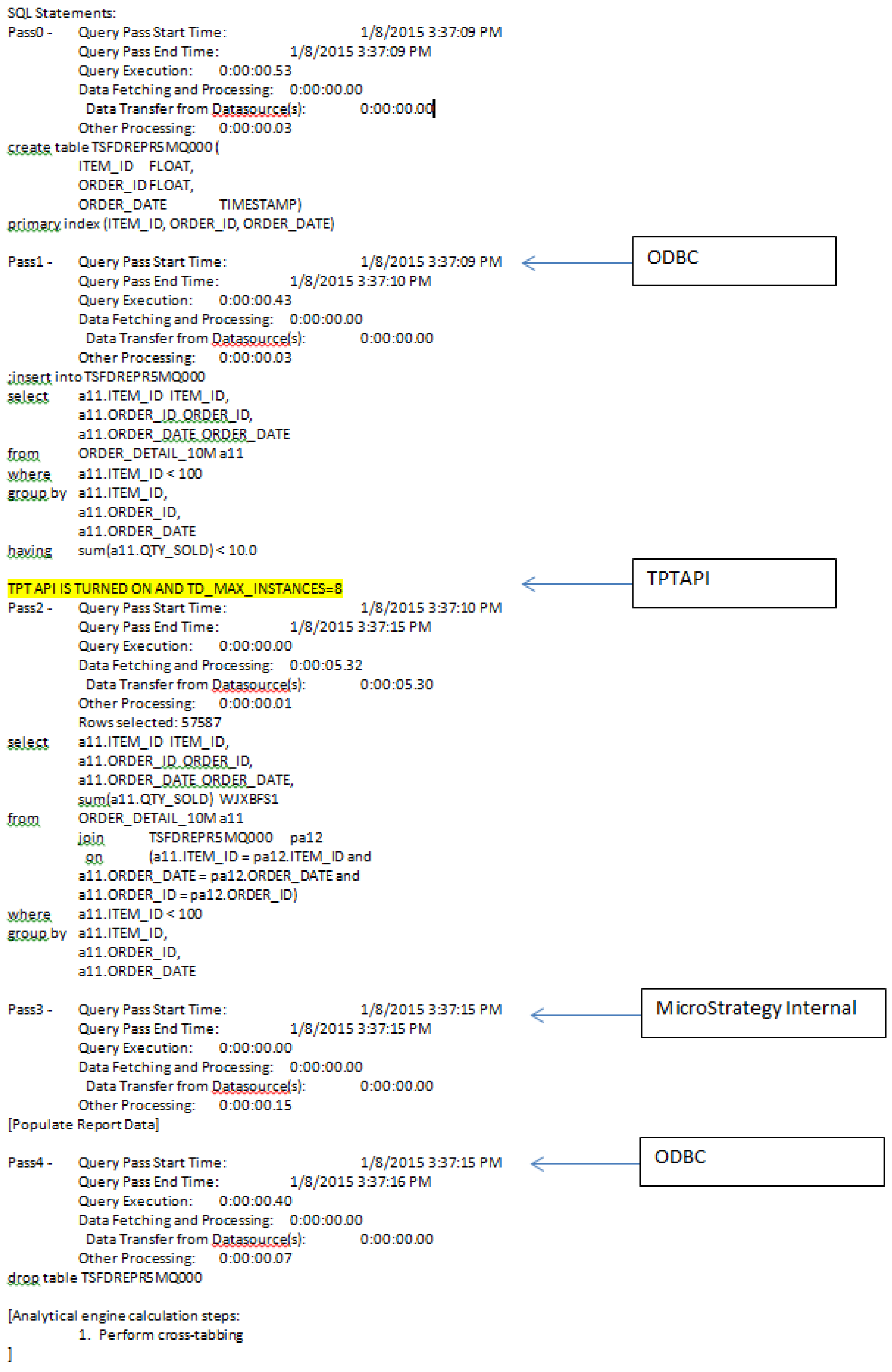
SQL preview

```
create table TEMP1 as
select a21.STORE_NBR STORE_NBR,
       (sum(a21.REG_SLS_DLR) + sum(a21.PML_
       sum(a21.PML_SLS_DLR) PRO_STR
from STORE_DIVISION a21
where a21.STORE_NBR = 1
```

```
a11.ITEM_ID,
a11.ORDER_DATE

drop table ZZMQ00
```

Execution complete | Execution Time: 00:00:00 | Rows: 0 | Columns: 0 | Local Template | Standard

3) Upon running the report, you will notice that the ‘Select Passes ‘are made with TPTAPI while the non-select passes are made via ODBC.



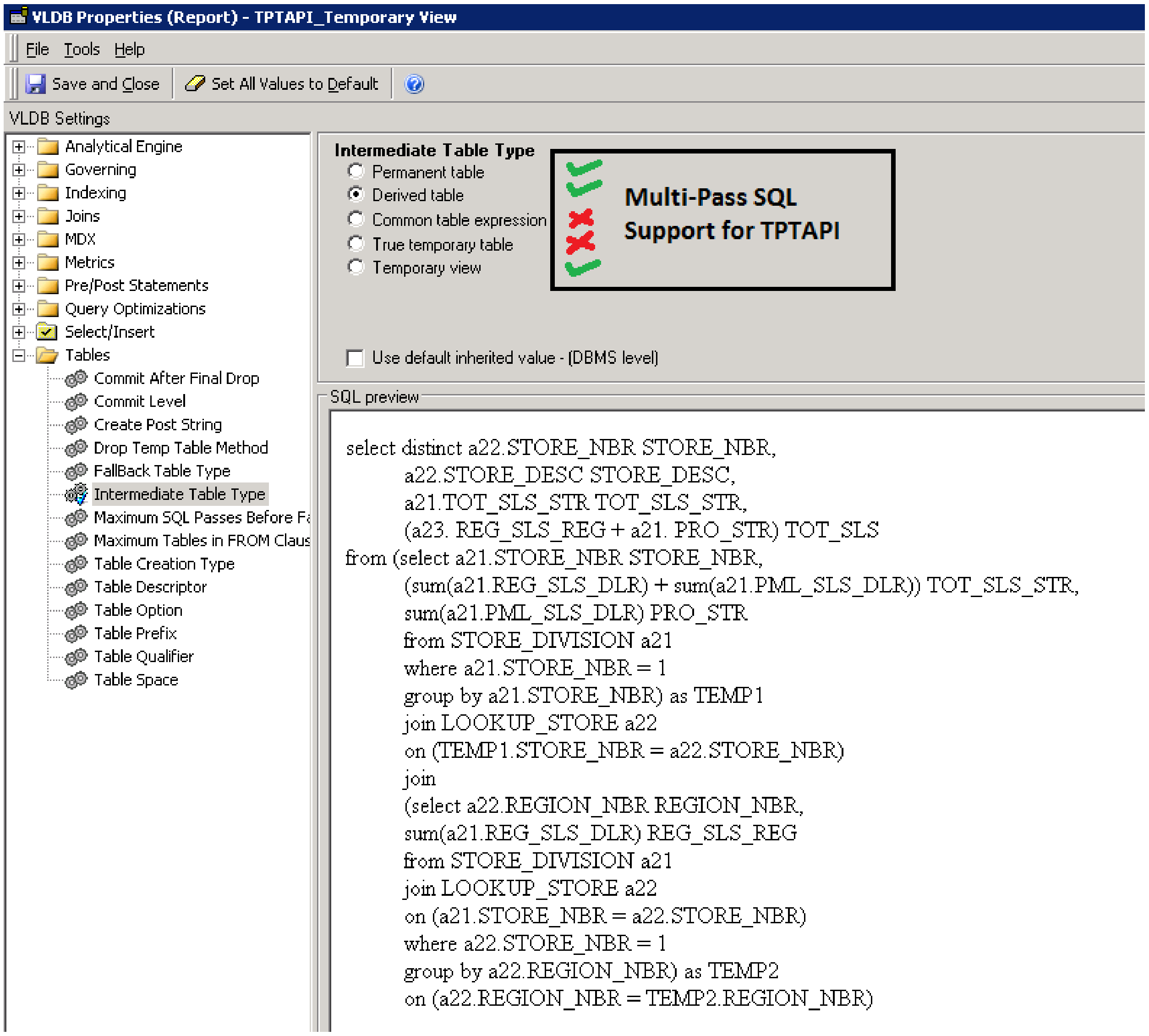
```
SQL Statements:
Pass0 -     Query Pass Start Time:              1/8/2015 3:37:09 PM
            Query Pass End Time:         1/8/2015 3:37:09 PM
            Query Execution:     0:00:00.53
            Data Fetching and Processing:    0:00:00.00
             Data Transfer from Datasource(s):          0:00:00.00
            Other Processing:    0:00:00.03
create table TSFDREPR5MQ000 (
            ITEM_ID    FLOAT,
            ORDER_ID FLOAT,
            ORDER_DATE          TIMESTAMP)
primary index (ITEM_ID, ORDER_ID, ORDER_DATE)

Pass1 -     Query Pass Start Time:              1/8/2015 3:37:09 PM
            Query Pass End Time:         1/8/2015 3:37:10 PM
            Query Execution:     0:00:00.43
            Data Fetching and Processing:    0:00:00.00
             Data Transfer from Datasource(s):          0:00:00.00
            Other Processing:    0:00:00.03
insert into TSFDREPR5MQ000
select      a11.ITEM_ID  ITEM_ID,
            a11.ORDER_ID  ORDER_ID,
            a11.ORDER_DATE  ORDER_DATE
from        ORDER_DETAIL_10M a11
where       a11.ITEM_ID < 100
group by    a11.ITEM_ID,
            a11.ORDER_ID,
            a11.ORDER_DATE
having      sum(a11.QTY_SOLD) < 10.0

TPT API IS TURNED ON AND TD_MAX_INSTANCES=8
Pass2 -     Query Pass Start Time:              1/8/2015 3:37:10 PM
            Query Pass End Time:         1/8/2015 3:37:15 PM
            Query Execution:     0:00:00.00
            Data Fetching and Processing:    0:00:05.32
             Data Transfer from Datasource(s):          0:00:05.30
            Other Processing:    0:00:00.01
            Rows selected: 57587
select      a11.ITEM_ID  ITEM_ID,
            a11.ORDER_ID  ORDER_ID,
            a11.ORDER_DATE  ORDER_DATE,
            sum(a11.QTY_SOLD)  WJXBFS1
from        ORDER_DETAIL_10M a11
            join        TSFDREPR5MQ000    pa12
             on         (a11.ITEM_ID = pa12.ITEM_ID and
            a11.ORDER_DATE = pa12.ORDER_DATE and
            a11.ORDER_ID = pa12.ORDER_ID)
where       a11.ITEM_ID < 100
group by    a11.ITEM_ID,
            a11.ORDER_ID,
            a11.ORDER_DATE

Pass3 -     Query Pass Start Time:              1/8/2015 3:37:15 PM
            Query Pass End Time:         1/8/2015 3:37:15 PM
            Query Execution:     0:00:00.00
            Data Fetching and Processing:    0:00:00.00
             Data Transfer from Datasource(s):          0:00:00.00
            Other Processing:    0:00:00.15
[Populate Report Data]

Pass4 -     Query Pass Start Time:              1/8/2015 3:37:15 PM
            Query Pass End Time:         1/8/2015 3:37:16 PM
            Query Execution:     0:00:00.40
            Data Fetching and Processing:    0:00:00.00
             Data Transfer from Datasource(s):          0:00:00.00
            Other Processing:    0:00:00.07
drop table TSFDREPR5MQ000

[Analytical engine calculation steps:
            1.  Perform cross-tabbing
]
```

11) For Multi-Pass SQL, What intermediate table types can be used ?

In the case of Teradata, the following intermediate table type settings are appropriate

1) Derived Table (Default): This is the default and it avoids creating intermediate tables by using the derived table syntax and this setting only generates a single pass, it fits best with TPTAPI.
2) Permanent Table: If the SQL generates intermediate passes, then the SQL engine will make the NON-SELECT passes via ODBC and these NON-SELECT passes would create permanent tables (with this setting) and these are accessible to the SELECT pass that is made via TPTAPI.
3) Temporary View: In this case, the intermediate passes would be created as views and these views are accessible to the SELECT pass that is made via TPTAPI.

The following Intermediate Table Types are NOT Supported with TPTAPI setting: CTE and True Temp Table: Choosing these settings would create errors as the temporary tables that are created in the non-select passes that are made via ODBC are not accessible via TPTAPI. CTE is not a recommended Intermediate Table Type for Teradata under any interface connectivity option.



Intermediate Table Type
- ○ Permanent table ✓
- ● Derived table ✓
- ○ Common table expression ✗
- ○ True temporary table ✗
- ○ Temporary view ✓

☐ Use default inherited value - (DBMS level)

```
select distinct a22.STORE_NBR STORE_NBR,
        a22.STORE_DESC STORE_DESC,
        a21.TOT_SLS_STR TOT_SLS_STR,
        (a23. REG_SLS_REG + a21. PRO_STR) TOT_SLS
from (select a21.STORE_NBR STORE_NBR,
        (sum(a21.REG_SLS_DLR) + sum(a21.PML_SLS_DLR)) TOT_SLS_STR,
        sum(a21.PML_SLS_DLR) PRO_STR
        from STORE_DIVISION a21
        where a21.STORE_NBR = 1
        group by a21.STORE_NBR) as TEMP1
        join LOOKUP_STORE a22
        on (TEMP1.STORE_NBR = a22.STORE_NBR)
        join
        (select a22.REGION_NBR REGION_NBR,
        sum(a21.REG_SLS_DLR) REG_SLS_REG
        from STORE_DIVISION a21
        join LOOKUP_STORE a22
        on (a21.STORE_NBR = a22.STORE_NBR)
        where a22.STORE_NBR = 1
        group by a22.REGION_NBR) as TEMP2
        on (a22.REGION_NBR = TEMP2.REGION_NBR)
```

12) What TPTAPI parameters can I customize via the Data Retrieval Parameters VLDB settings?

The TPT Application Programming Interface Programmer's Guide http://www.info.teradata.com/edownload.cfm?itemid=130420033 lists out all the parameters that can be passed on to the TPTAPI. Of these parameters, we only allow certain parameters to be input by the user as shown below.

Requirement	Attribute	Datatype	Notes	Can be changed by User ?
Required	TD_BUFFER_MODE	varchar		No
Required	TD_INSTANCE_NUM	integer		No

Required	TD_LOGSQL	varchar		No
Required	TD_MAX_INSTANCES			Yes
Required	TD_RESTARTMODE	integer		No
Required	TD_SELECT_STMT	varchar		No
Required	TD_SYSTEM_OPERATOR	varchar		No
Required	TD_USER_NAME	varchar	if not provided, TPT wrapper uses username and user pwd from dblogin	Yes
Required	TD_USER_PASSWORD	varchar	if not provided, TPT wrapper uses username and user pwd from dblogin	Yes
Optional	TD_ACCOUNT_ID	varchar		Yes
Optional	TD_AUTORESTART	varchar		No
Optional	TD_BLOCK_SIZE	integer		Yes
Optional	TD_BUFFER_HEADER_SIZE	integer		No
Optional	TD_BUFFER_LENGTH_SIZE	integer		No
Optional	TD_BUFFER_MAX_SIZE	integer		No
Optional	TD_BUFFER_TRAILER_SIZE	integer		No
Optional	TD_CHARSET	varchar	The users need not care about TD_CHARSET parameter, as behind we always set the session character to UTF16, the export width for each CHARacter is 2 bytes no matter it is ASCII or UNICODE column.	No
Optional	TD_DATA_ENCRYPTION	varchar		Yes
Optional	TD_DATE_FORM	varchar	Devopler rule it to the default Interger type, we will pare the interger type to inner DataTime object	No
Optional	TD_IGNORE_MAX_DECIMAL_DIGITS	varchar		Yes
Optional	TD_LOGON_MECH	varchar		Yes
Optional	TD_LOGON_MECH_DATA	varchar		Yes
Optional	TD_MAX_DECIMAL_DIGITS	integer		Yes
Optional	TD_MAX_SESSIONS	integer		Yes
Optional	TD_MIN_SESSIONS	integer		Yes

Optional	TD_MSG_ENCODING	TD_Encoding		No.
Optional	TD_NOTIFY_EXIT	varchar		No
Optional	TD_NOTIFY_LEVEL	varchar		No
Optional	TD_NOTIFY_METHOD	varchar		No
Optional	TD_NOTIFY_STRING	varchar		No
Optional	TD_OUTLIMIT	integer		No
Optional	TD_QUERY_BAND_SESS_INFO	varchar		Yes
Optional	TD_SPOOLMODE	varchar	TPT wrapper set default to "NoSpool"	Yes
Optional	TD_TDP_ID	varchar	According to the manual, TD_TDP_ID is an optional parameter. If not specified, it will use the default TptId established for the user by system administrator. In my tests I don't provide this parameter, then it will throw an error and the log message is like: PC_GETDEFAULTTDPID: default TdpId: 'dbc' … PC_ISSUEDBCHQE: using TDP ID for this query PC_ISSUEDBCHQE: my_cli->TdpId: 'dbc' PC_ISSUEDBCHQE: calling DBCHQE for TDP: 'dbc' …. **** 04:18:41 TPT10551: CLI '224' occurred while connecting to the RDBMS EXPORT_OPERATOR: TPT10507: CLI Error 224: MTDP: EM_NOHOST(224): name not in HOSTS file or names database. **** 04:18:41 TPT10507: CLI Error 224: MTDP: EM_NOHOST(224): name not in HOSTS file or names database. it says the default Tdpid is not in the names list, so it seems the DBA doesn't create that TptId for the user, in this case, TD_TDP_ID is required for user input.	Yes
Optional	TD_TENACITY_HOURS	integer		Yes
Optional	TD_TENACITY_SLEEP	integer		Yes

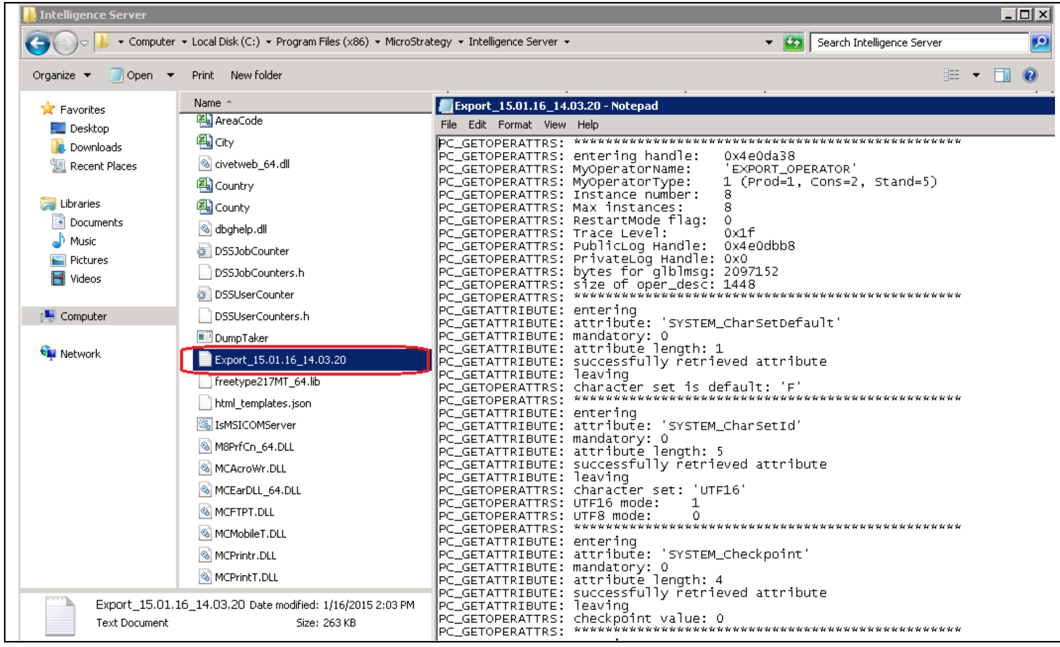
Optional	TD_TRACE_LEVEL	integer	The TraceLevel attribute is an internal diagnostic aid. Use only if instructed to by Teradata support. TD_OFF should always be specified.
			Please use the below numbers instead of the ENUMs
			"TD_OFF = 1, // DR89698 //
			TD_OPER =2
			TD_OPER_CLI = 3
			TD_OPER_NOTIFY =4
			TD_OPER_OPCOMMON =5
			TD_OPER_SPECIAL =6,//Special
			TD_OPER_ALL = 7,
			TD_GENERAL,
			TD_ROW,
			TD_MSG_CODES"
				Yes

Optional	TD_TRACE_OUTPUT	varchar		Yes
Optional	TD_WORKINGDATABASE	varchar		Yes

13) Examples of how the Data retrieval parameters could be used ?

Here are some example of how the data retrieval parameters can be set and the corresponding benefits

TD_TRACE_LEVEL PLEASE NOTE: TRACING should ONLY be enabled if requested by either MicroStrategy or Teradata Support Center personnel. Logs should be the primary general information repository to review runtime statistics for each TPTAPI job.
Data Retrieval Parameters *TD_TDP_ID=micro;TD_MAX_SESSIONS=8;TD_MIN_SESSIONS=8;TD_MAX_INSTANCES=8;TD_TRACE_LEVEL=7;*
Creates a Trace file in the folder containing the TPT code. In the case of windows, this is the following folder {C:\Program Files (x86)\MicroStrategy\Intelligence Server}



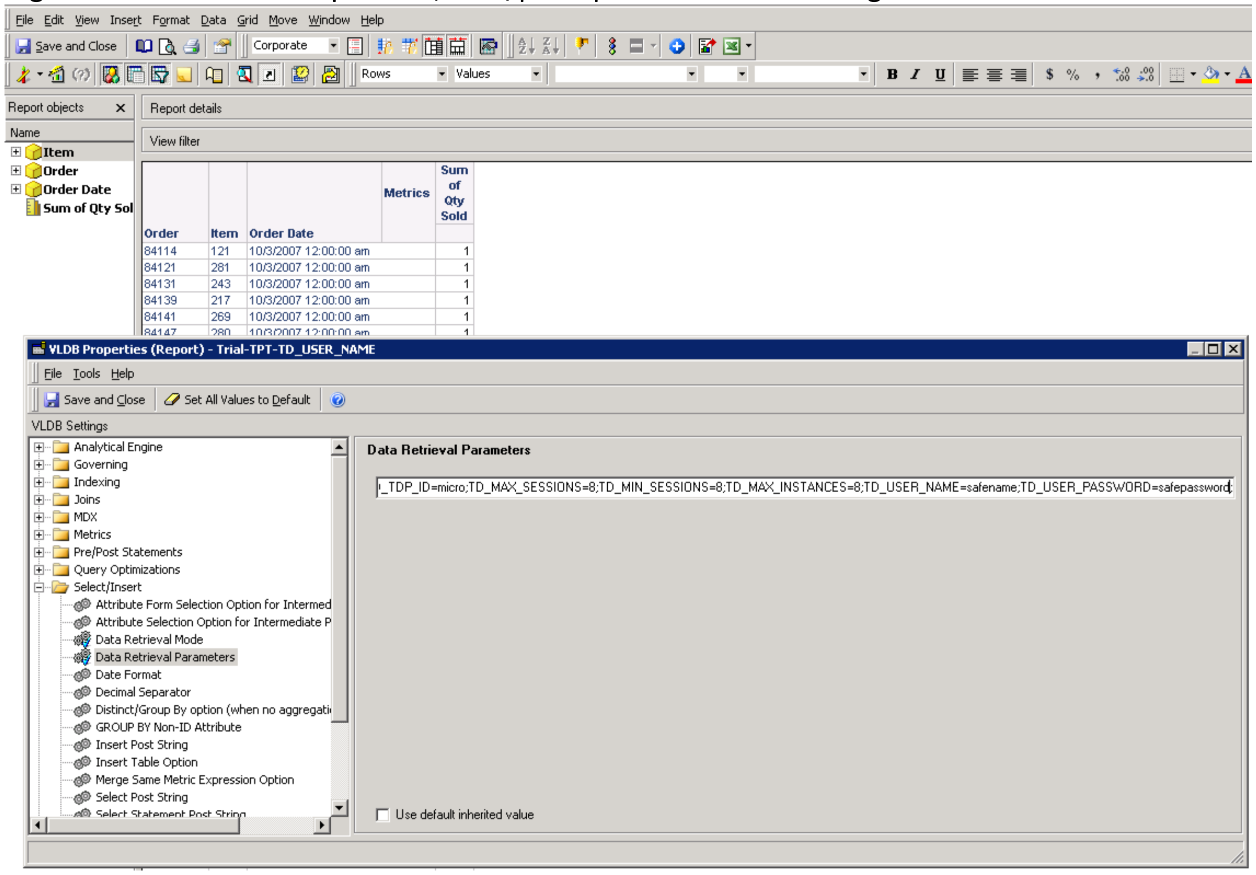
```
PC_GETOPERATTRS: ****************************************************
PC_GETOPERATTRS: entering handle:   0x4e0da38
PC_GETOPERATTRS: MyOperatorName:    'EXPORT_OPERATOR'
PC_GETOPERATTRS: MyOperatorType:    1 (Prod=1, Cons=2, Stand=5)
PC_GETOPERATTRS: Instance number:   8
PC_GETOPERATTRS: Max instances:     8
PC_GETOPERATTRS: RestartMode flag:  0
PC_GETOPERATTRS: Trace Level:       0x1f
PC_GETOPERATTRS: PublicLog Handle:  0x4e0dbb8
PC_GETOPERATTRS: PrivateLog Handle: 0x0
PC_GETOPERATTRS: bytes for glblmsg: 2097152
PC_GETOPERATTRS: size of oper_desc: 1448
PC_GETOPERATTRS: ****************************************************
PC_GETATTRIBUTE: entering
PC_GETATTRIBUTE: attribute: 'SYSTEM_CharSetDefault'
PC_GETATTRIBUTE: mandatory: 0
PC_GETATTRIBUTE: attribute length: 1
PC_GETATTRIBUTE: successfully retrieved attribute
PC_GETATTRIBUTE: leaving
PC_GETOPERATTRS: character set is default: 'F'
PC_GETOPERATTRS: ****************************************************
PC_GETATTRIBUTE: entering
PC_GETATTRIBUTE: attribute: 'SYSTEM_CharSetId'
PC_GETATTRIBUTE: mandatory: 0
PC_GETATTRIBUTE: attribute length: 5
PC_GETATTRIBUTE: successfully retrieved attribute
PC_GETATTRIBUTE: leaving
PC_GETOPERATTRS: character set: 'UTF16'
PC_GETOPERATTRS: UTF16 mode:    1
PC_GETOPERATTRS: UTF8 mode:     0
PC_GETOPERATTRS: ****************************************************
PC_GETATTRIBUTE: entering
PC_GETATTRIBUTE: attribute: 'SYSTEM_Checkpoint'
PC_GETATTRIBUTE: mandatory: 0
PC_GETATTRIBUTE: attribute length: 4
PC_GETATTRIBUTE: successfully retrieved attribute
PC_GETATTRIBUTE: leaving
PC_GETOPERATTRS: checkpoint value: 0
PC_GETOPERATTRS: ****************************************************
```

TD_USER_NAME and TD_USER_PASSWORD

Data Retrieval Parameters

TD_TDP_ID=micro;TD_MAX_SESSIONS=8;TD_MIN_SESSIONS=8;TD_MAX_INSTANCES=8;
TD_USER_NAME=safeusername;TD_USER_PASSWORD=safepassword;

Allows users to set a specific username and password for TPTAPI that is different from the ODBC login credentials. If not specified, user/pw is passed from ODBC login.



VLDB Properties (Report) - Trial-TPT-TD_USER_NAME

Order	Item	Order Date	Metrics / Sum of Qty Sold
84114	121	10/3/2007 12:00:00 am	1
84121	281	10/3/2007 12:00:00 am	1
84131	243	10/3/2007 12:00:00 am	1
84139	217	10/3/2007 12:00:00 am	1
84141	269	10/3/2007 12:00:00 am	1
84147	280	10/3/2007 12:00:00 am	1

Data Retrieval Parameters

```
_TDP_ID=micro;TD_MAX_SESSIONS=8;TD_MIN_SESSIONS=8;TD_MAX_INSTANCES=8;TD_USER_NAME=safename;TD_USER_PASSWORD=safepassword;
```

TD_QUERY_BAND_SESSION_INFO

Data Retrieval Parameters
TD_TDP_ID=micro;TD_MAX_SESSIONS=8;TD_MIN_SESSIONS=8;TD_MAX_INSTANCES=8;
TD_QUERY_BAND_SESS_INFO={a=1;b=2;c=3;d=4;};

Passes that query band along with the TPTAPI PLEASE NOTE: The Attribute Value format for the QueryBand TPTAPI Attribute differs from the Teradata TPTAPI documentation in that the VALUE portion must be enclosed with curly brackets or braces **{ }**. Refer to example above.

14) What are some of the other TPTAPI Usage Considerations ?

Key to optimizing TPTAPI throughput is utilizing the "multiple instance" and "multiple session" capabilities. There are no hard and fast rules with setting these, as each environment has unique capabilities, constraints, and bottlenecks. Testing with TPTAPI-script based jobs with trial and error modification of INSTANCES and SESSION settings may yield data points to help tweak the optimal settings for a particular environment. Alternatively, test with MicroStrategy, reviewing MB/sec results in logs. A possible starting point in understanding this is to review the TPT User Guide (Optimizing Job Performance with Sessions and Instances) in Chapter 4:

Teradata Database Effects on Job Scripts. A sample approach might be to start with a single instance or 2, and session count = amp count or ampcount/2 then watch top (linux) and/or examine the tlogview output after the job completes. If an instance is >70% busy or so, add another always ensuring the session count is an even multiple. Windows Performance Monitor can be used to assess network bandwidth consumption.

Be aware of TASM Utility Rules. FastExport job default is 4 sessions. If you want to run >4 sessions (i.e., the Max Sessions TPTAPI Attribute is executed), a TASM Utility Rule will need to be created in Viewpoint based on User or QueryBand.

Below is a sample workflow of how a QueryBand-enabled Utility Rule can be implemented via Viewpoint:

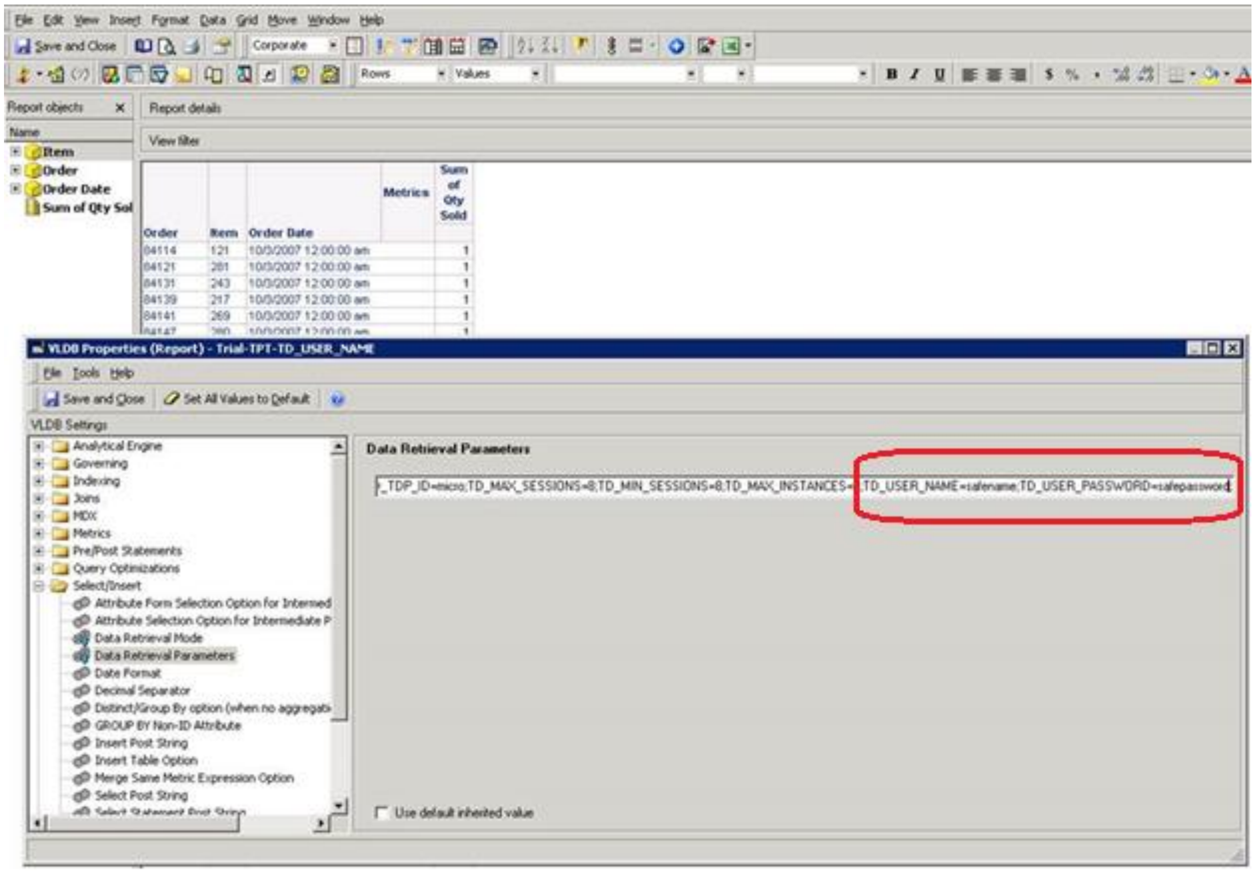
In Viewpoint: Workload Designer. >> Sessions.
Then click on: Utility Sessions.
Clone a Rule (e.g., FastExport Rule) – give it a name.
There's an area to modify # Sessions. (Export defaults to 4. Max allowed is 96).
Then there's another tab called: CLASSIFICATION.
There you can put in a QueryBand classification. I made a QueryBand up called MSTR_IF_TYPE.
And added a Value of:TPTAPI.
So, if it sees a Queryband: 'MSTR_IF_TYPE=TPTAPI'; it will apply XX sessions instead of 4.
MAKE READY, then ACTIVATE

15) In MicroStrategy 10 release, which products support TPTAPI?

TPTAPI is available only for the Enterprise Web and Enterprise Developer products. TPTAPI is not available for MicroStrategy Desktop for Analysts. This is mainly because desktop analysts are expected to use the software on their local desktops which have limited memories and that is not quite the intended application for TPTAPI which has been built to import large cubes.

16) How does authentication work for TPTAPI?

When you use tptapi, you are setting a VLDB setting called data retreival parameters and two of these parameters are TD_USER_NAME and TD_USER_PASSWORD.

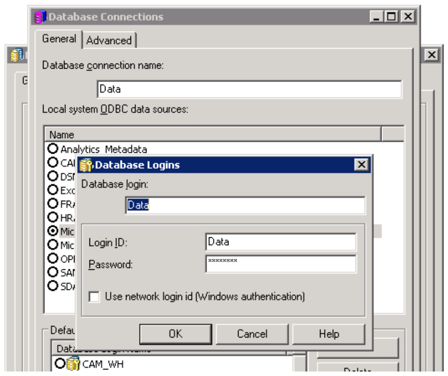

VLDB Properties (Report) - Trial-TPT-TD_USER_NAME

Data Retrieval Parameters

```
_TDP_ID=micro;TD_MAX_SESSIONS=8;TD_MIN_SESSIONS=8;TD_MAX_INSTANCES=8;TD_USER_NAME=safename;TD_USER_PASSWORD=safepassword
```

Order	Item	Order Date	Sum of Qty Sold
84114	121	10/3/2007 12:00:00 am	1
84121	281	10/3/2007 12:00:00 am	1
84131	243	10/3/2007 12:00:00 am	1
84139	217	10/3/2007 12:00:00 am	1
84141	269	10/3/2007 12:00:00 am	1

Now the user can set them to what ever they want and if they do set it then these are the credentials that are used for authentication against the teradata database. However, if the user doesnot provide these parameters (TD_USER_NAME and TD_USER_PASSWORD) then those values are extracted out of the ODBC connection string (taken from database login) Values provided in the 'Database Logins' interface are used to populate TD_USER_NAME and TD_USER_PASSWORD.


Database Connections — Database connection name: Data

Database Logins — Database login: Data; Login ID: Data; Password: ********

General
Advanced
Local system ODBC data sources:
Analytics Metadata
Use network login id (Windows authentication)
OK
Cancel
Help
CAM_WH

16.1) Does TPTAPI support LDAP authentication

Ans) The answer is yes. TPTAPI has a parameter called TD_LOGON_MECH and we need to set that to LDAP.

For example the data retrieval parameters vldb setting could be set as follows:

TD_TDP_ID=micro;TD_MAX_SESSIONS=8;TD_MIN_SESSIONS=8;TD_MAX_INSTANCES=8;TD_TRACE_LEVEL=7;TD_LOGON_MECH=LDAP;

However, we haven’t yet tested to see if this scenario is working.

16.2) Does TPTAPI support MicroStrategy warehouse pass-thru credentials ?
Ans) The answer to this question would also be yes. As we extract the credentials from the connection string and the connection string to have the warehouse pass-through credentials. However, we haven’t tested out this scenario and verified this.