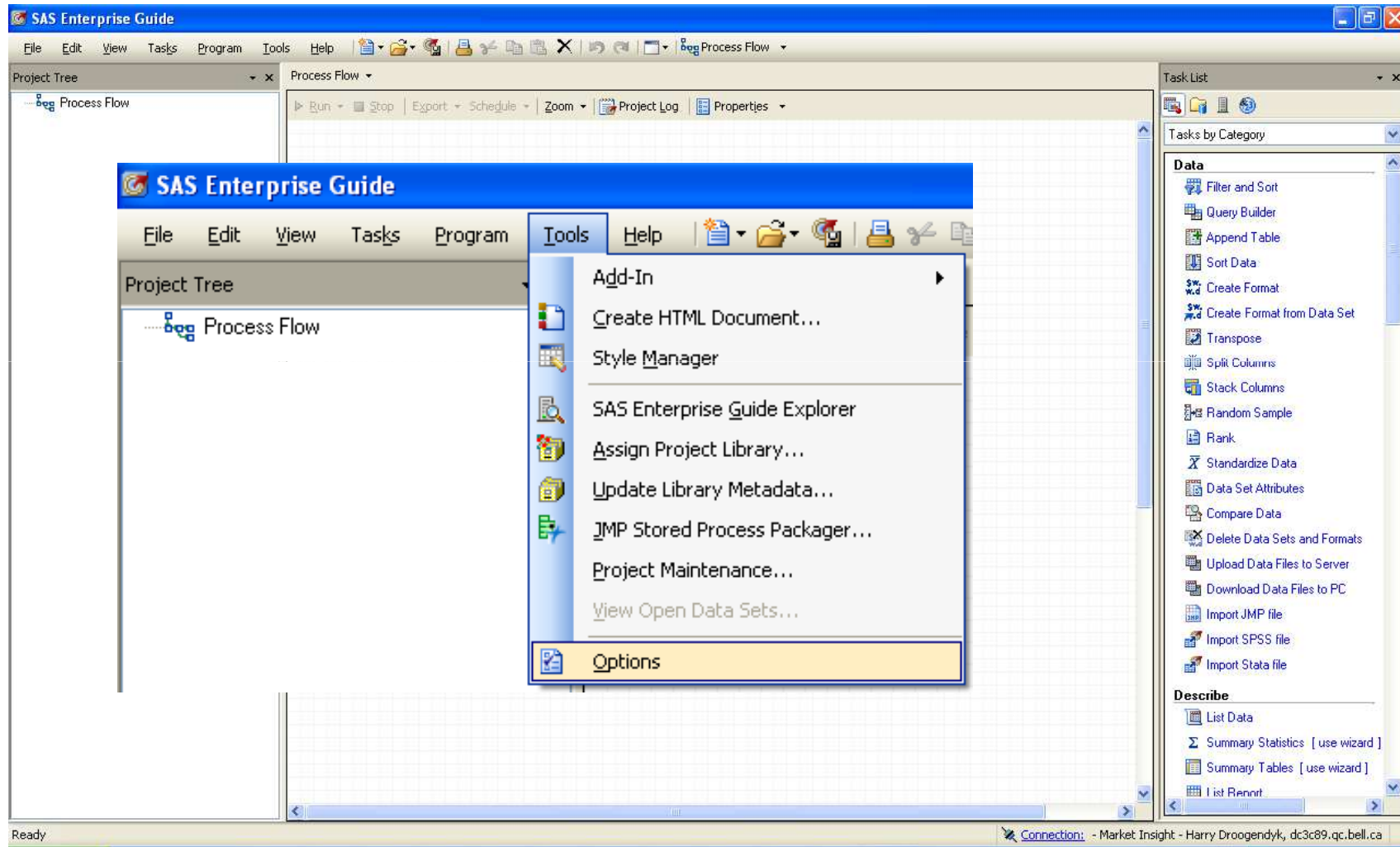


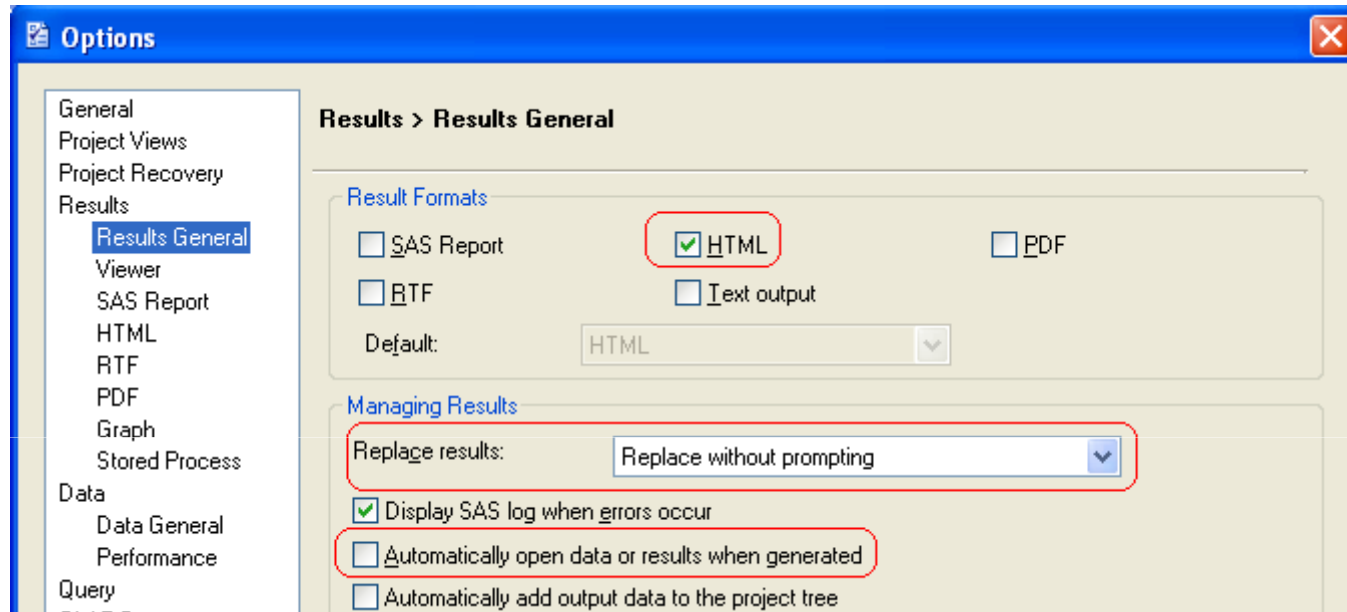
Using SAS Enterprise Guide Effectively

Who Does What Where ?

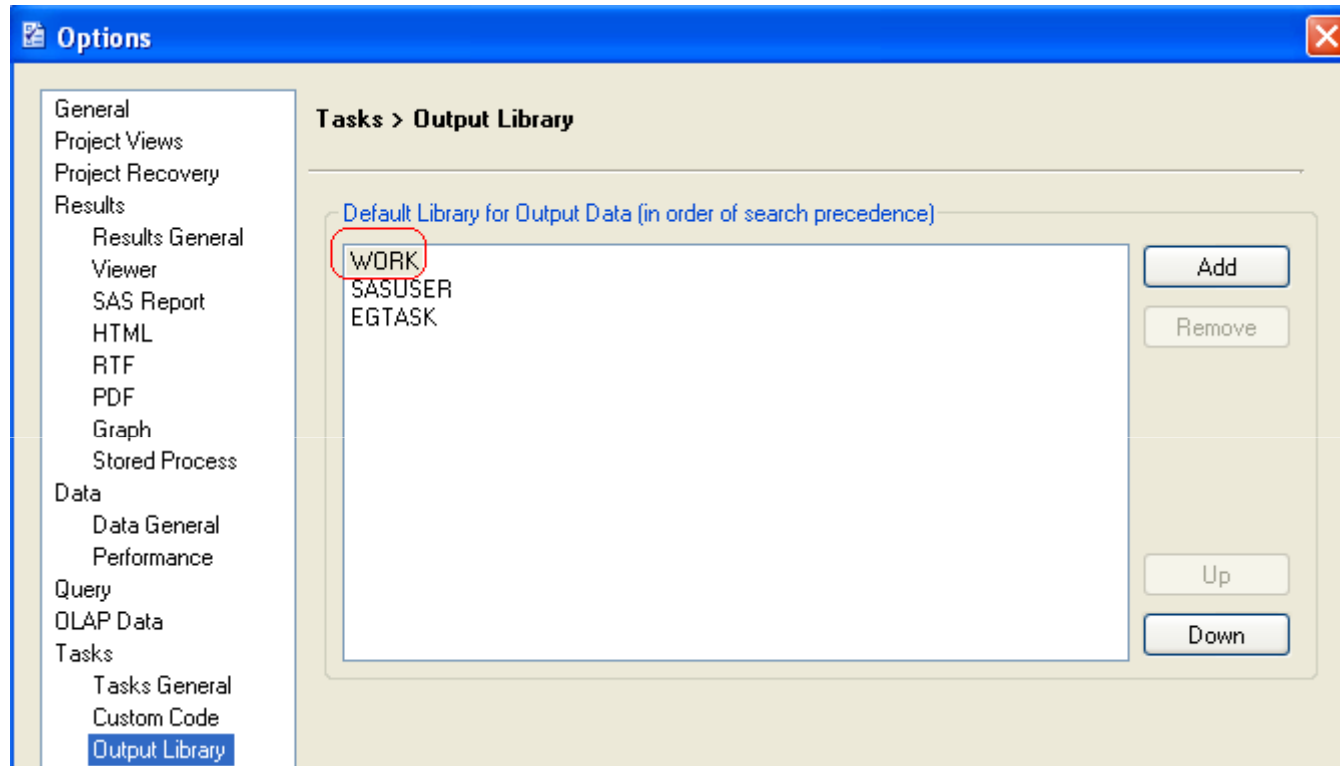
Now What ?



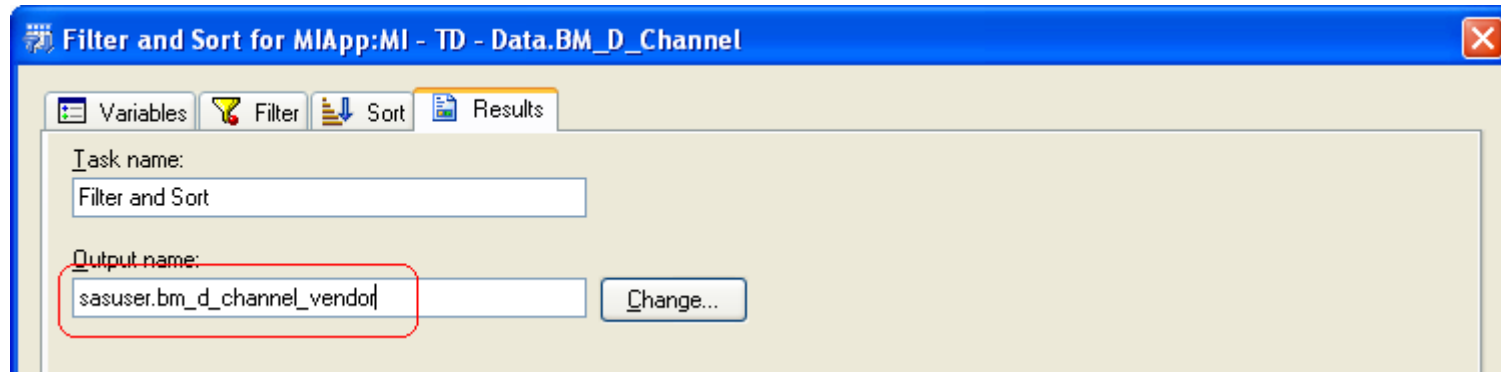
EG – Default Behaviour



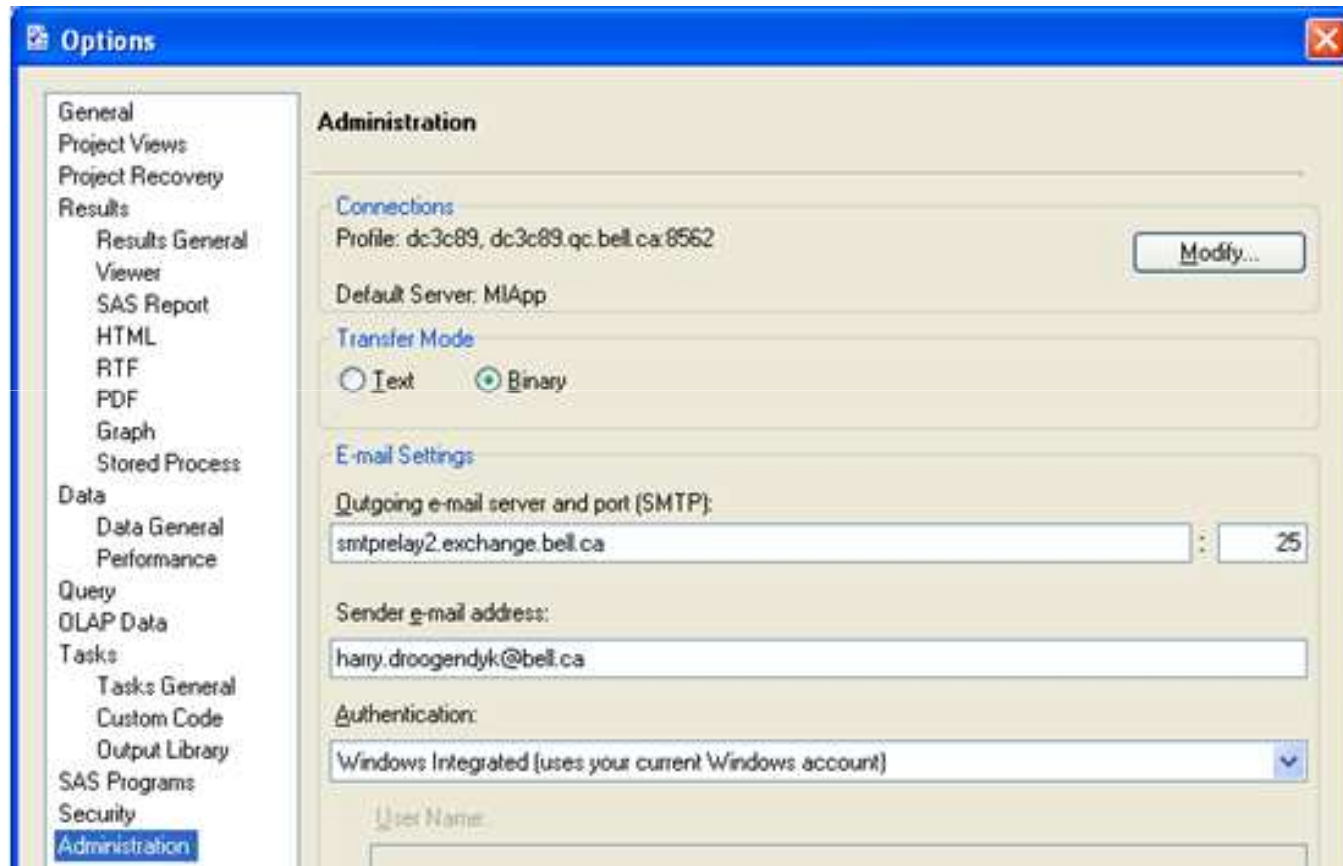
EG – Default Setup



EG – SASUSER



EG Profiles



EG Profiles

The screenshot displays the SAS Connections dialog box with the 'EG 4.3 Pre-Prod Properties' window open. The 'Profiles' list in the background shows 'dc2c1s' and 'dc3c89', with 'dc3c89' selected. The 'EG 4.3 Pre-Prod Properties' dialog has the 'Shortcut' tab active, showing the following configuration:

- Target type:** Application
- Target location:** 4.3
- Target:** terpriseGuide\4.3\SEGuide.exe /profile:dc3c89
- Start in:** "C:\Program Files\SAS92\EnterpriseGuide\4.3"
- Shortcut key:** None
- Run:** Normal window
- Comment:** (empty)

Buttons at the bottom of the dialog include 'Find Target...', 'Change Icon...', 'Advanced...', 'OK', 'Cancel', and 'Apply'. To the right, a taskbar shows various applications, with 'EG 4.3 Pre-Prod' and 'EG 4.3 Prod' highlighted.

EG – AUTOEXEC --- kinda

The image shows two overlapping windows from a SAS application. The background window is the 'Options' dialog box, and the foreground window is an 'Edit' window.

Options Dialog Box:

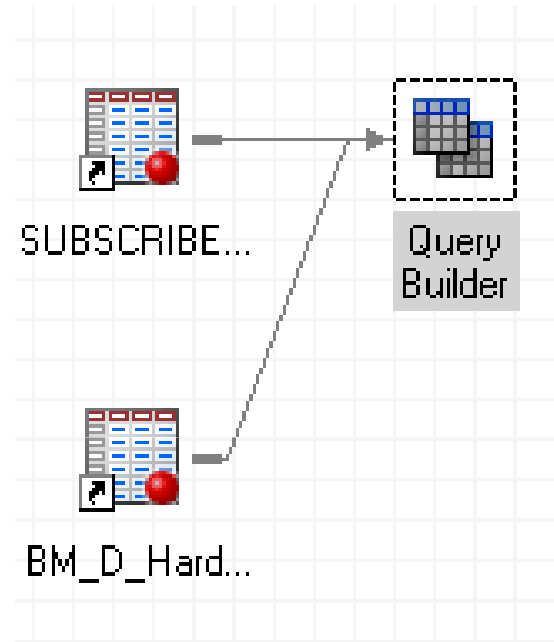
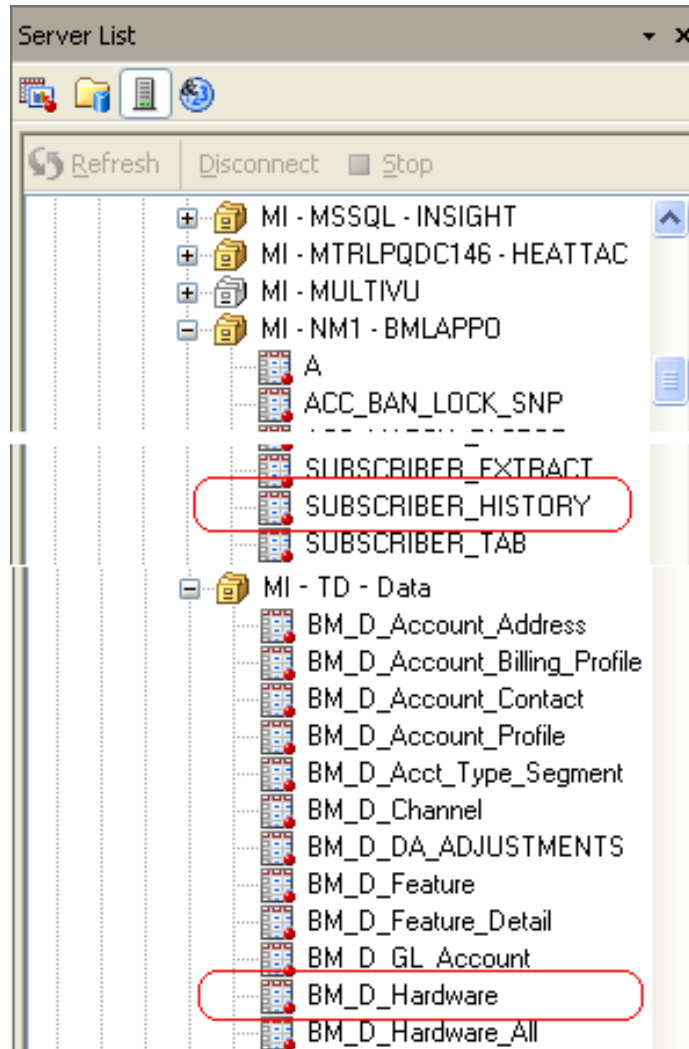
- Left Panel:** A tree view with categories: General, Project Views, Project Recovery, Results (Results General, Viewer, SAS Report, HTML, RTF, PDF, Graph, Stored Process), Data (Data General, Performance), Query, OLAP Data, Tasks (Tasks General, Custom Code, Output Library), SAS Programs, Security, and Administration.
- Right Panel (SAS Programs):**
 - General:** Close all open data
 - Additional SAS code:**
 - Insert custom SAS
 - Insert custom SAS
 - Submit SAS code when server is connected

Edit Window:

```
1 /* Insert custom code after server connection here */
2
3 %teradata_password_check(email=harry.droogendyk@bell.ca)
4
5 %let freq = nocol norow nopercnt missing;
6 options nocentre nodate;
7
8 libname myproj '/data/mi/data/sasdata/proj';
9
10 %include '/users/hdroogen/autoexec_test.sas';
11
```

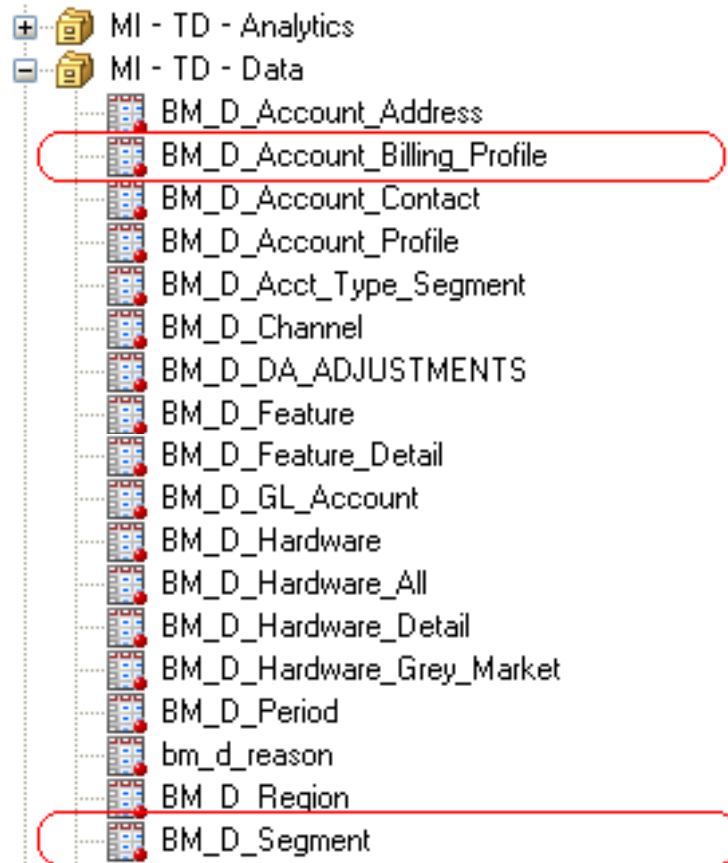
Buttons: Save, Cancel, Edit...

EG – Who Does What Where ?



- different libraries
- different databases
- millions of rows
- SY tomorrow !

EG – Who Does What Where ?



- same library
- same database
- candidate for pass-through



What is Pass-Through ?

- SAS Access “engines” for RDBMS
 - eg. Teradata, Oracle, ODBC
- essentially APIs to communicate with the DB
- explicit pass-through
 - SAS program connects to DB
 - SQL is “passed through” to DB
 - SQL must be native to DB
- implicit pass-through
 - SAS decides and takes care of it for you



Explicit Pass-Through

- open a program window in EG

```
proc sql;
  connect to teradata ( user=&user pass="&pass"
                       server=&server );

  create table pp_sum_cube as
    select * from connection to teradata
      (
        select s.brand_name, m.*
        from mi_analytics.v_bm_f_pp_migration_sum_cube m
        left join
        mi_data.bm_d_segment s
        on m.segment_cd = s.segment_cd
        where m.period_dt in ( '2012-03-01' , '2012-03-02' )
        order by m.period_dt, s.brand_name
      );

quit;
```



Implicit Pass-Through

- You write SAS
- SAS interprets your code and it writes Teradata SQL
 - functions
 - ... if possible ...
- In Database processing
 - SAS Procedures
 - SAS formats



Implicit Pass-Through

```
libname mitddata teradata user=&user pass="&pass"  
        server=&server sql_functions=all ;
```

```
proc sql;  
    select segment_tier_0, count(*) as cnt  
        from mitddata.BM_D_Segment  
        group by segment_tier_0  
        order by segment_tier_0  
    ;  
quit;
```

- SAS LIBNAME statement – Teradata engine
 - SAS SQL

 - summary query runs entirely in Teradata
 - small result set returned to SAS
-



Implicit Pass-Through

```
libname mitddata teradata user=&user pass="&pass"  
        server=&server sql_functions=all;
```

```
proc freq data = mitddata.BM_D_Segment;  
    tables segment_tier_0 /missing nopercnt ;  
run;
```

```
514 proc freq data = mitddata.BM_D_Segment;  
515     tables segment_tier_0 /missing nopercnt ;  
516 run;
```

NOTE: SQL generation will be used to construct frequency and crosstabulation tables.



Implicit Pass-Through – Uh oh...

```
proc sql;  
    select adj_type, adj_reason, count(*) as cnt  
        from mitddata.BM_D_DA_Adjustments  
        where substr(adj_reason,1,3) = 'M2M'  
            and intck('month',eff_dt,exp_dt) > 1  
        group by adj_type, adj_reason  
        order by adj_type, adj_reason  
        ;  
quit;
```

- runs *forevvvvvvvvvvver*
-



Implicit Pass-Through – Uh oh...

- SUBSTR and INTCK are SAS functions...
- SQL_Functions = ALL
- finite list
 - SUBSTR is there
 - not INTCK
 - what did SAS do ?
- better than you might imagine

```
options sastrace=',,,d'  
        sastraceloc=saslog nostsuffix;
```



Implicit Pass-Through – Uh oh...

SAS_SQL: Unable to convert the query to a DBMS specific SQL statement due to an error.

ACCESS ENGINE: SQL statement was not passed to the DBMS, SAS will do the processing.

TERADATA: trqacol- No casting. Raw row size=66, Casted size=74, CAST_OVERHEAD_MAXPERCENT=20%
TERADATA_8: Prepared: on connection 12


```
SELECT "ADJ_TYPE", "ADJ_REASON", "EFF_DT", "EXP_DT"  
      FROM mi_data."BM_D_DA_Adjustments"
```

```
WHERE ( ( SUBSTR("ADJ_REASON", 1, 3) = 'M2M' ) )
```

TERADATA_9: Executed: on connection 12
SELECT "ADJ_TYPE", "ADJ_REASON", "EFF_DT", "EXP_DT" FROM mi_data."BM_D_DA_Adjustments" WHERE ((SUBSTR("ADJ_REASON", 1, 3) = 'M2M'))

TERADATA: trget - rows to fetch: 30

TERADATA: trforc: COMMIT WORK



Implicit Pass-Through – Caveat

Due to incompatibility in date and time functions between Teradata and SAS, Teradata might not process them correctly.

Check your results to determine whether these functions are working as expected

- how to be certain ?
 - explicit pass-through

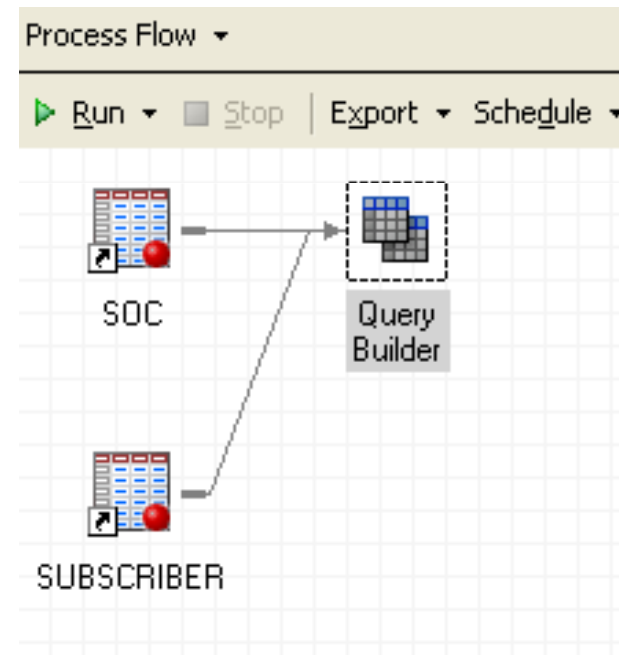


Implicit Pass-Through – Ummmm

- Oracle DB
 - two schemas

BMLAPPO BMLREF

- SAS will happily run in Oracle
- turns SQL into Oracle query
- much happiness



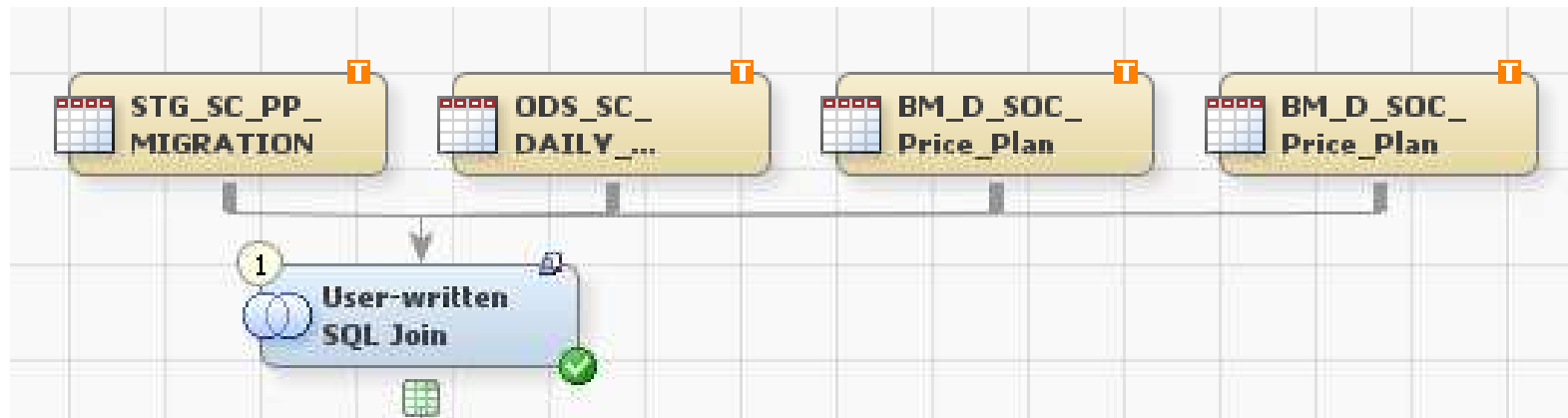
Implicit Pass-Through – Ummmm

- Teradata DB

- two schemas

MI_DATA

MI_STAGE



- 15,000 rows left joined to 120,000,000
 - 576 amps !?!
 - 48 minutes later...
-



Implicit Pass-Through – NOT !

```
TERADATA_7: Prepared: on connection 6
SELECT * FROM mi_data."BM_D_SOC_Price_Plan"
```

ERROR: This SQL statement **will not be passed to the DBMS for processing** because it involves a join across librefs with different connection properties.

```
TERADATA: trqacol- Casting decimals. Raw row size=38, Casted size=38, CAST_OVERHEAD_MAXPERCENT=20%
```

```
TERADATA_8: Prepared: on connection 4
SELECT "OLD_PP_SOC_CD", "NEW_PP_SOC_CD", CAST("SERVICE_ACTIVITY_KEY" AS
FLOAT), "PERIOD_DT", CAST("SERVICE_ACTIVITY_KEY_PREV" AS FLOAT)
FROM mi_stage."STG_SC_PP_MIGRATION"
```

NOTE: PROCEDURE SQL used (Total process time):

real time	48:12.89
user cpu time	30:29.85
system cpu time	6:40.33



Explicit Pass-Through – Joy

```
select * connection to teradata (  
  select ..  
    from MI_STAGE.ODS_SC_DAILY_ACTIVITY,  
         MI_STAGE.STG_SC_PP_MIGRATION left join  
         MI_DATA.BM_D_SOC_Price_Plan as BM_D_SOC_Price_Plan_prev  
    on STG_SC_PP_MIGRATION.OLD_PP_SOC_CD =  
       BM_D_SOC_Price_Plan_prev.PP_SOC_CD  
  where STG_SC_PP_MIGRATION.SERVICE_ACTIVITY_KEY =  
        ODS_SC_DAILY_ACTIVITY.SERVICE_ACTIVITY_KEY  
  and STG_SC_PP_MIGRATION.PERIOD_DT =  
      ODS_SC_DAILY_ACTIVITY.PERIOD_DT  
  and ODS_SC_DAILY_ACTIVITY.ADJ_KEY = 0  
);
```

NOTE: PROCEDURE SQL used (Total process time):

real time	4.66 seconds
user cpu time	0.22 seconds
system cpu time	0.04 seconds



Temporary Tables

- intermediate results
 - subsetting exercise
 - deep dive analysis
- need historical data for these 10,000 subscribers
 1. Pull subscriber history down to SAS, join
 2. where subscriber_no in ('123','345','567' x 10K)
 3. Push 10,000 subscriber numbers up
 - do the join in DB
 - pull only what you need down
 - minimize slow data transfers



Temporary Tables

- Oracle – “global temp” tables
- Teradata – volatile tables

- exist for the duration of the session
 - like SAS WORK datasets

- requires
 - LIBNAME engine
 - explicit pass-thru
 - EXECUTE (DDL statement) by teradata;
 - implicit pass-thru to load subset
 - explicit pass-thru for final result



Temporary Tables

```
libname td_volt teradata server="&server"  
      user=&user password="&pass" dbmstemp=yes  
      connection=global dbcommit=0;
```

```
proc sql;
```

```
  connect to teradata ( server="&server"  
    user=&user password="&pass"  
    connection=global mode=teradata );
```

```
  /* create an empty volatile table */  
  execute ( create multiset volatile table  
    test_vol ... ( .. columns .. )  
    ) by teradata;
```

```
quit;
```



Temporary Tables

```
/*  
Put rows into the volatile table via the libname -  
if lots of records, additional options necessary,  
e.g. fastload  
*/
```

```
proc sql;  
  insert into td_volt.test_vol  
    select subscriber_number as subscriber_no  
    from midata.dim_subscriber ( obs = 100 )  
  ;  
quit;
```



Temporary Tables

```
/* Execute pass-thru query to join Teradata table  
to volatile table */
```

```
proc sql;
```

```
    connect to teradata (server="&server" user=&user  
                        password="&pass" connection=global);
```

```
    create table sas_datastet_of_results as  
    select * from connection to teradata (  
        select s.*  
            from bmbi_view.vb_subscriber      s,  
                test_vol                      v  
            where v.subscriber_no = s.sub_no  
                );
```

```
quit;
```



Conclusion

- do stuff where it makes sense
 - use power of DB
 - summarize, subset, sort in DB
- not all implicit pass-thru is
 - use option `SASTRACE`
- use `EXPLAIN`



Contact

Harry Droogendyk

harry.droogendyk@bell.ca

harry@stratia.ca

Phone: 905-282-3476

Web: www.stratia.ca/papers

