Example 1. RSPT Services: Querying a Table in DB2

Purpose

This example shows how to query a DB2 table that is located on a remote host by using SQL statements issued from a local host SAS session.

Program

The following sequence of statements would be used in an OS/390 SAS session to establish a connection to DB2 and query the table SYSIBM.SYSTABLES:

```sas
connect to db2 (ssid=db2p);
select * from connection to db2
  (select name, creator, colcount
   from sysibm.systables
   where creator='THOMPSON' or creator='JONES');
```

The same connection and query could be performed from an OS/2 SAS session by using RSPT by means of a SAS server in a remote session under OS/390:

```sas
connect to remote
  (server=rmt dbms=db2 dbmsarg=(ssid=db2p));
select * from connection to remote
  (select name, creator, colcount
   from sysibm.systables
   where creator='THOMPSON' or creator='JONES');
```
Use the AS alias clause in the CONNECT TO statement to give the same name to the connection to the remote DBMS as it would have if you connected directly to it. This enables you to use queries without changing the FROM CONNECTION TO clause:

```sql
connect to remote as db2
  (server=rmt dbms=db2 dbmsarg=(ssid=db2p));

select * from connection to db2
  (select name, creator, colcount
   from sysibm.systables
   where creator='THOMPSON' or
   creator='JONES');
```

---

**Example 2. RSPT Services: Subsetting Remote SAS Data**

**Purpose**

The PROC SQL view SALES97 presents sales data for fiscal year 1997 and is defined on a UNIX workstation as follows:

```sql
create view servlib.sales97 as
  select sum(amount) as amount
  from sales
  where year=1997;
```

Processing this view (by using RLS from your local SAS session under Windows) is comparatively fast because the view is interpreted in the server SAS session. The summary function SUM() is applied when the view is interpreted and only the summary row is returned to your local SAS session.

If you want to obtain only your own sales data and break down the sales by customer, you could use RLS or RSPT. The following sections show you how either of these services could be used and explain why RSPT is a better choice.

**RLS Program**

You can create a new view in your local SAS library to access the underlying data by using RLS from your local SAS session under Windows, as follows:

```sql
libname mylib 'C:\sales';

libname servlib '/dept/sales/revenue'
  server=servername;

create view mylib.sales97 as
  select customer, sum(amount) as amount
  from servlib.sales
  where year=1997 and
  salesrep='L. PETERSON'
  group by customer
  order by customer;
```

However, processing this view is expensive because the summary is not performed until the data reaches the local SAS session. This means more data is sent across the
network. In the following RSPT example, the summary is done before data is transferred. This reduces the amount of data that crosses the network.

**RSPT Program**

The following statements create a new PROC SQL view in a local SAS library that uses RSPT to access the remote SAS data:

```sas
libname mylib 'C:sales';
libname servlib '/dept/sales/revenue' server=servername;
proc sql;
connect to remote (server=servername);
create view mylib.sales97 as
  select * from connection to remote
  (select customer, sum(amount) as amount
   from servlib.sales
   where year=1997 and
     salesrep='L. PETERSON'
   group by customer
   order by customer);
```

*Note: The libref SERVLIB must be defined for the remote SAS library either in your SAS session or in the server SAS session. In this example, a LIBNAME statement is executed in the local SAS session to access the library through the server that is running in the remote session. Alternatively, you could remote submit a LIBNAME statement to define the library.*

You may want to create a view in the remote server, which can be used by many people. By modifying the previous example to include all sales reps, the view satisfies the needs of users who are interested in the sales made by more than one sales rep. The following example creates a view in the server session that summarizes the data by customer for all sales reps:

```sas
libname servlib '/dept/sales/revenue' server=servername;
proc sql;
connect to remote (server=servername);
execute
  (create view servlib.cust97 as
   select customer,
     sum(amount) as amount from sales
   where year=1997
   group by customer) by remote;
```