Paper 27-27

U.S. Census Bureau Goes OLAP

Hung X Phan, U.S. Census Bureau
Lori A Guido, U.S. Census Bureau
Richard A Denby, U.S. Census Bureau

ABSTRACT

This paper describes a technique used to build an Online Analytical Processing (OLAP) system that was developed at the U.S. Census Bureau to be used for the review of Census 2000 long form (sample) data. The system uses the multidimensional database (MDDB) report object in SAS/EIS® to create 74 commonly used reports categorized by subject, demographic characteristics and geography to allow users to review the data. These reports are presummarized, which enables the users to quickly retrieve multiple levels of aggregate data through a multidimensional view. The system also uses Hybrid On-Line Analytical Processing (HOLAP) techniques to build a "proxy" HOLAP cube. Through the use of the proxy cube, the system registers data from the 50 states and the District of Columbia (D.C.) uses the same 74 reports to review the data for all states. Without utilizing the "proxy" HOLAP cube technology, one would have to build reports for each state or 3,774 reports. However, the proxy cube uses a SAS® view to gain access to the Multidimensional Data Provider (MDP), where the data sets for the 50 states and D.C. reside. Since the SAS view cannot be indexed, surfacing the expected 45 million detail data records in the system would be inherently slow. The paper shows how overriding methods written in SCL can be used to overcome this obstacle to improve the performance.

INTRODUCTION

As required by its constitution, the United States conducts a census in years ending in "0" to count the population and housing units. The population counts determine how seats in the U.S. House of Representatives are apportioned and also are required to draw congressional and state legislative district boundaries, to allocate federal and state funds, to formulate public policy, and to assist with planning and decision-making in the private sector. The decennial census uses both short- and long-form questionnaires to gather information. The short form asks a limited number of basic questions of all people and housing units and are often referred to as 100-percent questions because they are asked of the entire population. The long form asks more detailed information from approximately a 1-in-6 sample of households, and includes the 100-percent questions as well as questions on education, employment, income, ancestry, homeowner costs, units in a structure, number of rooms, plumbing facilities and other topics. The data received from the long form sample are used to estimate various demographic and socioeconomic characteristics for the U.S. population and for lower levels of geography.

The Census Bureau's Housing and Household Economics Statistics Division (HHES) is responsible for reviewing, analyzing, verifying, and validating the Census 2000 data on housing, income, and other socioeconomic areas.

We built this OLAP system to help our analysts gain access to multi-dimensional data from all 50 states and D.C. stored in SAS data sets so they analyze the data from all different view points.

CLIENT/SERVER MODEL

We store the 51 data sets, one for each state and D.C., on a Unix Sun server running Solaris 8. Each survey analyst has a Pentium® III PC running Windows 95/98/2000 and can gain access to the data on the server via a SAS/CONNECT® session running SASV8.2. A script file called tcpunix.scr is used to start the SAS session on the unix server. A desktop icon is created on each PC which invokes the following autoexec.sas at run time.

```
/* autoexec.sas used to set up the
OLAP long form review environment. */

libname censusv8 'c:\LF_REVIEW';
options comamid=tcp remote=hhessas;
filename rlink
'!sasroot\connect\saslink\tcpunix.scr
';
signon;

libname holaptst remote
slibref=holaptst server=hhessas;
libname holaploc
'c:\windows\personal\my sas
files\v8\HOLAPLOC';
options fmtsearch=(holaptst);
```

THE OLAP LONG FORM DATA ACCESS & REVIEW SYSTEM

The survey analyst is initially presented with the OLAP Long Form Data Access & Review screen, Figure 1, where he clicks on one or multiple states from the U.S. map. The analyst clicks on the GO button after making a geography selection.

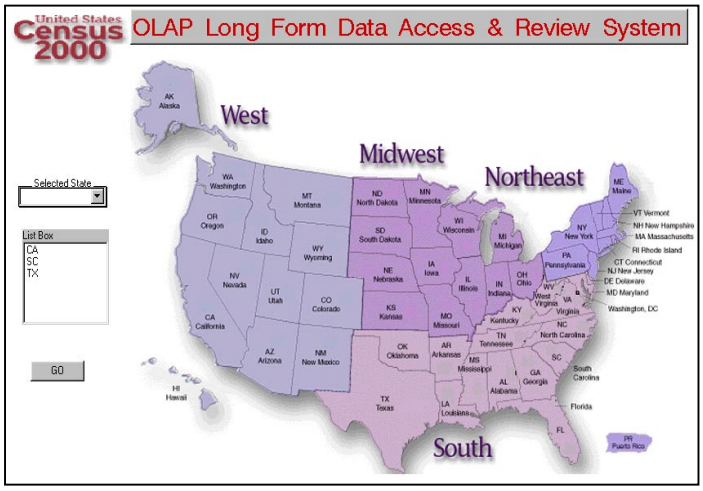


Figure 1: OLAP Long Form Data Access & Review System screen

The analyst subsequently clicks and makes a selection on subject characteristics such as education, income, migration, ancestry, labor force, disability, place of work, industry and occupation. For the purpose of this paper, we will take the subject of education as an illustration. In Figure 2, the analyst can select one of the six multidimensional reports built for education by clicking on the Education icon.

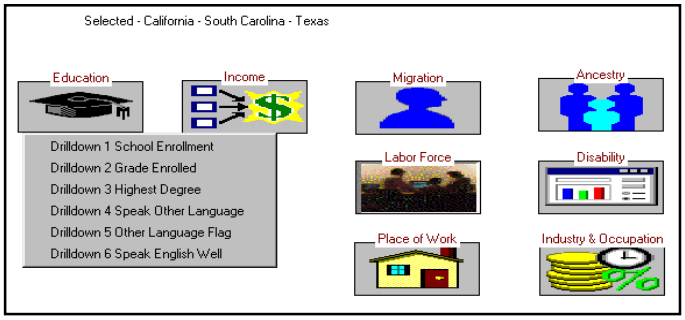
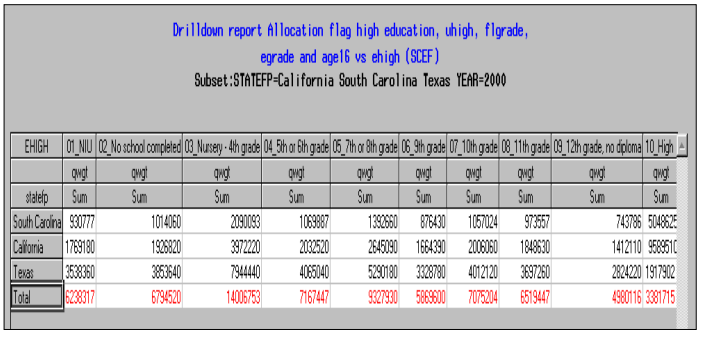


Figure 2: Subject Characteristics Menu Selection screen

Figure 3 below shows the education report number 3 of highest degree attained for the selected states, California, South Carolina, and Texas. This is the default hierarchy. The analyst has multiple options accessible by pressing the right mouse button to make a list of hierarchies appear for selection through which he can view and analyze the data. For example, the report layout and statistics can be changed or customized.



Drilldown report Allocation flag high education, uhigh, flgrade, egrade and age16 vs ehigh (SCEF)
Subset:STATEFP=California South Carolina Texas YEAR=2000

EHIGH	01_NIU	02_No school completed	03_Nursery - 4th grade	04_5th or 6th grade	05_7th or 8th grade	06_9th grade	07_10th grade	08_11th grade	09_12th grade, no diploma	10_High
	qwgt	qwgt	qwgt	qwgt	qwgt	qwgt	qwgt	qwgt	qwgt	qwgt
statefp	Sum	Sum	Sum	Sum	Sum	Sum	Sum	Sum	Sum	Sum
South Carolina	930777	1014060	2090093	1063887	1392660	876430	1057024	973557	743786	5048625
California	1769180	1926820	3972220	2032520	2645090	1664390	2006060	1848630	1412110	9589510
Texas	3538360	3853640	7944440	4065040	5290180	3328780	4012120	3697260	2824220	1917902
Total	6238317	6794520	14006753	7167447	9327930	5869600	7075204	6519447	4980116	3381715

Figure 3: Drilldown Education Report Number 3 of Highest Degree screen

THE SAS DATA SETS AND THE MDDBS

The 51 SAS data sets, one for each state, contain the detail data for a specific subject, demographic characteristics, and geography. As a demonstration, in this paper we will examine a set of education characteristics data sets. The naming conventions we employ, for example EDU45_00, denotes education for the state of South Carolina (45 is South Carolina's FIPS – Federal Information Processing Standards – code) for 2000 (denoted by 00). Similarly, EDU45_90 refers to education characteristics for South Carolina in the 1990 census.

We create 51 MDDBs from the 51 SAS detail data sets using PROC MDDB. The MDDB procedure reads and summarizes the SAS data sets and stores the summarized data in the multidimensional database for fast and easy access. The MDDBs contain the class variables and hierarchies needed to create the reports from which the survey analysts can examine, analyze, and navigate through large amounts of data with great speed and from all angles. Listing below contains a sample of the PROC MDDB code used to create the 51 MDDBs.

```
proc mddb data=cen2000.edu&state._&yr.
out=cen2000.edu&state._&yr;

  class attend;              * attended school
  class grade;               * grade or level attending
  class high;                * highest degree
  class speak;               * speak language other than English
  class langcod;             * language code
  class pob;                 * place of birth
  class span;                * Hispanic origin
  class age;                 * ages
  class sex;                 * gender
  class citizen;             * citizenship
  class race;                * race
  class incwg;               * income, wages
  class occupa;              * occupation
  class year statefip;       * year statefip
  class county tract block;  * county tract block


hierarchy statefip attend racehisp age16 age2;
hierarchy statefip flgrade ugrade flhigh high racehisp age16 age2;
hierarchy statefip flhigh uhigh flgrade egrade age16;
hierarchy statefip speak flspeak citizen racehisp agelang;
hierarchy statefip fllang citizen racehisp;
hierarchy statefip flabil citizen yr2usr racehisp;
hierarchy statefip coufip tract block;
var qwgt/ n sum;
run;
```

THE VIEW

We then create a SAS view called EDUVIEW. The EDUVIEW contains no data but only information about the location of all of the education SAS data sets. The EDUVIEW is used as input to the proxy MDDB which is used in the generation of the EDU HOLAP cube. Listing below contains a sample of the code used to create the view.

```
data cen2000.eduview / view=cen2000.eduview;
 set
     cen2000.edu01_00   cen2000.edu01_90
     cen2000.edu02_00   cen2000.edu02_90
     cen2000.edu04_00   cen2000.edu04_90
     cen2000.edu05_00   cen2000.edu05_90
     cen2000.edu06_00   cen2000.edu06_90
     cen2000.edu08_00   cen2000.edu08_90
     cen2000.edu09_00   cen2000.edu09_90

…

      cen2000.edu49_00   cen2000.edu49_90
      cen2000.edu50_00   cen2000.edu50_90;
run;
```

THE PROXY MDDB

Next we create the subject education template proxy MDDB called EDUTMPLT. The EDUTMPLT stores metadata information such as the hierarchies, formats, and BASETBL attribute value locally on the client. This information will speed up performance, especially at initialization time.

We use PROC MDDB to build a Proxy MDDB with one observation that represents the structure of the education data in a OLAP Group called EDUMDDB. The EDUMDDB maintains a link to the 50 MDDBs which are stored on the Unix server. Listing below contains a sample of the code used to create the template for proxy MDDB cube.

```
proc mddb data=cen2000.eduview (obs=1) out=cen2000.edutmplt;
  class attend;                 * attended school
  class grade;                  * grade or level attending
  class high;                   * highest degree
  class speak;                  * speak language other than English at home
  class langcod;                * language code
  class pob;                    * place of birth
  class span;                   * Hispanic origin
  class age;                    * ages
  class sex;                    * gender
  class citizen;                * citizenship
  class race;                   * race
  class incwg;                  * income, wages
  class occupa;                 * occupation
  class year statefip;          * year statefip
  class county tract block; * county tract block

hierarchy statefip attend racehisp age16 age2/name=
          'attend/racehisp/age16/age2' display=nodata;
hierarchy statefip flgrade ugrade flhigh high racehisp age16 age2/name=
          'Flgrade/grade/flhigh/ehigh/age16/age2' display=nodata;
hierarchy statefip flhigh uhigh flgrade egrade age16/name=
          'Flhigh/high/flgrade/grade/age16' display=nodata;
hierarchy statefip speak flspeak citizen racehisp agelang/name=
          'speak/flspeak/citizen/racehisp/agelang' display=nodata;
hierarchy statefip fllang citizen racehisp/name=
          'Fllang/citizen/racehisp' display=nodata;
hierarchy statefip flabil citizen yr2usr racehisp/name=
          'Flabil/citizen/yr2usr/racehisp' display=nodata;
hierarchy statefip coufip tract block/name=
          'Coufp/tract/block' display=nodata;
var qwgt/ n sum;
run;
```

SET UP REPOSITORY MANAGER

Before we can utilize the Proxy MDDB EDUTMPLT in the Metadata, we must first set up a repository manager. The repository manager manages metadata repositories and provides a central point of reference for metadata in the SAS system. Starting with Version 8 of the SAS® System, the SAS/EIS metabase facility has been converted to the new Common Metadata Manager (CMR). The CMR enables SAS/EIS software to share metadata with other SAS System products. One of the major components of the Common Metadata Repository is the Repository Manager.

To set up the repository manager, we create a repository directory, d:\cen_olap\repos, which is reserved exclusively for the storage of repository manager files. At a SAS command line, type REPOSMGR, and press Enter. Once the Repository Manager window opens as seen in Figure 4, select Setup Repository Manager.

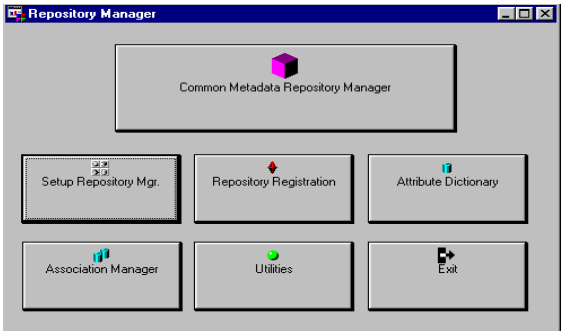


Figure 4: Repository Manager window

In the Repository Manager Setup window, keep the library name default to RPOSMGR. For path, specify d:\cen_olap\repos, and select the Write values to system registry check box as seen in Figure 5, then click OK.

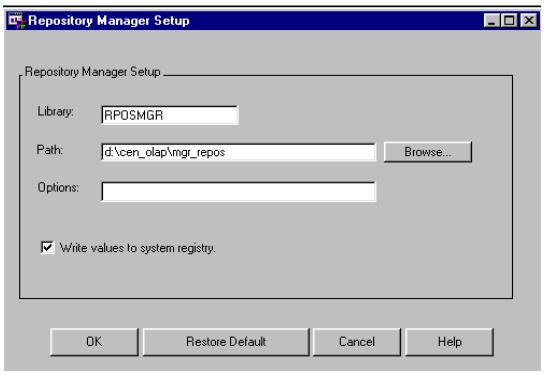


Figure 5: Repository Manager Setup window

SET UP METADATA REPOSITORY

Once the repository manager is built, we need to set up the metadata repository. Create a directory, d:\cen_olap\meta_repos, which is reserved exclusively to the storage of metadata repository files. From the Repository Manager window, select Repository Registration. Enter META_REPOS for Registration Name, d:\cen_olap\meta_repos for Path, and Metadata Repository for Description. Leave the Readonly access check box unchecked as seen in Figure 6, and click OK.

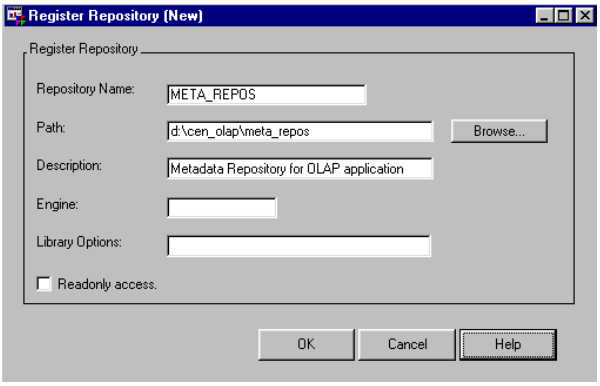


Figure 6: Register Repository window

SET UP THE MDP METADATA

The Multidimensional Data Provider (MDP) uses a set of metadata that describes where each component of a logical data group is located and what it contains. MDP facility is used to define data groups and servers. To invoke the MDP facility, type MDMDDB at a SAS command line and press Enter. Once the Distributed Multidimensional Metadata window opens, as seen in Figure 7, click on the Edit button. Notice that the META_REPOS is the repository we assigned previously in the set up metadata repository step.

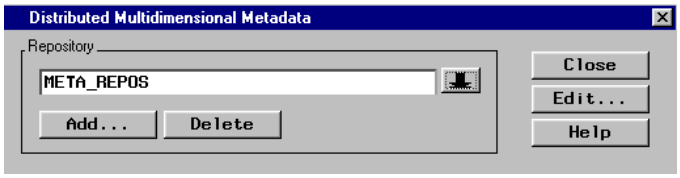


Figure 7: Distributed Multidimensional Metadata window

DEFINE THE MDP SERVER

An MDP server definition is required if data sources reside on non-local machines. Note that data sources can only use MDP server definitions that reside within the same repository. We use a SAS/CONNECT® session to establish a connection to the Unix Sun server running Solaris 8. In the Server tab, Figure 8, click on the Add button to define a new server definition that identifies the remote SAS session.

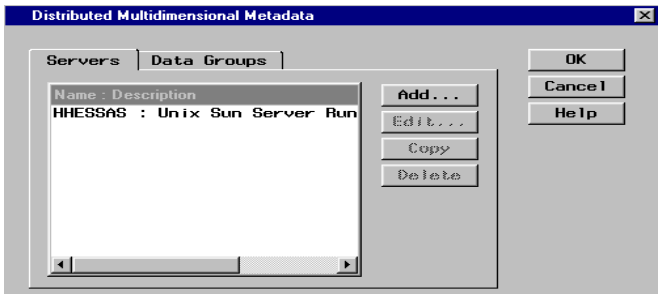


Figure 8: Distributed Multidimensional Metadata – Server tab window

In the General tab, Figure 9, we specify HHESSAS as the remote server name and give it a description. The communication protocol used to connect the client PCs to the server is TCP/IP, and the script is d:\cen_olap\tcpunix.scr script. The DNS for the IP address assigned to the Unix server is hhessas.hhes.census.gov.

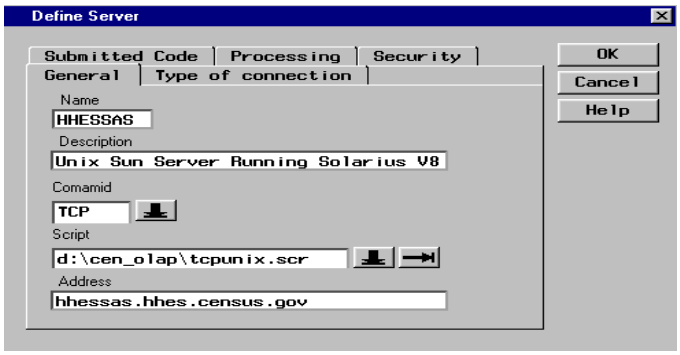


Figure 9: Define Server – General tab window

In the Type of connection tab, Figure 10, specify server invocation and processing load distribution options. Specify when to sign on and sign off and whether to use Remote Library Services (RLS) or Remote Submit or Both (let MDP decide). We select Both for our application. We take the default on Processing and Security tabs.

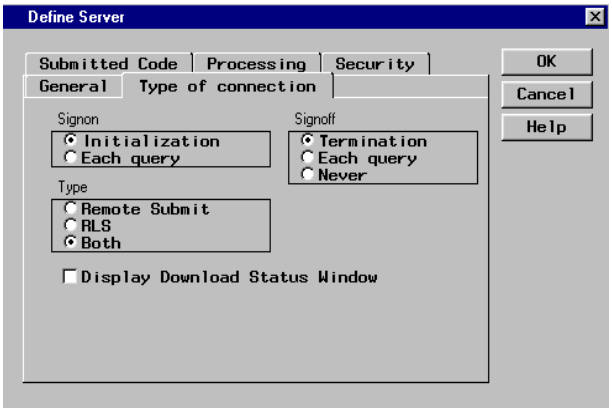


Figure 10: Define Server – Type of connection tab window

DEFINE THE DATA GROUP & THE DATA SOURCES

To define a data group called EDUMDDB, we open the Distributed Multidimensional Metadata window and select the Data Groups tab, then click on the Add button. In the General tab, type the name EDUMDDB and the description EDU HOLAP MDDBs as seen in Figure 11.

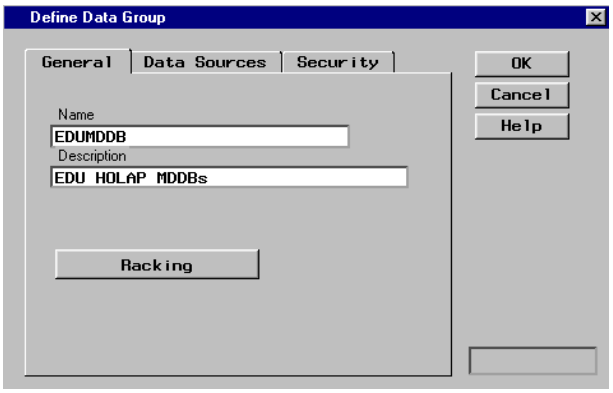


Figure 11: Define Data Group – General tab window

To define all of the data sources for the data group, select the Data Sources tab and click on the Add button to start adding data sources to the data group EDUMDDB. Figure 12 below shows an example of the data sources that have been added to the group.

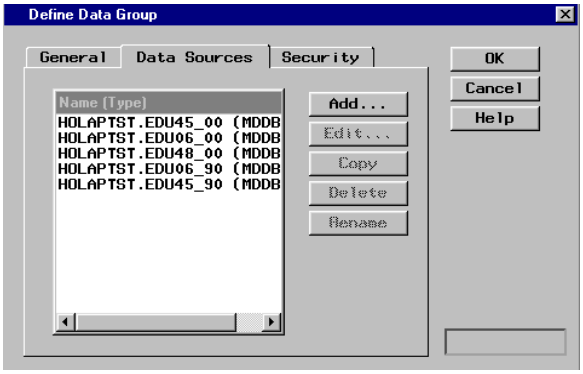


Figure 12: Define Data Group – Data Sources tab window

EXPORT DATA GROUP TO METABASE

Once the EDUMDDB data group is defined, it needs to be exported and registered in the Metabase. From the Data Groups tab in the Distributed Multidimensional Metadata window, click on the Export button. The Export to Metabase window appears where we enter the information as seen in Figure 13. This enables us to use the EDUMDDB data group and server information transparently in EIS applications.

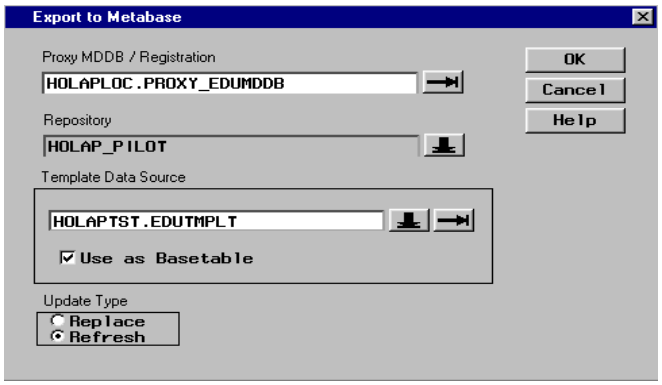


Figure 13: Export to Metabase window

BUILD THE EIS APPLICATIONS & THE OVERRIDE METHODS

We build the Graphical User Interface (GUI) that provides a U.S. map and acts as a front end to the main reporting screens as seen in Figure 1 using SAS/AF® software. The reporting screens as seen in Figure 2 and Figure 3 are built using EIS multidimensional table objects.

Giving the analysts access to the detail data in the MDDBs from a cell which contains the summary value based on a specific criteria presents a challenge. This is because the "show detail" selection from the right-mouse-button pop-up menu uses the reachthrough method via the SAS view, EDUVIEW, as the default. The EDUVIEW cannot be indexed because it is not a real data table. The EDUVIEW is a read-only object which contains no data, but instead the metadata information about the data. Hence, it is extremely slow reaching through the specified detail data sets from a cell using the default method, because it is performing a sequential read of the records in the base table.

To overcome this challenge, the MODEL of the SAS/EIS object must be overridden. When the detail data from a cell is requested, the following SCL code is executed. It gets the information from the active cell, constructs the necessary criteria and submits the query directly to the base table without first going to the view.

```
entry dataset $20;  /* the working ds */

INIT:

*get the user selection from the active cell;

 if nameditem(getnitemn(_self_,'ACTIVE_CELL'),'STATEFP') >0
   then
statefp=getnitemc(getnitemn(_self_,'ACTIVE_CELL'),'STATEFP');
   else = 'U.S.'; *default to entire U.S.
 end;

if nameditem(getnitemn(_self_,'ACTIVE_CELL'),'YEAR') >0
  then
year=getnitemc(getnitemn(_self_,'ACTIVE_CELL'),'YEAR');
  else   year='2000';*default 2000 census data;
end;

*build the where clause from active cell information;
 listid=makelist();

do i=1 to listlen(active_cell);
 *name of the cell;
 where_name=trim(left(nameitem(active_cell,i)));

*construct the where clause
if i=1 then do;

fmt_name=trim(left(getitemc(fmt_list,nameditem(fmt_list,getitem
c(active_cell,i)))));

where_clause=where_name||'='||quote(fmt_name);
 listid=insertc(listid, where_clause ,-1);
end;

 call putlist(listid, 'WHERE clause', 0);

* submit the where clause;
  call send(widid, '_set_where_', listid);
  dataid=getnitemn(widid,'dataid');
  call
send(dataid,'_SET_INSTANCE_METHOD_','_GET_ACTIONS_',
'sashelp.eis.actions.scl','GETACT');


end;

*remove where clause and close the working ds;
  rc=where(dsid);
  if dsid then rc=close(dsid);
RETURN;

MAIN: /* execute when a right-mouse-button is clicked */
  cmdline=word(1,'U');
  call nextcmd();
  whlist=makelist();
  call send(dataid,'_GET_WHERE_',whlist);
 RETURN;

TERM:
   if dsid>0 then rc=close(dsid);
RETURN;
```

Below is the SCL code for overwriting the _POSTINIT_ method of the multipledimensional report object in SAS/EIS.

```
dcl char(200) _method_ ;
/* pop-up menu: override _POSINIT_ */
posinit: method;
   call super(_self_ , _method_);
  _self_.modelid._setInstanceMethod('_displayReachThru',
'censusv8.mainmenu.reach.frame');
endmethod;
```

Once the method is writen, it needs to be associated with the SAS/EIS multiple dimensional report object. From the Build/EIS Multidimensional Report window, click on the Advanced button. From the SAS/EIS – Advanced window, click on the Method tab. Figure 14 shows what the completed window should look like.

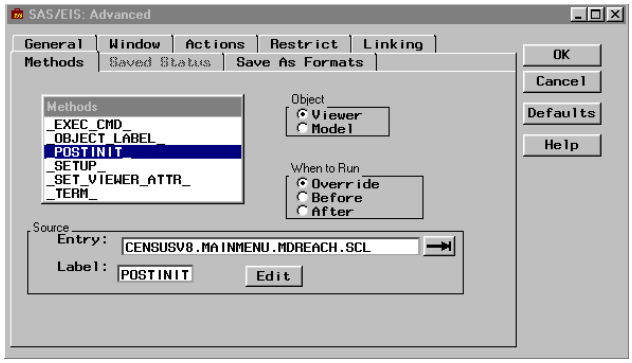


Figure 14: SAS/EIS: Advanced – Method tab window

CONCLUSION

This paper presents a step-by-step technique for how to build a client/server OLAP system. The system was built with a very generic and generalized approach in order to work with any SAS data sets. The paper also describes a reach through override method to change the behavior of the drill down to the detail data directly to the base table and not via a SAS view.

ACKNOWLEDGEMENT

The authors would like to acknowledge the technical contributions of Deborah Gattuso, Ahsan Ullah, and Joe Zilka of SAS Institute Inc.

REFERENCES

SAS Institute Inc. SAS® OLAP Server Administrator's Guide

Release 8.1, Cary, NC: SAS Institute Inc., 2000.

The authors can be contacted at:

Mr. Hung X Phan

U.S. Census Bureau

1483/3 RDPB/HHES

Washington, D.C. 20233 - 8500

Tel: 301-457-3204

E-mail: hung.x.phan@census.gov

Ms. Lori A Guido

U.S. Census Bureau

1483/3 RDPB/HHES

Washington, D.C. 20233 - 8500

Tel: 301-457-3204

E-mail: lori.a.guido@census.gov

Mr. Richard A Denby

U.S. Census Bureau

1065/3 HHES

Washington, D.C. 20233 - 8500

Tel: 301-457-6810

E-mail: richard.a.denby@census.gov

DISCLAIMER

This paper reports the results of a research and development project undertaken by the authors. It is not an official report of the U.S. Census Bureau and does not necessarily reflect the views of the U.S. government. The authors are solely responsible for any errors and/or inaccuracies in the contents of this paper.