

# A Dynamic Web Application: Using SAS/IntrNet™ and Web Publishing Software

Vecdet Mehmet-Ali, Statistics Canada, Ottawa, Ontario.

*Many companies use Intranets to deliver private corporate information to internal users. At Statistics Canada the Informatics Branch utilizes Intranet technology to communicate with its clients and provide them with up-to-date information on its services and rates. Until recently, we were executing SAS programs, which generated static Web pages for our clients. Static pages fulfill the purpose of distributing the information, however they don't provide the ability to query the data. With the introduction of SAS/IntrNet and Web publishing tools to our application, our Web pages provide clients with the ability to invoke SAS programs on the server. These programs create customized, dynamic Web pages that are sent back to the client's browser. This paper looks at our fully automated Intranet application.*

## Introduction

Intranet functionality is often reduced to the dissemination of static documents and company information. This does not demonstrate the power of an Intranet as a contributor to an organization's overall profitability. Intranets and Web based technologies are presenting unique opportunities today. To capitalize on these, we need to link Intranet applications to enterprise business processes. With the introduction of SAS/IntrNet tools we now have the ability to access and display enterprise data on customized dynamic web pages.

In this paper, steps for the deployment of Intranet applications, which use SAS/IntrNet tools to display mainframe data, are introduced. I will use the example of a fully automated system, which provides the clients with the ability to generate customized reports from mainframe billing data, on the Intranet.

## Mainframe Accounting (Charge-back) System

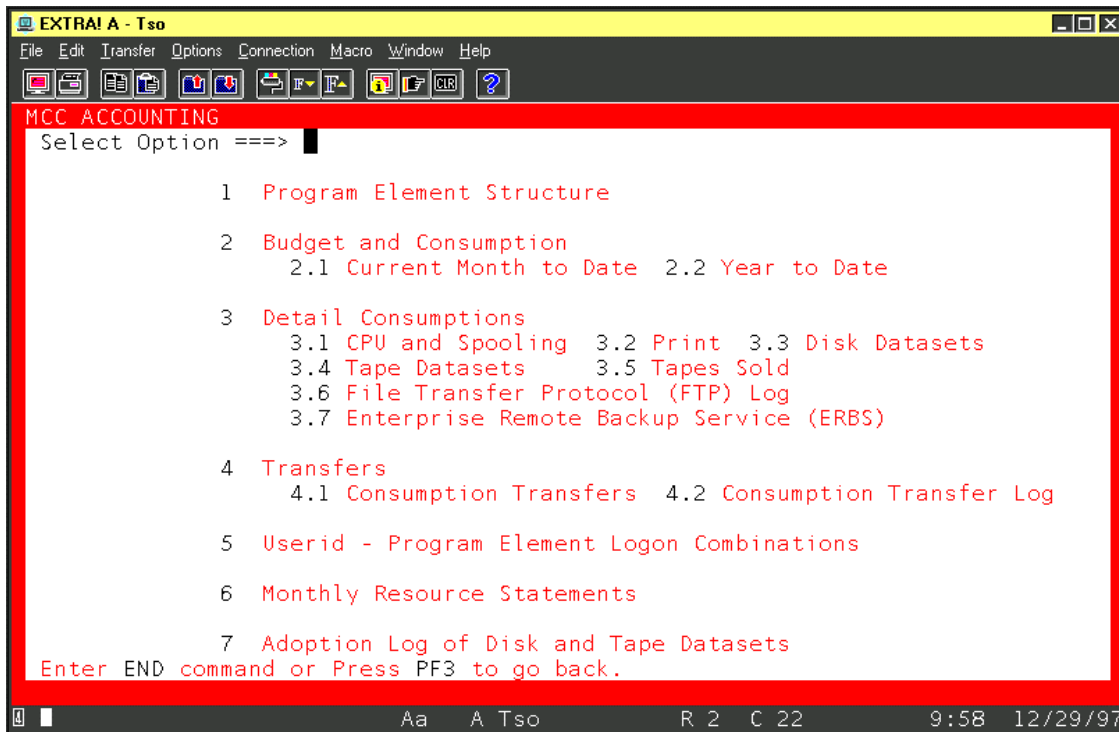
System Management Facility (SMF) data is produced by the MVS operating system to record activity, at the rate of about 300 megabytes per day. At the end of each day,

SMF data is dumped by a utility and then processed by accounting jobs within a charge-back system. The accounting data, both detail and summarized levels, is then maintained in generally accessible SAS data libraries and is used to charge the clients and to facilitate report production, performance analysis and capacity planning.

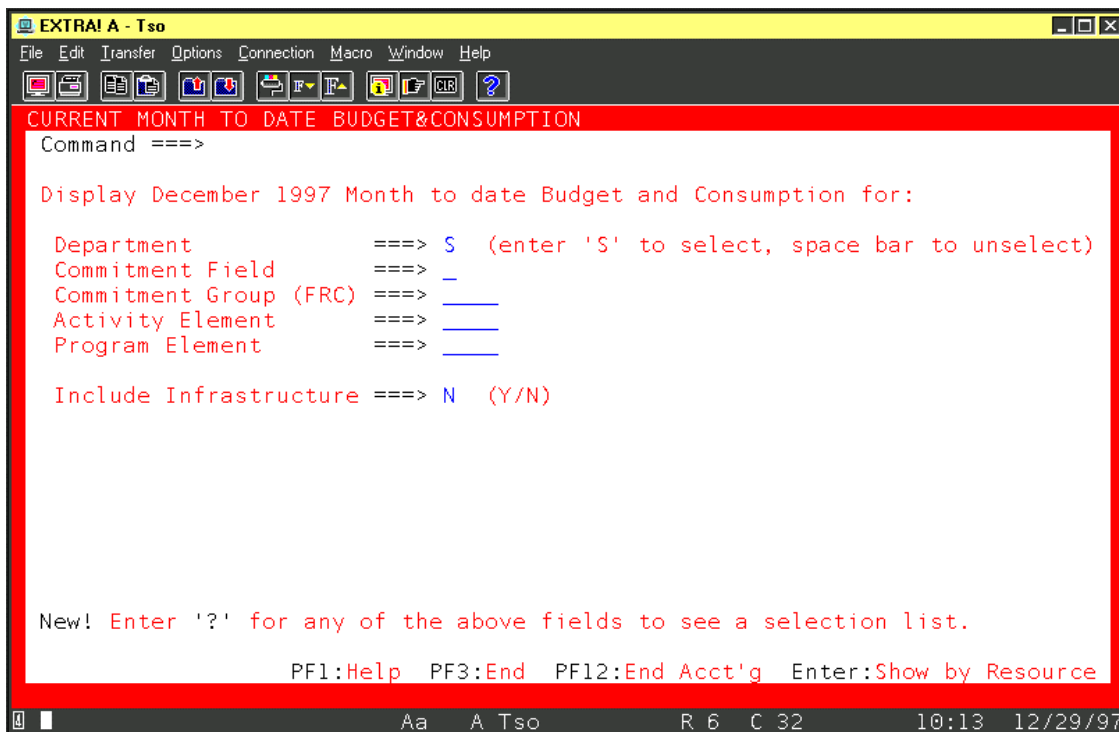
## TSO based (What we used to do)

A series of SCL screens (see Figure 1 to Figure 3.2) provide the clients with an interface to generate customized electronic reports on the mainframe. This interface had been annoying for users who need to spend more money to view their bill. The client cost includes a TSO session on the mainframe, and prime-time use of SAS and mainframe CPU cycles.

Clients, who would like to display accounting information, logon to the mainframe using TSO. Then they start ISPF and select the panel for Mainframe Accounting and Capacity Planning. This starts a SAS session, allocates necessary libraries and displays the SCL screen shown in (Figure 1). We will select one of the options and follow it to demonstrate how the report is generated.



**Figure 1. Mainframe Accounting Interface (Main menu).** Written in SCL, it had been annoying for the clients. Typing 2.1 on the command line will take us to the Current Month to Date Budget and Consumption screen shown in Figure 2.



**Figure 2. Current Month to Date Budget and Consumption screen.** Clients can query budget and consumption amounts at one of the five hierarchical administration levels. This screen always points to the current month's data so there is no field for date selection. The client has to select one of the administration levels to generate the report. A question mark typed on any of the fields provides a selection list to choose from. Clients are also allowed to type in values - an 'S' in front of 'Department' will generate the report for this level shown in Figure 3.1.

EXTRA! A - Tso

File Edit Transfer Options Connection Macro Window Help

CURRENT MONTH TO DATE BY RESOURCE

Command ==>

Date: 29DEC97:10:14

Department  
(Excluding Infrastructure)

	Budget	Consumption	Remaining Budget	Projected Consumption	Projected + or -	Proj %%%%
Computing	\$635,614	\$333,727	\$301,887	\$372,990	\$262,624	59%
Print	\$0	\$73,479	-\$73,479	\$82,124	-\$82,124	***%
Tape	\$0	\$91,607	-\$91,607	\$101,423	-\$101,423	***%
Disk	\$0	\$125,848	-\$125,848	\$139,332	-\$139,332	***%
	\$635,614	\$624,662	\$10,952	\$695,869	-\$60,255	109%

Consumption by Cost-Recovery PE's \$18,248 (excluded from above)

Note: Use PF1 for an explanation and mapping to detail panels

PF1:Help PF2:Save PF3:End PF4:Sub-Level PF9:Restart PF12:End Acct'g

Aa A Tso R 2 C 16 10:13 12/29/97

**Figure 3.1** Report generated as a result of a query at the 'Department' level. Budget, consumption, projected consumption, total amounts, etc. are displayed for different resources, at the selected level. Users can view budget and consumption amounts at the sub-level of the hierarchy by pressing a PF key such as PF 4 to go to Figure 3.2.

EXTRA! A - Tso

File Edit Transfer Options Connection Macro Window Help

CURRENT MONTH TO DATE BY SUB-LEVEL

Command ==>

PF1:Help PF2:Save PF3:End S+PF4:SubLevel PF9:Restart S+Enter:Resource

Date: 29DEC97:10:14 Department (Excluding Infrastructure)

Commitment Field	Budget	Consumption	Projected Consumption	Projected + or -	Proj %%%%	Cost-Recvry Consumption
Total	\$635,614	\$624,662	\$695,869	-\$60,255	109%	\$18,248
— 3	\$11,928	\$16,718	\$18,639	-\$6,711	156%	\$47
— 4	\$35,323	\$40,080	\$44,699	-\$9,376	127%	\$0
— 5	\$270,755	\$314,277	\$350,312	-\$79,557	129%	\$7,934
— 6	\$55,725	\$46,619	\$51,837	\$3,888	93%	\$7,813
— 7	\$20,769	\$21,706	\$24,202	-\$3,433	117%	\$84
— 8	\$241,114	\$185,262	\$206,180	\$34,934	86%	\$2,371

Aa A Tso R 10 C 3 16:07 1/05/98

**Figure 3.2** One line summary values are displayed for each of the sub-level entries. This gives users the ability to drill down the hierarchy. Each row is a summary line for a sub-level entry, in this case it is a financial administration unit called a 'Field'. Users may choose one by typing an 'S' on its left and pressing on the 'Enter' key. This will display the detail for that 'Field' using a screen similar to the one in (Figure 3.1). In this way, the users could drill down all the way through the hierarchy.

## Web based environment

The basic Web services coupled with SAS's additional tools have provided an environment for evolution of our reporting technology.

### The Internal Web (Intranet)

All companies have information that must be distributed to internal employees or external customers and suppliers. The Web technology gives us a way to put this information on-line for instantaneous access by authorized users. Intranets, provide security within a controlled audience and dramatically reduce the cost and time of content development, duplication, distribution and usage. The basic procedure for displaying static web pages involves:

- creation of content
- migration of content to intranet environment

### SAS/IntrNet Software

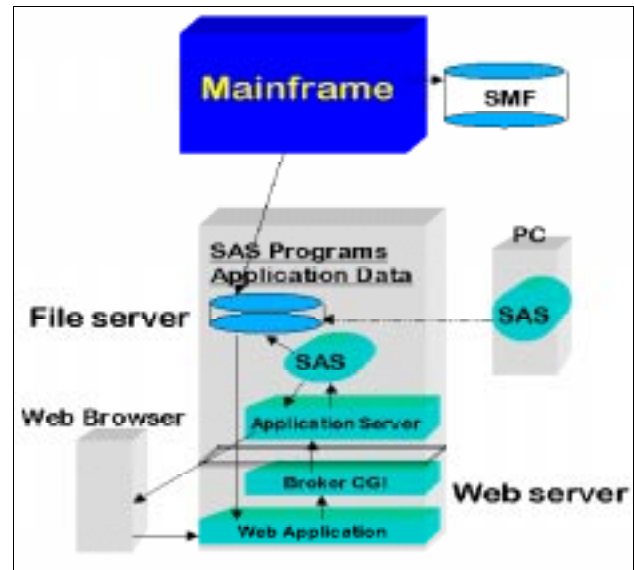
With the introduction of SAS/IntrNet tools we now have the ability to display data dynamically on intranets.

**The Application Dispatcher** was introduced in a SUGI 22 paper [1] and is one of the components of SAS/IntrNet Software. It provides a general-purpose gateway that allows a web browser to trigger execution of specific SAS code that resides on the web server or any other machine, and receive the results back as an HTML page. The Application Dispatcher has three components:

- **The Application Broker CGI** is installed on the web server. It provides the link between the web browser and the compute services. It forwards web browser requests, to the Application Server, for processing.
- **The Application Server** is a SAS session that listens for requests from the Broker CGI. Upon receiving a request it runs the specified SAS program.
- **SAS Applications** are programs that are written to perform a predefined processing. Result is an HTML page, which is returned to the web browser.

## Web based (What we do now)

To satisfy our clients' need for a variety of reports at different levels of detail and their preference for web services, we developed an intranet interface to the Mainframe Accounting data. The data is copied to the network server (Figure 4) and the new interface, which uses SAS/IntrNet Application Dispatcher, allows our clients to dynamically generate reports using their favorite browser.



**Figure 4.** In the web based system clients can dynamically generate customized reports using their favorite browser, from the Mainframe Accounting data, with the help of SAS/IntrNet Application Dispatcher.

### Processing on the Mainframe

After the SMF data has been processed, a production job copies various data sets, to a SAS library. This library is used to transport the data from the mainframe to the network server and it contains:

- SAS data sets for the applications on the web
- A data set with a last update entry for each data set in the library.

This library is then converted into transport format using CPORT procedure. The transport file is then copied to the Server using FTP. The single, simple job step (Figure 5) is submitted by the mainframe job scheduling system.

```
//NA%DDD.FTP JOB (0165,MN06),USERID
/*
/* * FTP transfer file to Network Server
/*
//P1S1 EXEC PGM=FTP,REGION=4M
//SYSPRINT DD SYSOUT=*
//OUTPUT DD SYSOUT=*
//INPUT DD *
HostName
userid
password
binary
put 'cars.aa.mframe.text' Mframe.txt
QUIT
```

**Figure 5. Copying SAS data to a Network Server**  
Sample JCL and FTP commands to copy 'cars.aa.mframe.text' from the mainframe to the server.

## Processing on the PC

A series of SAS jobs are submitted to update the web applications with the new data. The first job, in the series converts the transport file into a SAS library.

Some applications receive new data daily while others receive it monthly. Applications find out if they are getting new data by checking the last update data set.

Most of the applications view the data at five hierarchical administration levels:

- Department
- Field
- FRC (Financial Responsibility Code)
- Activity Element
- Program Element

To improve performance, data is summarized at these levels by daily and monthly jobs, in ways that make it ready for all possible client selections and drill downs.

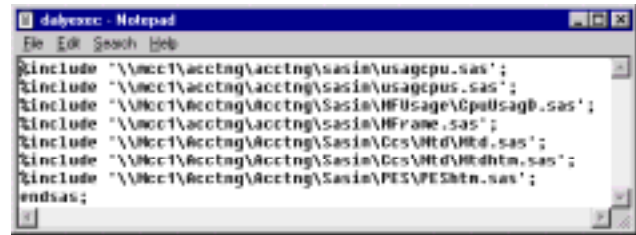
After the data is processed as above, SAS programs generate the HTML pages for applications which received new data. Clients are provided with drop-down lists to make selections from. These lists are generated from the data, hence no validation is necessary when client makes a selection on the web page. To reflect the changes in data, web pages have to be re-generated daily or monthly, with the new updated drop-down lists and URLs. These pages are then stored on the web server. The name of a page is constructed from the application name and year and month of its data.

## Automating the daily process on the PC

The SMF data is dumped at 01:00 a.m. every night. The processing of the data, updating of the files and FTP'ing the transport file to the PC generally completes around 02:00 a.m. Processing on the PC has to be done sometimes after that. Our environment is SAS 6.12 running under Windows NT 4.0. We have a job scheduler that would start SAS with an autoexec file, at 03:00 a.m., each day. The command to start SAS and execute the daily job is entered on the command line:

```
d:\sas612nt\sas.exe -autoexec
\\mcc1\acctng\acctng\sasin\dalyexec.sas
```

The SAS autoexec file needed to include the daily jobs for execution is dalyexec.sas:



```
dalyexec - Notepad
File Edit Search Help
%include "\\mcc1\acctng\acctng\sasin\usagepu.sas";
%include "\\mcc1\acctng\acctng\sasin\usagepus.sas";
%include "\\mcc1\acctng\acctng\sasin\HFusage\Gpulsagb.sas";
%include "\\mcc1\acctng\acctng\sasin\HFframe.sas";
%include "\\mcc1\acctng\acctng\sasin\Gcs\Htd\Htd.sas";
%include "\\mcc1\acctng\acctng\sasin\Gcs\Htd\Htdhtm.sas";
%include "\\mcc1\acctng\acctng\sasin\PES\PEShtm.sas";
endsas;
```

**Figure 6. SAS autoexec file.** It submits a series of production jobs to process the data that was received from the mainframe by FTP.

## The Web Applications

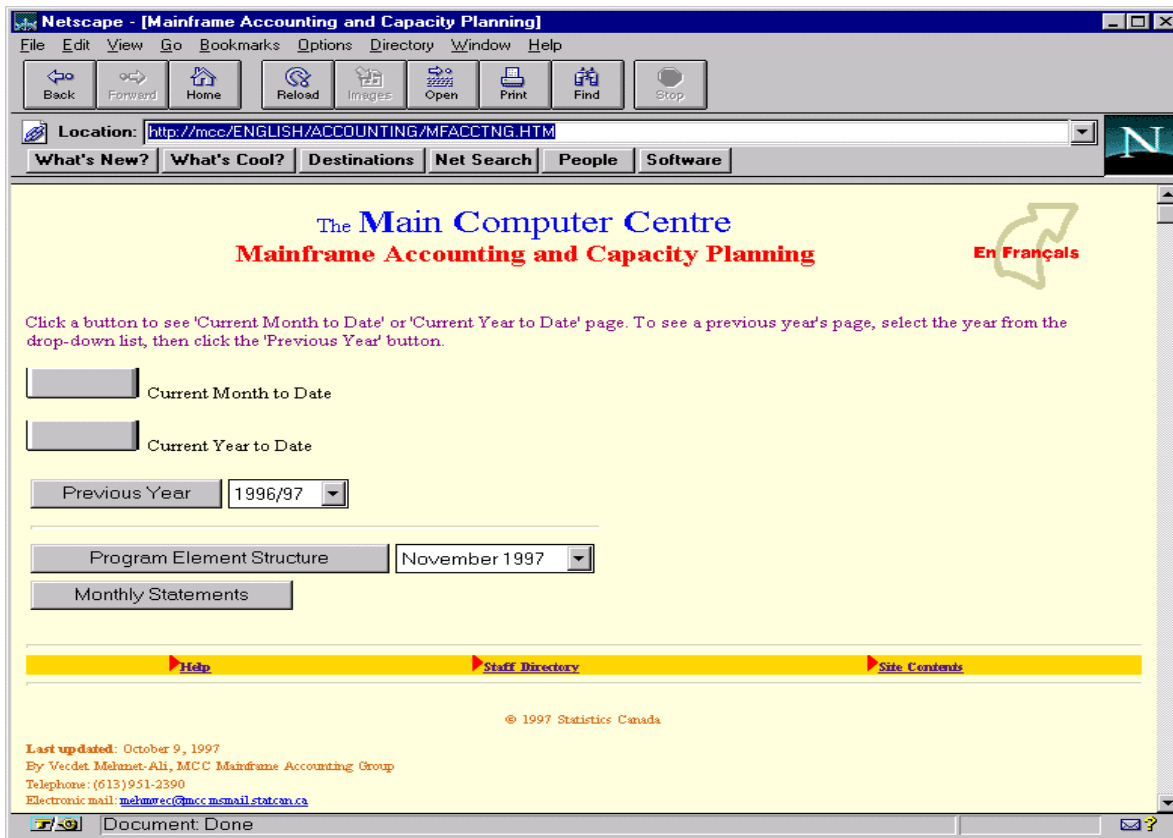
By integrating SAS/IntrNet tools, we provided our clients with the ability to generate customized reports dynamically on our intranet. The only thing client needs is a web browser. The interface to our web applications (see Figure 7) is generated by SAS programs, which are submitted at the beginning of the month. Applications update the drop-down lists by adding the new month or the new year to the selection lists.

A new and important feature is the ability of the web pages to use the same data to generate the reports in either English or French. The switch is done by a simple click on the icon, which is available at the top right hand corner of each page.

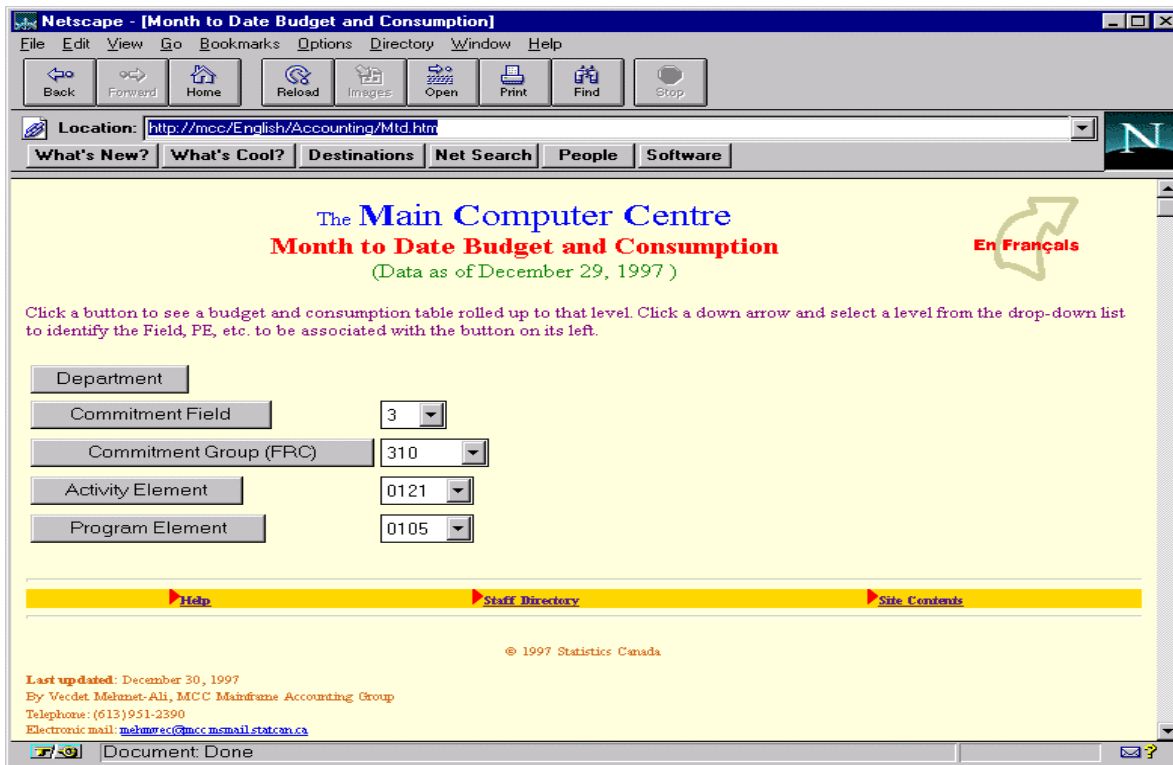
One other major difference of the web pages from the mainframe screens is that they are all mouse driven. Selections are made through drop down lists, buttons and icons. This made the query validation process, which is resource intensive on the mainframe, unnecessary.

One of the differences between this page and its counterpart on the mainframe is the fact that some of the applications require date selection here. These are the applications, which have an HTML page for each month with its unique hierarchical drop down selection lists. Selected date is used, in constructing the name of the HTML page and the name of the data set, which will be allocated.

We will make the same selections as we did on the mainframe screens to give us an idea about the same application running on the web. Clicking on the Current Month to Date button takes us to that application's web page (see Figure 8).

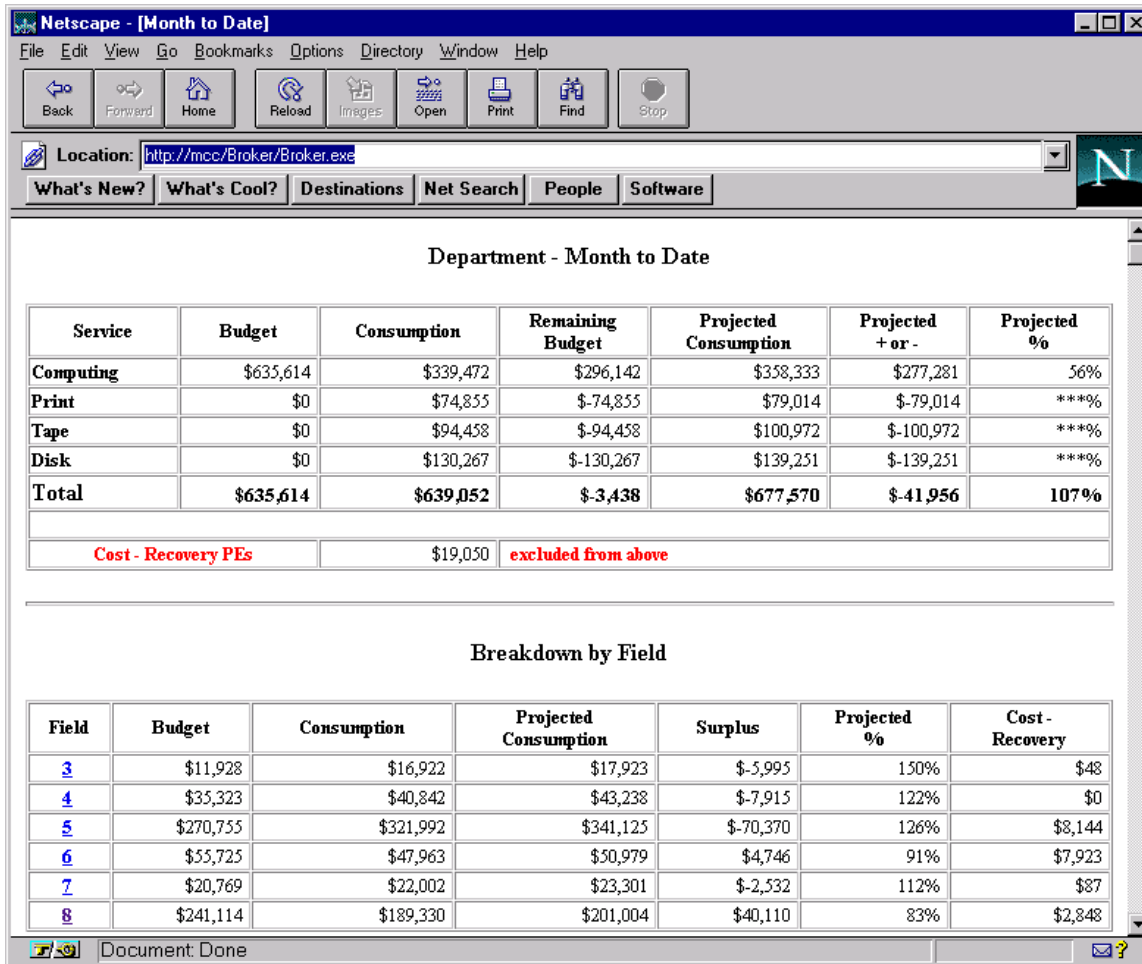


**Figure 7. Mainframe Accounting on Intranet (Main menu).** SAS programs regenerate this page with updated drop-down lists, every month. Clicking on the Current Month to Date button takes us to that application's web page (Figure 8).



**Figure 8. Month to Date Budget and Consumption query page.** This application page displays budget and consumption amounts rolled up to the selected level. . Clicking on the Department button takes us to a dynamic web page (see Figure 9).

In Figure 8 we see the five hierarchical levels and their selection lists. Clicking on one of the hierarchical buttons on this page triggers the submission of a SAS program (see Section B in Appendix) associated with the form on this page, to the SAS session through the Application Server. This program checks the hidden parameter 'LEVEL' and finds out which hierarchical level was selected. Then it submits the SAS program for the selected level, which produces the HTML page with two tables (Figure 9) and returns it to the web browser.



**Figure 9.** A dynamic page showing the values generated as a result of querying at the 'Department' level in (Figure 8). Clicking on one of the "Field Numbers" drills down to a similar page for the next lower level. Such pages are generated by a SAS program (see Section A in Appendix).

The first table is a detail display of the amounts at the selected level. The second table contains a summary row for each element at the next hierarchical level. Each sub-level element is presented as a URL to provide the client with the ability to drill down the hierarchy. An example of URL structure in this table is:

```
<A HREF='/Broker/broker.exe?_PROGRAM=pgms.MtdFld.sas&_SERVICE=default&field=3'>3</A>
```

Looking at the counterpart on the mainframe we see that there are two SCL screens (Figure 3.1 and Figure 3.2) which correspond to this page.

## Problem Areas

### **Drill downs**

One of the problems is the passing of the program name and the selections to the CGI script. We chose URLs to drill down. Therefore everything that has to be passed to the successive dynamic pages is packed into these URLs. In applications where the client selects the date of the data on the first page of the application, it has to be in the URLs all the way through the drill down. For an example, see Section C in Appendix.

### **Performance Issues**

Our development environment was Windows NT 4.0 Server running on a 486 DX4 to approximate a busy server. IIS 3.0 (web server) and SAS 6.12 ran on the same machine, making life pretty hectic for the server. Even with a single user, queries were too slow. To improve the performance, we took the following steps:

- Summarize the data ahead of time. This improved the performance dramatically.
- Display ready data (summarized as noted above) does not require the tabulate formatter to generate the tables. We were able to use simple 'put' statements in a 'Data' step to generate the dynamic pages (see Section A in Appendix). In some applications the output formatter is used to capture the output from PROC REPORT.

In the development environment the application server listened on only one port. In production with a faster server multiple ports can be utilized.

### **Security Considerations**

Leaving a workstation powered up and running overnight in an office environment has security problems of its own. To circumvent the problem, use a password protected screen saver. To do this, select the Display icon in the Control Panel, you will see Display Properties screen. From Screen Saver list, select Marquee Display. This will open up a window to define message text, font, color, etc. Select an appropriate text to prevent a co-worker from accidentally powering your workstation off. To lock the computer against unauthorized access, select the "Password protected" box in the Display Properties screen. Under NT, your password to logon to the computer is taken automatically.

## Conclusion

The application of Internet technologies in an Intranet setting can dramatically increase the flow and value of information within an organization. Users can gain quick and timely access to a much wider variety of existing information residing in a variety of original forms and sources. In addition, traditional paper-based information distribution can be displaced by Intranet applications, lowering costs and increasing the timeliness of information flow. Introduction of SAS/IntrNet tools gave us the ability to display enterprise data on customized dynamic web pages on our intranet. It provided the link between intranet applications and enterprise business processes.

Intranet applications can start at small scale and grow upwards over time, gradually providing or facilitating access to an increasing breadth of information, thus improving both employee productivity and satisfaction.

## References

1. Donald J. Henderson, Edmund Burnette, Vincent DelGobbo, John Leveille (1997), "The SAS/IntrNet™ Application Dispatcher", *Proceedings of the Twenty Second Annual SAS Users Group International Conference*, 22, 856-864.
2. Vecdet Mehmet-Ali (1997), "Displaying Information from Mainframe SAS Libraries on Your Intranet", *Proceedings of the Twenty Second Annual SAS Users Group International Conference*, 22, 890-895.

## Acknowledgements

The author wishes to thank **Roy Hammond** for his supervision and invaluable assistance.

SAS is a registered trademark of SAS Institute, Inc., in the USA and other countries.

Windows NT is a registered trademark of Microsoft, Inc. Other brand and product names are registered trademarks or trademarks of their respective companies.

## Author

Vecdet Mehmet-Ali  
 Statistics Canada  
 3117 Main Building, Tunney's Pasture  
 Ottawa ON K1A 0T6

Phone : (613) 951-2390  
 Fax : (613) 951-5198  
 E-mail : mehvec@statcan.ca



## Appendix

### Section A - Sample code that generates a dynamic web page

```

libname mtddata '\\mcc1\acctng\acctng\sasdata\mtd';
%let prjdate=10;
OPTIONS MISSING= ' ' NONUMBER NODATE;
DATA _NULL_;
  set mtddata.dep;
  format _numeric_ dollar12.;
  format pccpu pcprnt pcdisk pctape prpc PCNTFMT9.;
  FILE webout;
  put 'Content-type: text/html';
  put ;
  put '<HTML>';
  put '<HEAD>';
  put " <TITLE>Month to Date</TITLE>";
  put '</HEAD>';
  put "<BODY BGCOLOR=white>";
  put "<P>";
  put "<TABLE BORDER=1 WIDTH=100% ALIGN=CENTER CELLPADDING=1 CELLSPACING=1>";
  put " <CAPTION ALIGN=LEFT VALIGN=TOP><FONT SIZE=+1><B>Department - Month to Date</B></FONT><P>
  </CAPTION>";
  put " <TR>";
  put " <TH ALIGN=CENTER VALIGN=CENTER NOWRAP><FONT COLOR=black>Service</TH>";
  put " <TH ALIGN=CENTER VALIGN=CENTER NOWRAP><FONT COLOR=black>Budget</FONT></TH>";
  put " <TH ALIGN=CENTER VALIGN=CENTER NOWRAP><FONT COLOR=black>Consumption</FONT></TH>";
  put " <TH ALIGN=CENTER VALIGN=CENTER NOWRAP><FONT COLOR=black>Remaining<BR>Budget</FONT></TH>";
  put " <TH ALIGN=CENTER VALIGN=CENTER NOWRAP><FONT COLOR=black>Projected<BR>Consumption</FONT></TH>";
  put " <TH ALIGN=CENTER VALIGN=CENTER NOWRAP><FONT COLOR=black>Projected<BR>+ or -</FONT></TH>";
  put " <TH ALIGN=CENTER VALIGN=CENTER NOWRAP><FONT COLOR=black>Projected<BR>%</FONT></TH>";
  put "</TR>";
  put "<TR>";
  put " <TH ALIGN=LEFT VALIGN=CENTER NOWRAP><FONT COLOR=black>Computing</FONT></TH>";
  put " <TD ALIGN=RIGHT VALIGN=BOTTOM NOWRAP><FONT COLOR=black>"CCPU"</FONT></TD>";
  put " <TD ALIGN=RIGHT VALIGN=BOTTOM NOWRAP><FONT COLOR=black>"ACPU"</FONT></TD>";
  put " <TD ALIGN=RIGHT VALIGN=BOTTOM NOWRAP><FONT COLOR=black>"REMCPU"</FONT></TD>";
  put " <TD ALIGN=RIGHT VALIGN=BOTTOM NOWRAP><FONT COLOR=black>%nrstr(&nbsp;) "PRJCPU"</FONT></TD>";
  put " <TD ALIGN=RIGHT VALIGN=BOTTOM NOWRAP><FONT COLOR=black>%nrstr(&nbsp;) "SRCPU"</FONT></TD>";
  put " <TD ALIGN=RIGHT VALIGN=BOTTOM NOWRAP><FONT COLOR=black>%nrstr(&nbsp;) "PCCPU"</FONT></TD>";
  put "</TR>";
  put "<TR>";
  put " <TH ALIGN=LEFT VALIGN=CENTER NOWRAP><FONT COLOR=black>Print</FONT></TH>";
  put " <TD ALIGN=RIGHT VALIGN=BOTTOM NOWRAP><FONT COLOR=black>" CPRNT "</FONT></TD>";
  put " <TD ALIGN=RIGHT VALIGN=BOTTOM NOWRAP><FONT COLOR=black>" APRNT "</FONT></TD>";
  put " <TD ALIGN=RIGHT VALIGN=BOTTOM NOWRAP><FONT COLOR=black>" REMPRNT "</FONT></TD>";
  put " <TD ALIGN=RIGHT VALIGN=BOTTOM NOWRAP><FONT COLOR=black>%nrstr(&nbsp;) " PRJPRNT "</FONT></TD>";
  put " <TD ALIGN=RIGHT VALIGN=BOTTOM NOWRAP><FONT COLOR=black>%nrstr(&nbsp;) " SRPRNT "</FONT></TD>";
  put " <TD ALIGN=RIGHT VALIGN=BOTTOM NOWRAP><FONT COLOR=black>%nrstr(&nbsp;) " PCPRNT "</FONT></TD>";
  put "</TR>";
  put "<TR>";
  put " <TH ALIGN=LEFT VALIGN=CENTER NOWRAP><FONT COLOR=black>Tape</FONT></TH>";
  put " <TD ALIGN=RIGHT VALIGN=BOTTOM NOWRAP><FONT COLOR=black>" CTAPE "</FONT></TD>";
  put " <TD ALIGN=RIGHT VALIGN=BOTTOM NOWRAP><FONT COLOR=black>" ATAPE "</FONT></TD>";
  put " <TD ALIGN=RIGHT VALIGN=BOTTOM NOWRAP><FONT COLOR=black>" REMTAPE "</FONT></TD>";
  put " <TD ALIGN=RIGHT VALIGN=BOTTOM NOWRAP><FONT COLOR=black>%nrstr(&nbsp;) " PRJTAPE "</FONT></TD>";
  put " <TD ALIGN=RIGHT VALIGN=BOTTOM NOWRAP><FONT COLOR=black>%nrstr(&nbsp;) " SRTAPE "</FONT></TD>";
  put " <TD ALIGN=RIGHT VALIGN=BOTTOM NOWRAP><FONT COLOR=black>%nrstr(&nbsp;) " PCTAPE "</FONT></TD>";
  put "</TR>";
  put "<TR>";
  put " <TH ALIGN=LEFT VALIGN=CENTER NOWRAP><FONT COLOR=black>Disk</FONT></TH>";
  put " <TD ALIGN=RIGHT VALIGN=BOTTOM NOWRAP><FONT COLOR=black>" CDISK "</FONT></TD>";
  put " <TD ALIGN=RIGHT VALIGN=BOTTOM NOWRAP><FONT COLOR=black>" ADISK "</FONT></TD>";
  put " <TD ALIGN=RIGHT VALIGN=BOTTOM NOWRAP><FONT COLOR=black>" REMDISK "</FONT></TD>";
  put " <TD ALIGN=RIGHT VALIGN=BOTTOM NOWRAP><FONT COLOR=black>%nrstr(&nbsp;) " PRJDISK "</FONT></TD>";
  put " <TD ALIGN=RIGHT VALIGN=BOTTOM NOWRAP><FONT COLOR=black>%nrstr(&nbsp;) " SRDISK "</FONT></TD>";
  put " <TD ALIGN=RIGHT VALIGN=BOTTOM NOWRAP><FONT COLOR=black>%nrstr(&nbsp;) " PCDISK "</FONT></TD>";
  put "</TR>";
  put "<TR>";
  put " <TH ALIGN=LEFT VALIGN=CENTER NOWRAP><FONT COLOR=black size=+1><B>Total</B></FONT></TH>";
  put " <TD ALIGN=RIGHT VALIGN=BOTTOM NOWRAP><FONT COLOR=black><B>" CTOTAL "</B></FONT></TD>";
  put " <TD ALIGN=RIGHT VALIGN=BOTTOM NOWRAP><FONT COLOR=black><B>" ATOTAL "</B></FONT></TD>";
  put " <TD ALIGN=RIGHT VALIGN=BOTTOM NOWRAP><FONT COLOR=black><B>" REMTOT "</B></FONT></TD>";
  put " <TD ALIGN=RIGHT VALIGN=BOTTOM NOWRAP><FONT COLOR=black><B>%nrstr(&nbsp;) " PRJTOT "</B></FONT></TD>";
  put " <TD ALIGN=RIGHT VALIGN=BOTTOM NOWRAP><FONT COLOR=black><B>%nrstr(&nbsp;) " SRTOT "</B></FONT></TD>";
  put " <TD ALIGN=RIGHT VALIGN=BOTTOM NOWRAP><FONT COLOR=black><B>%nrstr(&nbsp;) " PRPC "</B></FONT></TD>";
  put "</TR>";
  put "<TR>";
  put " <TD COLSPAN=7 NOWRAP>%nrstr(&nbsp;)</TD>";
  put "</TR>";
  put "<TR>";
  put " <TH ALIGN=CENTER VALIGN=CENTER COLSPAN=2 NOWRAP><FONT COLOR=red>Cost - Recovery PEs</FONT></TH>";

```

```

put "      <TD ALIGN=RIGHT VALIGN=BOTTOM NOWRAP><FONT COLOR=black>" RTOTAL "</FONT></TD>";
put "      <TH ALIGN=LEFT VALIGN=CENTER COLSPAN=4 NOWRAP><FONT COLOR=red>%nrstr(&nbsp;) excluded from
above</FONT></TH>";
put "    </TR>";
IF day(input(put('PESUM',$LASTUPD.),DATE9.)) <= &prjdate THEN DO;
  put "    <TR>";
  put "      <TD ALIGN=LEFT VALIGN=BOTTOM COLSPAN=7 NOWRAP><FONT COLOR=GREEN>&CCMSG1</FONT></TD>";
  put "    </TR>";
END;
put "</TABLE>";
put "<P>";
put "<HR><P>";
run;

DATA NULL;
set mtddata.field END=LAST;
format _numeric_dollar12.;
format pccpu pcprnt pcdisk pctape prpc PCNTFMT9.;
FILE webout;
IF _N_ = 1 THEN DO;
  put "<HTML>";
  put "<P>";
  put "<TABLE BORDER=1 WIDTH=100% ALIGN=CENTER CELLPADDING=1 CELLSPACING=1>";
  put "  <CAPTION ALIGN=LEFT VALIGN=TOP>";
  put "    <FONT SIZE=+1><B>Breakdown by Field</B></FONT><P>";
  put "  </CAPTION>";
  put "  <TR>";
  put "    <TH ALIGN=CENTER VALIGN=CENTER NOWRAP><FONT COLOR=black>Field</TH>";
  put "    <TH ALIGN=CENTER VALIGN=CENTER NOWRAP><FONT COLOR=black>Budget</FONT></TH>";
  put "    <TH ALIGN=CENTER VALIGN=CENTER NOWRAP><FONT COLOR=black>Consumption</FONT></TH>";
  put "    <TH ALIGN=CENTER VALIGN=CENTER NOWRAP><FONT COLOR=black>Projected<BR>Consumption</FONT></TH>";
  put "    <TH ALIGN=CENTER VALIGN=CENTER NOWRAP><FONT COLOR=black>Surplus</FONT></TH>";
  put "    <TH ALIGN=CENTER VALIGN=CENTER NOWRAP><FONT COLOR=black>Projected<BR>%</FONT></TH>";
  put "    <TH ALIGN=CENTER VALIGN=CENTER NOWRAP><FONT COLOR=black>Cost -<BR>Recovery</FONT></TH>";
  put "  </TR>";
END;
put "  <TR>";
put "    <TD ALIGN=CENTER VALIGN=CENTER NOWRAP><FONT COLOR=black><B>" FLDHREF "</B></FONT></TD>";
put "    <TD ALIGN=RIGHT VALIGN=BOTTOM NOWRAP><FONT COLOR=black>" CTOTAL "</FONT></TD>";
put "    <TD ALIGN=RIGHT VALIGN=BOTTOM NOWRAP><FONT COLOR=black>" ATOTAL "</FONT></TD>";
put "    <TD ALIGN=RIGHT VALIGN=BOTTOM NOWRAP><FONT COLOR=black>%nrstr(&nbsp;) " PRJTOT "</FONT></TD>";
put "    <TD ALIGN=RIGHT VALIGN=BOTTOM NOWRAP><FONT COLOR=black>%nrstr(&nbsp;) " SRTOT "</FONT></TD>";
put "    <TD ALIGN=RIGHT VALIGN=BOTTOM NOWRAP><FONT COLOR=black>%nrstr(&nbsp;) " PRPC "</FONT></TD>";
put "    <TD ALIGN=RIGHT VALIGN=BOTTOM NOWRAP><FONT COLOR=black>" RTOTAL "</FONT></TD>";
put "  </TR>";
IF LAST THEN DO;
  put "</TABLE>";
  put "<P>";
  put "<HR>";
  put "</BODY>";
  put "</HTML>";
END;
run;

```

### Section B - Sample code determines the level selected and submits the associated SAS program.

```

libname mtddata '\\mcc1\acctng\acctng\sasdata\mtd';
%let PrjDate = 10;
filename dep '\\mcc1\acctng\acctng\pgms\MtdDep.sas';
filename cf '\\mcc1\acctng\acctng\pgms\MtdFld.sas';
filename cg '\\mcc1\acctng\acctng\pgms\MtdFRC.sas';
filename ae '\\mcc1\acctng\acctng\pgms\MtdAE.sas';
filename pe '\\mcc1\acctng\acctng\pgms\MtdPE.sas';
%MACRO CHKLEVEL;
  %IF "&LEVEL" = "Department" %THEN %include dep;
  %ELSE %IF "&LEVEL" = "Commitment Field" %THEN %include cf;
  %ELSE %IF "&LEVEL" = "Commitment Group (FRC)" %THEN %include cg;
  %ELSE %IF "&LEVEL" = "Activity Element" %THEN %include ae;
  %ELSE %IF "&LEVEL" = "Program Element" %THEN %include pe;
%MEND;
%CHKLEVEL;

```

### Section C - Sample code generates URLs for the hierarchical level 'Field'

```

Stmtdate is a macro variable that contains the year and the month of the data in 'yymmdd4.' Format.
DATA STMT.FLD&Stmtdate;
  SET FIELD;
  FLDHREF="<A HREF=' /Broker/broker.exe?_PROGRAM=pgms.StmtdFld.sas%nrstr(&_SERVICE)
=default%nrstr(&)Comfld="||
COMFLD||"%nrstr(&)Stmtdate=&Stmtdate'>"||COMFLD||'</A>';
RUN;

```