

## Paper 178-31

## State Agency Personnel Information Reports: A SAS/IntrNet® Project

Thomas J. Winn Jr., State Auditor's Office, Austin, Texas

### ABSTRACT

This paper describes how the author used SAS/IntrNet® to develop a simple intranet application to provide auditors and investigators at the State Auditor's Office with information extracted from several databases containing personnel information data for the State of Texas.

### INTRODUCTION

Sometimes auditors need information pertaining to the employment status or authorized salary of certain individuals, as of a particular date. The source data for responding to such queries are contained in three different databases that are maintained by the State Comptroller's Office. Instead of executing a separate query against the personnel databases each time a data request is received, the State Auditor's Office (SAO) decided to create its own version of the essential tables every quarter, containing the most frequently-requested columns, and to use those data for data requests, as needed. The SAO also decided to develop an intranet application which auditors could use, without having to solicit assistance from programmers, in order to create standardized reports or data files containing the personnel data that they need. Since the initial implementation of this application, it has expanded to include additional modules pertaining to more types of information. This paper will describe some of the "nuts and bolts" used in building the application.

### COMPUTING ENVIRONMENT

In November 2002, in order to create a facility capable of analyzing large data files, the State Auditor's Office established a SAS server system consisting of a Dell PowerEdge™ 2650 server (with two Pentium® III Xeon™ processors) connected to a Dell PowerVault™ storage system. The storage system consists of one PowerVault 660F 14-bay fiber channel RAID array and one PowerVault 220S 14-bay SCSI RAID array, both in RAID 5 configuration. The server is configured with Microsoft® Windows® 2000 Server. Initially, the SAS System software installed on our server was limited to Base SAS®, SAS/SHARE®, SAS/STAT®, SAS/GRAPH®, SAS/CONNECT®, and SAS/ACCESS® Interface to PC File Formats. And in October 2003, we added SAS/IntrNet®, and SAS/ACCESS® Interface to Sybase®.

The SAO's Web server has the same basic specifications as the SAS Server, except that there is no external storage system attached to it. It is a Dell PowerEdge 2650 server with dual Pentium III Xeon processors. The internal hard drives are configured in RAID 5 configuration. It is running Windows 2000 Server with Internet Information Services.

At the SAO, there are only five Systems Analysts who are knowledgeable about SAS programming. There are a few other programmers, who are proficient with certain development tools besides SAS. And there are many non-programmers, who possess varying levels of familiarity with using several application software packages. All SAO employees know how to access information on our intranet. Therefore, SAS/IntrNet applications seem to be an excellent method for delivering information to SAO employees, since using such applications requires only a web browser on the user's machine, and their users don't need to have any SAS knowledge.

### PERSONNEL DATA FOR THE APPLICATION

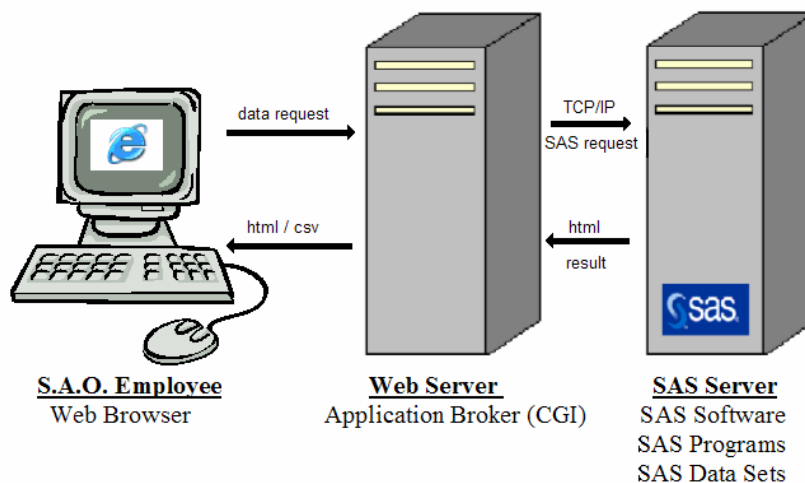
Data concerning employees of the State of Texas is distributed between three different personnel systems, each system containing information for different collections of state agencies and institutions. Although one might suppose that a single personnel and payroll system would be desirable, for historical reasons, Texas has three distinct systems, and each system has different content and capabilities. Moreover, over time many agencies have migrated from one system to another. Some types of information are common to all three systems (for example, employee names, social security numbers, gender, race, birthdates, and job titles), but the column names and other attributes for similar data items often differ from one system to the next.

The data for all three personnel systems are stored in DB2 databases on the Comptroller's mainframe computer. Unfortunately, that agency does not have SAS Integration Technologies, so we are unable to take advantage of the interoperability and easy information distribution that would be available using that SAS System component. Except during transitional periods, personnel information for any particular agency is stored in one, and only one, of the three personnel systems. Therefore, to obtain historical information concerning the current or former employees of most agencies, one would have to look in different systems for various intervals of dates.

At regular intervals (either quarterly or monthly, depending on the type of data), the most frequently-requested personnel information for all state employees are extracted from the pertinent databases on the Comptroller's mainframe computer. These data are downloaded to the SAO network environment, and are saved in a SAS Data Library on the SAS Server. This is accomplished using SQL Pass-Through, DATA step programming, certain basic SAS procedures, and FTP.

### AN OVERVIEW OF THE PERSONNEL INFORMATION REPORTS APPLICATION

Here is a conceptual diagram which describes the interaction of the operational components that are used by the Personnel Information Reports intranet application at the SAO:



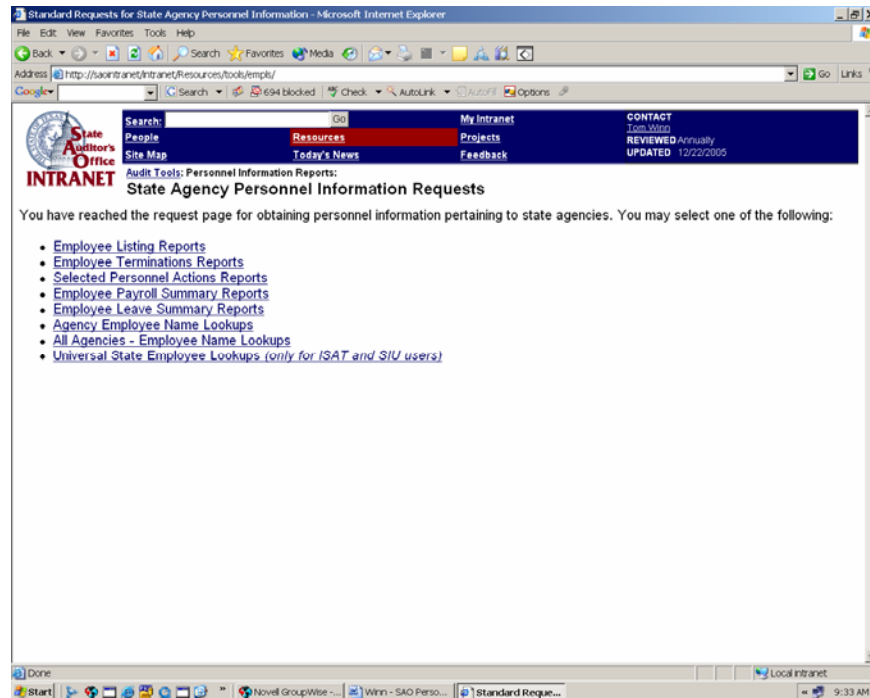
The Personnel Information Reports intranet application is made possible by SAS/IntrNet software, using Compute Services with the Application Dispatcher. Application Dispatcher is composed of two pieces: (1) the Application Broker, which is a Common Gateway Interface (CGI) program that resides on the web server, and communicates between the browser and the SAS Application Server, and (2) the SAS/IntrNet compute server, which is a SAS program on the SAS Server. Since the SAO's has SAS/SHARE installed on the SAS Server, it also would be possible for the SAO to develop SAS/IntrNet applications that make use of Data Services through execution of the htmSQL CGI program on the Web server; however, the Personnel Information Reports application does not use htmSQL. *[Note: certain other SAS/IntrNet applications at the State Auditor's Office use htmSQL.]*

The Personnel Information Reports application is available to all SAO employees by following a straightforward path from any web page on our intranet. In the application, requestors are prompted to make selections from a web page, and ultimately from one of several HTML forms. The parameters from the HTML form are passed to a SAS program as values of macro variables, and the program executes a query and generates HTML output, which is sent back to the requestor. All of the HTML documents, are kept in a single folder on the web server. They are retrieved from the web server by referencing their URLs. The SAS programs and data are stored on the SAS Server.

The web pages pertaining to the data requests were constructed using Notepad and Macromedia Dreamweaver MX software. Like all SAO internet/intranet applications, the Personnel Information Reports application conforms to agency standards for web applications. For example, a particular style sheet is used repeatedly by all of the HTML files which generate data requests. It provides a standardized header, as well as some links to certain other SAO web pages. There also are standards which pertain to the coding and formatting of web content. For example, we are prohibited from using frames, and also from using certain colors.

## OPERATION OF THE PERSONNEL INFORMATION REPORTS APPLICATION

Here is the main request page for the application:



The main request page presents the user with links to several data request pages. Let us suppose that the user is interested in reports containing an employee listing for a specific agency and as-of date. So, for example, if the user clicked on the link for "Employee Listing Reports", then the following HTML form would be returned:

The screenshot shows the 'Employee List Request Form' page. It contains the following sections:

- 1. Select Agency:** A dropdown menu labeled 'Select an Agency from the Listbox Below'.
- 2. Select as-of date:** A dropdown menu labeled 'Select a Date from the listbox below' with options: 30Nov2001, 28Feb2002, 31May2002.
- 3. Sort by:** A dropdown menu labeled 'Select a sorting variable from the listbox below' with options: Employee Name, Monthly Salary Amount.
- 4. Output format:** Radio buttons for 'HTML' (selected) and 'Comma-Separated Values'.
- 5. Generate the report or reset the values as necessary:** 'Generate Report' and 'Reset Values' buttons.

At the bottom, contact information for Tom Winn, Information Systems Support Team, is provided: phone: (512) 936-9735, e-mail: twinn@sao.state.tx.us.

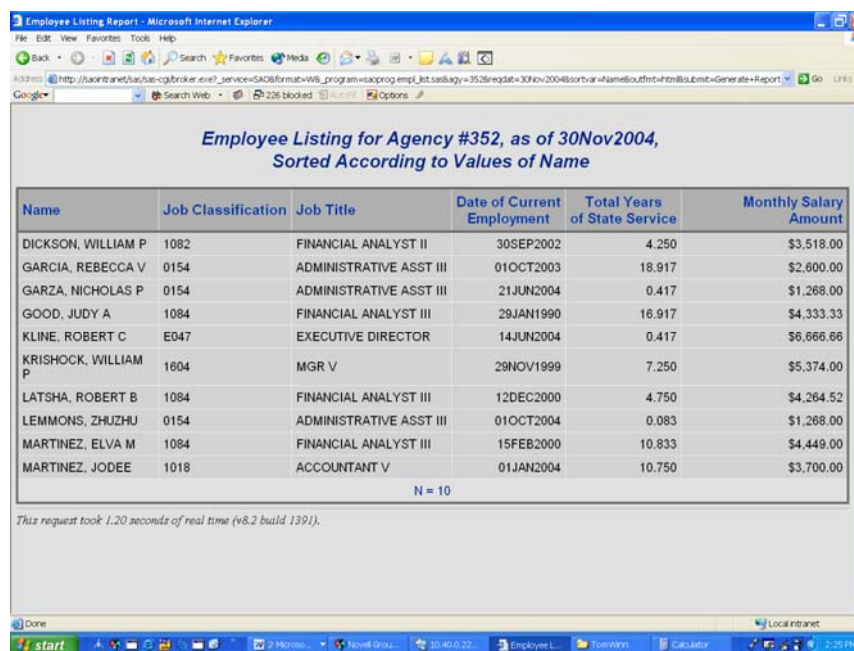
The user would make his/her selections from the drop-down boxes, and then would click on the "Generate Report" button. At this point, the important details for the specific data request would be sent from the Web browser to the

SAS Application Broker CGI program, which runs on the Web server. Then, the Application Broker would access the SAS Application Server, and would pass the parameters for the user's particular selections to the appropriate SAS programs.

In the Personnel Information Reports application, all data requests are handled dynamically, which means that the application does not include a collection of pre-prepared reports, covering all possible requests, stored as HTML documents, which would be displayed whenever one of them was requested. The application could have been designed to work that way, but it wasn't -- because I didn't want to have to manage a very large collection of separate, static reports. Instead, each data request results in the execution of SAS programs which extract the data which are pertinent to the request from appropriate SAS data sets, and then create HTML web content using the SAS Output Delivery System (ODS).

If I had decided to build this application using static web pages, then accommodating just five fiscal years of data for the Personnel Information Reports application would have required hundreds of thousands of static web pages!

Let us suppose that the user made the following selections for his/her data request: "Agency 352 – Texas Bond Review Board", employees as of "30Nov2004", sorted according to "Employee Name", and with the output to be formatted as HTML. The next display that the user would see would be the following web page:



**Employee Listing for Agency #352, as of 30Nov2004,  
Sorted According to Values of Name**

Name	Job Classification	Job Title	Date of Current Employment	Total Years of State Service	Monthly Salary Amount
DICKSON, WILLIAM P	1082	FINANCIAL ANALYST II	30SEP2002	4.250	\$3,518.00
GARCIA, REBECCA V	0154	ADMINISTRATIVE ASST III	01OCT2003	18.917	\$2,600.00
GARZA, NICHOLAS P	0154	ADMINISTRATIVE ASST III	21JUN2004	0.417	\$1,268.00
GOOD, JUDY A	1084	FINANCIAL ANALYST III	29JAN1990	16.917	\$4,333.33
KLINE, ROBERT C	E047	EXECUTIVE DIRECTOR	14JUN2004	0.417	\$6,666.66
KRISHOCK, WILLIAM P	1604	MGR V	29NOV1999	7.250	\$5,374.00
LATSHA, ROBERT B	1084	FINANCIAL ANALYST III	12DEC2000	4.750	\$4,264.52
LEMMONS, ZHUZHU	0154	ADMINISTRATIVE ASST III	01OCT2004	0.083	\$1,268.00
MARTINEZ, ELVA M	1084	FINANCIAL ANALYST III	15FEB2000	10.833	\$4,449.00
MARTINEZ, JODEE	1018	ACCOUNTANT V	01JAN2004	10.750	\$3,700.00

N = 10

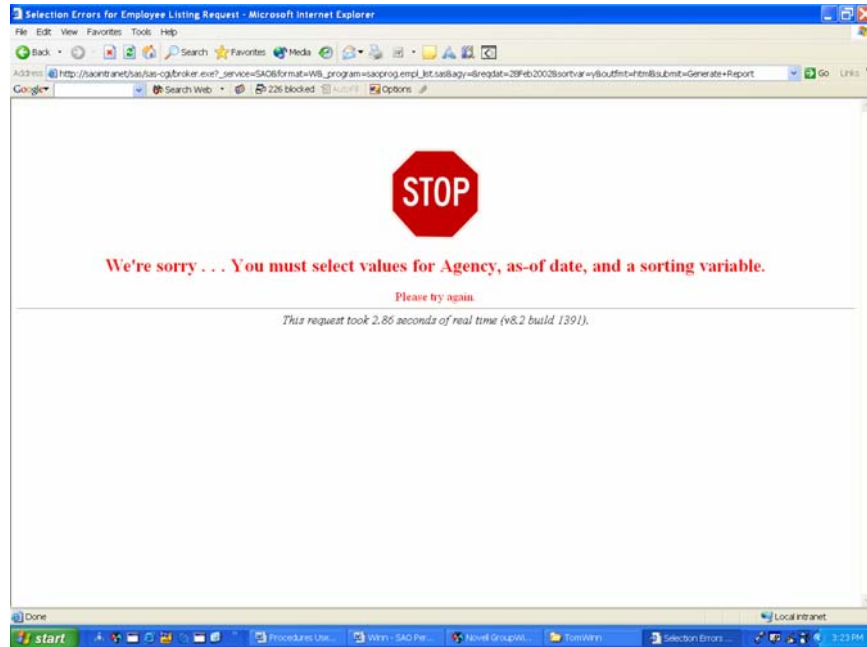
This request took 1.20 seconds of real time (v8.2 build 1391).

Notice the URL in the "Address" line near the top of the Internet Explorer window.

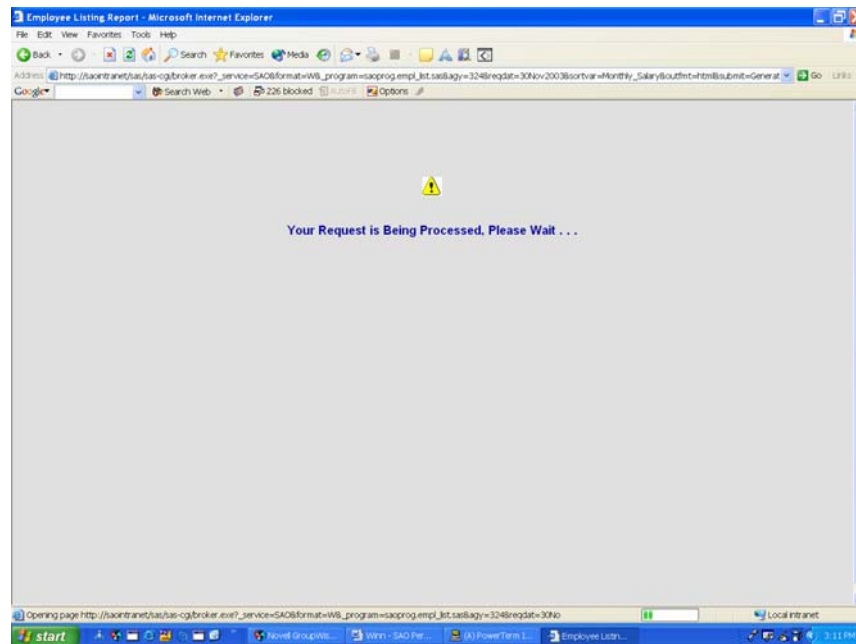
Address [http://saoitrinet/sas/sas-cgi/broker.exe?\\_service=SAO&format=W&\\_program=saoprogram empl\\_list.sas&agy=352&reqdat=30Nov2004&sortvar=Name&outfmt=html&submit=Generate+Report](http://saoitrinet/sas/sas-cgi/broker.exe?_service=SAO&format=W&_program=saoprogram empl_list.sas&agy=352&reqdat=30Nov2004&sortvar=Name&outfmt=html&submit=Generate+Report)

This line tells us that the SAS program launched on the SAS Server is empl\_list.sas. The URL also contains the values of parameters that pertain to the user's specific selections for this query, "agy", "reqdat", "sortvar", and "outfmt". In the request, agy=352 identifies Agency #352, which is the Texas Bond Review Board, the "as-of-date" reqdat=30Nov2004, the sorting-variable sortvar=Name, and the output-format outfmt=html. Those parameters were inserted into the particular SAS program, empl\_list.sas, as values of macro variables, before it was executed. The program generated a standardized report containing data items have been found to be useful for most data requests of the same type. The program used ODS to write the output to "\_webout", which specifies that the results are to be sent directly to the Web browser. I could have hidden those value-pairs in the browser address line, by changing the METHOD=GET attribute to METHOD=POST in the FORM tag of the HTML that was sent to SAS by the web server.

The SAS program checks the values of the parameters that are sent to it. If there are no user-specified values for agy, reqdat, or sortvar, then the following page is sent to the user.

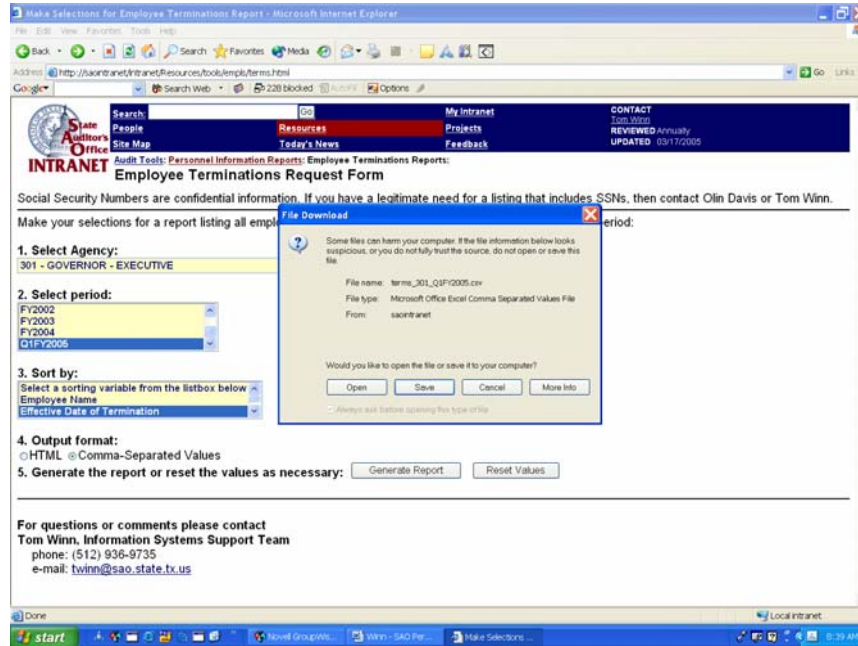


After the user submits a request, SAS extracts the data from the tables. This generally doesn't take too long. After the data are selected, but just before the desired report is generated, SAS sends a message back to the browser to run some Javascript code that asks for the user to be patient while the report is being formatted. For short queries, the user may never actually see the "Please Wait"-message. But for longer-running requests the user will see the following screen, until SAS completes the report and sends it to the browser:



I have demonstrated that the application can generate output reports as HTML. If desired, the user could have selected an output file of comma-separated values (CSV), instead. A CSV file is a presentation of a table as a series of ASCII text lines, in which each column value in the table row is separated from the value in the next column by a comma, and each table row is a new line. CSV files can be read by various application programs, including Microsoft Excel, Microsoft Access, and ACL. An example of how the Personnel Information Reports application can be used

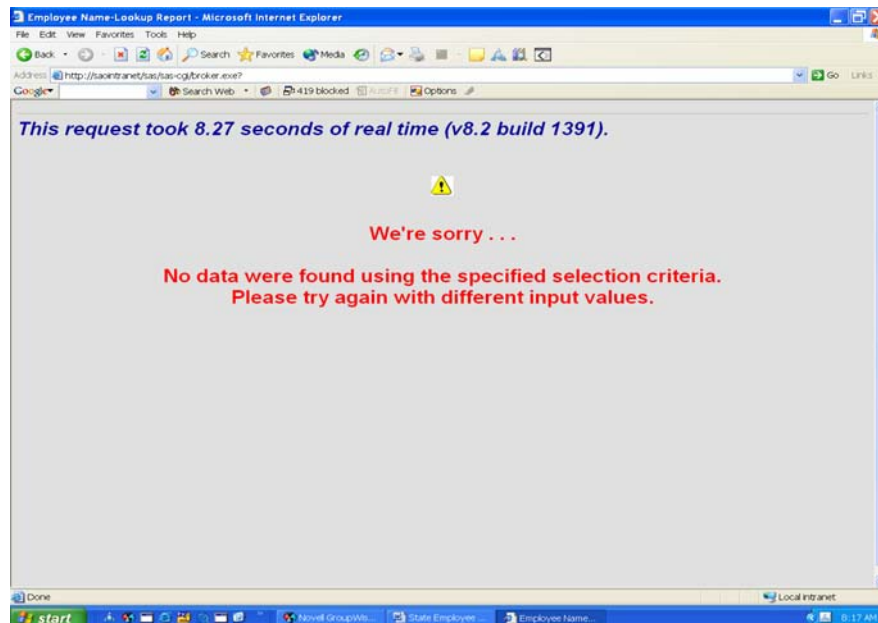
to generate CSV files follows. Notice that the name of the CSV file that is created for download includes elements which identify the type of request (empls, terms, actions, payroll, leave), the agency number, and the period.



The screenshot shows a Microsoft Excel spreadsheet titled "Microsoft Excel - terms\_301\_Q1FY2005(1).csv". The spreadsheet contains a table with the following data:

	A	B	C	D	E	F	G	H
	Effective Date	Name	Job Class/Job Title	Date of Last Employment	Reason Code	Reason		Amount of M
2	9-Sep-04	TOOMEY, MICHAEL V	1630 DEPUTY DIRECTOR	1-Dec-02	69	Retirement		\$11,250.00
3	16-Sep-04	ALONSO, MARIA T	279 WEB ADMIN III	16-Aug-04	67	Dismissal for cause		\$5,348.00
4	30-Sep-04	WATKINS, CATHRYN L	800 RESEARCH SPEC I	6-Jan-03	60	Voluntary termination		\$2,589.00
5	30-Sep-04	VALENTINE, JULIE A	1575 PROG SPEC VI	6-Dec-04	69	Retirement		\$5,736.66
6	17-Oct-04	POINSETT, ROYCE P	3505 ATTORNEY V	15-Oct-01	65	Transfer to another state agency/institution		\$6,812.00
7	10-Nov-04	MEDINA, DAVID M	1624 DIRECTOR V	1-Jan-04	65	Transfer to another state agency/institution		\$11,180.00
8	11-Nov-04	GARCIA, HECTOR S	1621 DIRECTOR II	1-Jan-01	65	Transfer to another state agency/institution		\$6,666.67
9	30-Nov-04	KIMBROUGH, JAY T	1624 DIRECTOR V	7-Sep-04	69	Retirement		\$10,000.00
10	30-Nov-04	GONZALES JR, ALEJANDRO	1014 ACCOUNTANT III	12-Mar-01	65	Transfer to another state agency/institution		\$2,912.00
11	30-Nov-04	GAONA, CYNTHIA B	156 ADMINISTRATIVE ASST IV	23-Jun-99	65	Transfer to another state agency/institution		\$3,010.42

If there are no data which satisfy the user's request, then the following page would be sent to the user.



### MISCELLANEOUS REMARKS

During development and testing of the application, a very useful technique for debugging SAS programs invoked through the Application Dispatcher was to append the debug flag and associated parameter value "&\_debug=131" to the URL. This returns all values passed to the SAS Server, the SAS Log, and the total elapsed time.

The Personnel Information Reports intranet application does not include data analysis capabilities, and no customizable details pertaining to the data products themselves. Providing those functionalities would have required a more complicated application than the one that I developed. However, our auditors prefer performing their own data analysis using other software tools (currently Microsoft Excel, Microsoft Access, and ACL). Therefore, the Personnel Information Reports application seems to meet their needs very well. It is easy to use, and it includes the most-commonly requested types of personnel information needed by auditors (employee listings, as well as terminations, acquisitions, promotions, demotions, merit increases, reclassifications, salary actions, and various other personnel actions, and name lookups). Of course, not all of our audits involve personnel information.

The Personnel Information Reports application is a "work in progress", and making additions and improvements to it is an ongoing task.

### CONCLUSION

SAS/IntrNet software was used to create an intranet application for auditors and investigators (non-SAS programmers) for creating standardized data products from a database of personnel information. This paper provided descriptions of the development and operation of the Personnel Information Reports intranet application.

### REFERENCES

Teresia Arthur & Mary Jafri, "Web Enable Your SAS Applications," Proceedings of the 28<sup>th</sup> Annual SAS Users Group International Conference (2003), Paper 35-28.

Kirby Cossey, Olin Davis, and Tom Winn, "Using SAS/IntrNet to Generate Data Products from a Database: The SAO Online USAS Guide," Proceedings of SCSUG 2004, pp. 113-125 (sponsored by the South-Central SAS Users' Group, on November 7-9, 2004).

Keith Cranford & Dan Hammarstrom, "Tricks and Tips with SAS/IntrNet," Proceedings of SCSUG 2003, pp. 70-83 (sponsored by South-Central SAS Users' Group, on October 26-28, 2003).

Kevin Davidson, "Using SAS/IntrNet Software," SSU 2001 Proceedings, pp. 204-210 (sponsored by the SouthEast SAS Users Group, and the South-Central SAS Users' Group, on August 19-22, 2001).

Matthew Grover, "SAS Solutions for the Web: Static and Dynamic Alternatives," Proceedings of SCSUG 2003, pp. 167-178 (sponsored by South-Central SAS Users' Group, on October 26-28, 2003).

Lauren Haworth, "HTML for the SAS Programmer," Proceedings of the 26<sup>th</sup> Annual SAS Users Group International Conference (2000), pp. 235-255.

Lauren E. Haworth, "HTML Output," Chapter 3 in Output Delivery System: The Basics, Cary, NC: SAS Institute Inc., 2001.

Doyle McDonald & Chas Webb, "SAS/IntrNet Configuration: Deploying a Web-Enabled SAS Environment," Proceedings of the Louisiana SAS Users Conference, pp. 77-140 (sponsored by South-Central SAS Users' Group, on June 16, 2003).

Frederick E Pratter, Web Development with SAS by Example, Cary, NC: SAS Institute Inc., 2003.

### **CONTACT INFORMATION**

Your comments and questions are valued and encouraged. You may contact the author at:

Tom Winn, Ph.D.  
Senior Systems Analyst  
Texas State Auditor's Office  
P.O. Box 12067  
Austin, TX 78711-2067  
Phone: 512 / 936-9735  
E-mail: twin@sao.state.tx.us

SAS and all other SAS Institute Inc. product or service names are registered trademarks or trademarks of SAS Institute Inc. in the USA and other countries. ® indicates USA registration.

Other brand and product names are trademarks of their respective companies.