Communicating Data Effectively with e-Data and e-Notes

Sy Truong, Meta-Xceed, Inc., Milpitas, CA

Abstract

Wouldn't it be nice to share SAS data with anyone in the world even if they don't have SAS? Delivering and communicating information pertaining to SAS data across the Internet can be a challenge. The data may be too large to be delivered across slow modem connections. Even if the data were successfully compressed and sent via email, it is difficult to discuss and view the details of the data if team members have different ways of browsing the data. This paper will explore the use of two new tools, *e-Data* and *e-Notes*, which combine the ability to view SAS data and conduct discussions over the Internet. These tools are available over the Internet and do not require SAS to view data, so they are ideal for delivering data to large audiences. They have features similar to SAS System Viewer® which enables quick access and search capabilities of SAS data. *e-Notes* combines Internet discussion group and bulletin board technologies to facilitate optimal communication pertinent to the data while maintaining an audit trail of the discussion. Information is often held by a small group of power users. This paper will show how *e-Data* and *e-Notes* can close the gap between data managers and end users by enabling the delivery of information to anyone over the Internet.

Introduction

Data is the lifeblood that drives the new information intensive economy. Transforming data into useful information for a specific business objective is key and SAS is the best tool for accomplishing this. There are many tools which SAS delivers to help users manipulate and analyze data. Even though the data may be entered and queried through a relational database, the best tools to bring meaning from data remains in the domain of SAS tools. Since SAS is the preferred tool, it is more efficient to access data in SAS format during the critical stages of analysis and data warehousing. The SAS system continues to grow in complexity as it delivers more functionality to users. The many features of SAS have made it one of the most powerful and successful software systems. However, the sophistication of the system has also hindered its proliferation among the masses since there is a steep learning curve for novice or non-technical users. This chasm has limited the full potential of SAS data since it is refrained to a small audience of only a few power users. In order for an organization to effectively meet its business objectives, it must deliver SAS data to all its team members. The challenge of delivering data from a sophisticated SAS system to novice end users is what this paper will focus on. It will describe how the use of a web browser as a common tool, combined with tools developed with SAS/IntrNet® can close this gap between

various users to form a cohesive team working and sharing data with greater effective collaboration.

Email as Information Delivery

Only ten years ago, memos were still being typed onto physical pieces of paper as a form of communicating information. Memos had the ability to display detailed numbers and figures which could communicate data more effectively compared to verbal face-to-face meetings. The memo has since been usurped by email since email is more cost effective and instantaneous. Messages containing time-dependent information can be instantly delivered across the world through email faster than any paper memo. This has empowered a new class of users and has taken information delivery to a higher level than before.

Email has its advantages in communicating interpersonal and detailed data but it also has its drawbacks. Some of the pitfalls of using email as a form of delivering and sharing data include:

- Inability to scale to large data sets. Email clients and servers are designed to handle moderately sized data. When a large database is attached to an email message, the system breaks down.
- Viewing data as email attachments is not universal among all users. If the sender has SAS on their machine and the recipient does not, the attached SAS data may not be accessible.
- The lack of a centralized audit trial. Recipients may have old emails in their inboxes but there is no global or project based collection of the communication.

The shortcomings of email as a data delivery and communication tool have limited its effectiveness. However, email remains a versatile and popular way of communicating small sets of data. If email were to be extended to include features which resolve its limitations, it would prove to be an effective tool for all users.

What is *e-Data*

e-Data originated as a web-based utility developed as a module of the Trialex System™. The goals and requirements for *e-Data* included:

- Enable users to securely view large SAS data sets through a browser even with a low bandwidth network connection.
- Browse contents and attributes of SAS data from a web client which does not necessarily have SAS installed.

- Contain search and sort capabilities for users to quickly get to specific data values.
- View SAS specific data features such as SAS formatted data or SAS labeled variables.

The system is designed for a team to share SAS data in a clinical environment where not all team members are familiar with SAS or have SAS on their desktop. Although *e-Data* was originally developed to view clinical data, it is a general tool to view any SAS data.

e-Data and e-Notes Server
Web Server
SAS Data
Client Browser

e-Data is implemented within a thin client architecture which allows SAS data to be viewed by any user who has access to the network via a web browser. This can be through an intranet or through the Internet. It is optimized to handle large data sets so data is readily available even with slow modem connections across the Internet.

A sample view of a data set named *a_ae* is shown here:
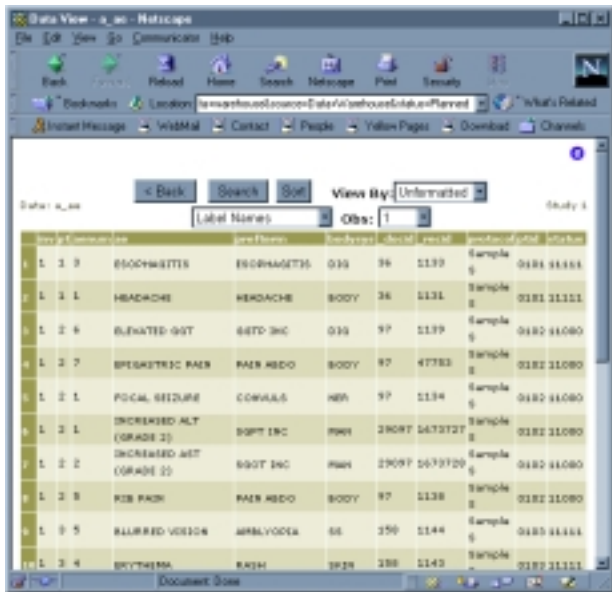
The data values are displayed in a grid style spreadsheet similar to the display on an application such as Microsoft Excel. The distinction between *e-Data* and a spreadsheet is that the information is delivered through a web browser. This poses some unique challenges. Web browsers are designed to receive and display a sequence of small packets of information. This allows browsers to perform

effectively even in a low bandwidth situation. *e-Data* was implemented with this in mind, so it segments the data view into blocks of 20 observations. At the bottom of the view, a navigational hyperlink allows users to flip to the next 20 observations.
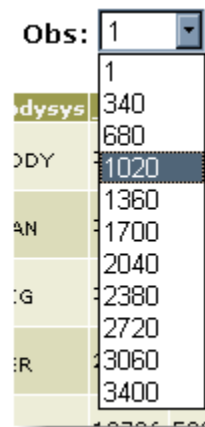

In a similar way, if the user is no longer at the first page, a new navigational hyperlink appears at the top.

This approach of segmenting the data into blocks of 20 observations allows the user to quickly navigate at 20 observation increments. In the scenario where the data set is large, an additional pull-down menu is supplied.

The *observation* menu divides the data into sizable chunks so users can select the specific "chunk" of data which contains the desired data point. This navigational menu combined with the *previous* and *next* links allows one to quickly navigate across the data.

Navigating through the data sequentially is more effective if the data is sorted properly. The sort button allows users to sort the data by up to three index variables.

Data: a_ae Sort
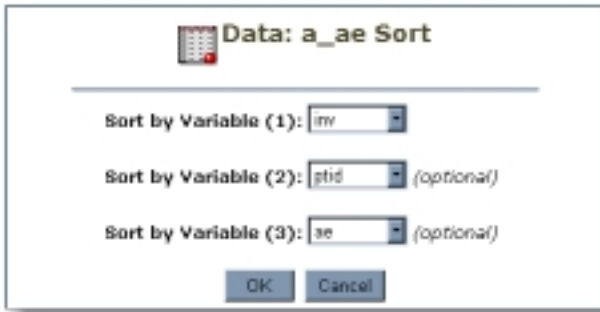
The *index sort variables* are similar to applying a PROC SORT to the data before viewing it. Once the sort variables are selected, the view will be shown with this new order, as shown here:

Sorted views are also very useful when applied in conjunction with a search. Applying a search is accomplished through the search dialog box accessed through the search button.

The search criteria are placed on top of the sort order. The resulting view will contain values which match the search criteria along with being sorted.

The search features can surface data points to non-SAS users who would not otherwise be able to easily find what

they need. Since the data is delivered in a web browser, a couple of other features are inherently available including:

- Cutting and pasting data points from the browser to other application on the desktop.
- Printing the current view through the browser.

The combination of these features allows users to find specific data points rapidly without having to understand the intricacies of the SAS system. The web interface is not only simple for the users but also makes it more universal since a web client could be an Intel based PC or a Mac or a UNIX work station. The ease of use and the universal access closes the information gap for all users.

What are *e-Notes*

e-Note is a collaborative communication tool similar to email designed to manage discussion threads. It allows a team to work together by conducting communication threads though email. Similar to *e-Data*, *e-Notes* was originally developed as a module to the Trialex System intended to mange the communication of FDA submission reports. *e-Notes* has been adapted to facilitate the communications pertaining to SAS data. Some of the objectives and features of *e-Notes* include:

- Enabling users to enter and send electronic notes pertaining to a specific data view among team members.
- Keeping a centralized searchable audit trail of the notes entered.
- Organizing the discussion thread for easy review including the snapshot view of the data pertaining to the discussion.

e-Notes are similar to bulletin board discussions on the Internet but it is unique in that its discussions are linked to the data and therefore the conversation is context sensitive to the data.

The main review screen for *e-Notes* lists all discussions as they pertains to a particular topic. This view is sorted by the latest post date and is collapsed by subject.

The subject describes the specific data and observation of discussions. The subject also acts as a drill down link to the specific *e-Note* of the discussion thread. The following view shows an excerpt of the data *a_ae* discussion thread.

Author	Subject: Data: a_ae, obs(20-40)	
1 Author: Eugene Smith Programmer Analyst Sent to: David Foresight, Eugene Smith, Sy Truong CC:	posted 06/01/2000, 7:59:46 pm	Retire Note: ☐
	Hi David, The data value for patient 5 is entered as EDEMA. I think this is incorrectly entered since the case report form seems to indicate that it should be: "BILATERAL KNEE". Please review and suggest corrections. Eugene	
2 Author: Sy Truong Administrator Sent to: Eugene Smith, Marilyn Feather, Sy Truong CC:	posted 06/15/2000, 9:24:26 am	Retire Note: ☐
	Hi Eugene, Thanks for pointing this out. I have found the same discrepancy and I will look into this. Thanks, Sy	
3 Author: David Foresight Sent to: Eugene Smith, Marilyn Feather, David Foresight, Sy Truong CC:	posted 09/01/2000, 4:12:27 pm	Retire Note: ☐
	Hi Sy, It turns out that this is a a data entry problem and I will have to send this back to the site for further investigation. David	
4 Author: Sy Truong	posted 09/01/2000, 4:33:32 pm	Retire Note: ☐

If the user were to scroll down to the bottom of the discussion thread, the report will also display the specific *data snapshot view* of interest.

The centralized audit trail keeps a log of the conversation and associates it with the data view so that the discussions are placed in the proper context. This adds much more meaning to discussions since it references specific data points.

Aside from displaying a list of all the discussion threads,

users can search for specific words which appear in any of the discussion notes.

The search capability allows users to find any specific discussion points of interests. An example could be a search for the follow-up on all data discrepancy issues.

Each note has a status associated with it. Once it is set to retired, it is no longer available in the main review screens. However, this does not delete retired notes, since all retired notes can still be surfaced through a search.

An entirely new note can be added with a different subject matter or a note can be added to an existing discussion thread. The interface is similar to email as shown here:


This example is a start of a new *e-Note* discussion, which is initialized for the termination data. The subject matter defaults to the current data view to save users from additional typing. The subject name is significant since it is how the discussion threads are identified and grouped.

There is nothing ground breaking in the way *e-Notes*

4

facilitates the communication about data through capturing the user's notes and sending it to the recipient as an email message. However, it is worth noting that the messages are populated in a central database and therefore maintain a history of the discussion. This documentation is important in discussions which lead to changes to the data. A database may be able to track what data values have changed, but *e-Notes* along with *e-Data* can capture the reason why the data values have changed. If the team has a high turn over in membership, this "project memory" plays a significant role in maintaining data integrity and "group intelligence".

Communication Work Flow

These tools were developed with the SAS/IntrNet software. This was the natural choice since the underlying data which it accesses is SAS data. The thin client architecture has the following workflow:
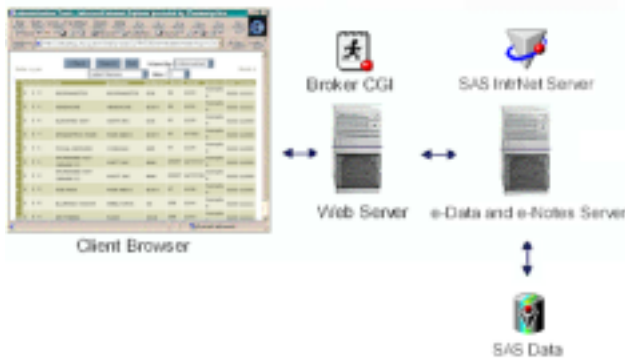
The only requirement on the user's machine is a web browser. It communicates to the web server via the standard TCP/IP network protocol. This is no different from accessing a static website on the Internet except for the fact that the web page is dynamically generated through SAS/Intrnet. The web server in turn passes the user's requests to the application server running SAS/IntrNet. *e-Data* and *e-Notes* handle these requests and display the SAS data back to the users in HTML format. This methodology and workflow is typical of how a SAS/IntrNet application is developed.

How does e-Data and e-Notes Work?

e-Data and *e-Notes* were developed primarily with SCL since this is a versatile and efficient way of accessing SAS data and generating HTML in response to user requests. In order for pages to be interactive, HTML form fields are used to store key information. Consider the following screen of *e-Data*:

Within this page, there are many HTML form fields which store information such as the program to run and the button users clicked on. The HTML code for these fields looks like:

```
<form name="myForm"
 action="/cgi-bin/broker.exe">
  <input type="hidden"
   name="_PROGRAM" value=
   "edata.edata.datview2.scl">
  ...
  <input type="button"
   value="&lt; Back" name="b_back">
   
  <input type="button" value="Search"
   name="b_search"> 
  <input  type="button" value="Sort"
   name="b_sort">

</form>
```

The specified program is a SCL entry named *dataview2.scl*. It is stored in a catalog named *edata* and is defined as a hidden field. There are many fields similar to the program field, which do not have a display component. This illustrates how the system keeps track of which program to run without displaying anything on the screen. On the other hand, buttons have a display component so the *back*, *search* and *sort* buttons are defined as "button" types accordingly.

The fields in the HTML forms are passed to the broker CGI program which is a component of the SAS/IntrNet software. This software reinterprets the HTML form fields and converts them into SAS macro variables. For example, the back button named *b_back* can be captured in a SAS variable by the following code:

```
*** Capture the form fields ***;
b_back = symget('b_back');
```

In this example, the new SCL variable named *b_back* will contain the value of the button pressed. This value will determine the action taken to display the next screen according to the user's request. The concept of navigating from one page to the next is a common occurrence on a website. The difference is that the HTML page is not a static page on a web server, but rather it is dynamically redrawn through an algorithm within a SAS program.

SAS Dynamic HTML

One interpretation of "dynamic HTML" is through the use of JavaScript or XML. This enables the HTML page to be dynamically redrawn depending on the data which it is displaying or depending on the users interactions with the page. Another approach for dynamic HTML is "SAS dynamic HTML". The HTML page is redrawn using SAS algorithms rather than scripting language such as JavaScript. For example, the first view of the data set containing the first 20 observations may look like:

	inv	pt	aenum	ae
1	1	1	3	ESOPHAGITIS
2	1	1	1	HEADACHE
3	1	2	6	ELEVATED GGT
4	1	2	7	EPIGASTRIC PAIN
5	1	2	1	FOCAL SEIZURE
6	1	2	1	INCREASED ALT (GRADE 2)
7	1	2	2	INCREASED AST (GRADE 2)

This is essentially an HTML table with cells containing the data values. The use of alternating background cell colors helps clarify the view of the data. This page is dynamically redrawn with the proper 20 observations per user request. The redrawing of the screen to generate the desired view is within a SAS program. This method of creating SAS dynamic HTML proves to be much more powerful than other languages, especially when it comes to accessing SAS data. SAS supports extensive data manipulation and analysis tools along with a rich set of text functions to form the ultimate dynamic HTML engine.

At first glance, a traditional website may appear similar to one generated with SAS dynamic HTML. They both are accessed through a web browser containing visual navigational buttons. With further investigation, it becomes apparent that the interactivity of the website with SAS proves to be much more functional compared to a static HTML page. This comparison is analogous to the method of accessing data stored on physical disk using disk operating system commands, compared to a more sophisticated relational database system.

What State are You In?

The concept of state is a unique challenge specific to web application development. This is due to the nature of how web servers work. A typical request is normally initiated from a web browser to a web server. The web server responds to the request and then releases the user.

Client Browser — Web Server

Once the server releases the user, it knows nothing about what the user is doing until the next request. This disconnect creates a challenge since the system does not know where the user is coming from or what "state" the user is in, unless it finds alternative ways of keeping track.

e-Notes and *e-Data* use several techniques to keep track of the user's state. These include:

- Storing a data base of all users and their current state
- Use of HTML fields to track users
- Storing a profile of user specific information

The moment the user logs in, the system starts to track the user through a database. In this database, there is a unique session identification number kept for each request. A corresponding session ID is stored on the HTML page as a hidden field as shown in this example:

```
<input type="hidden" name="session"
 value="0.77135559903053">
```

Each time a new request is placed, the session field is compared to the database. This tracking allows the system to know the current location of the user. The combination of HTML hidden fields with a backend database is essential in managing users' states, especially when there are multiple users simultaneously using the system.

In conjunction with states, a user may have selected options on one page which affect the logic of another page at a later point. A user profile catalog is created in this case. Similar to how SAS has a user profile catalog for each user, *e-Notes* stores user preferences and selections in individual user profile catalogues.

Conclusion

Email has truly democratized communication around the world. It has transformed written communication and has brought about a new way of communicating information instantaneously. There still are challenges in communicating large sets of data even with the advent of email. *e-Data* and *e-Notes* are furthering the evolution of this technology to facilitate communication and

collaboration with the "stuff" which drives the information age, primarily data. It is important to bring about useful information from data, and the best way of accomplishing this is through tools like SAS. In addition, bringing value to data is to deliver it beyond the realm of data managers and analysts and into the hands of end users and decision makers. Data warehouses that drive pivotal business decisions of organizations are therefore no longer regulated to a few, but rather they can be shared among the masses. Once this is accomplished, the full potential of data can be realized.

Author:
Sy Truong
Meta-Xceed, Inc.
sy.truong@meta-x.com
http://www.meta-x.com