

## Paper CT-02

## An email macro: Exploring metadata EG and user credentials in Linux to automate email notifications

Jason Baucom, Ateb Inc., Raleigh, NC

### ABSTRACT

Enterprise Guide (EG) provides useful metadata variables to identify user credentials. Metadata can be exploited to send program notifications, but a more generic macro is required for usage in batch mode where user credentials may not be required. A flexible macro is presented that can extract email addresses by exploring metadata in an EG session or by identifying email addresses by using user credentials on a Linux system.

### INTRODUCTION

One useful utility SAS offers is the ability to send email from programs. This can be used to notify programmers of program status, send event based notifications or deliver program results and metrics to interested parties. The typical approach is demonstrated below:

```
FILENAME output EMAIL
SUBJECT="Errors detected"
FROM="admin@ateb.com"
      TO="jason.baucom@ateb.com";
DATA _NULL_;
FILE output;
PUT "We found errors...";
RUN;
```

The standard email utility is useful, but may require time consuming modification when code is frequently shared among team members. Our team found it bothersome to keep changing code each time an Enterprise Guide® project or code segment was shared, so we sought a general and flexible solution. We operate in a heterogeneous environment, running code in both batch mode from a Linux environment and Enterprise Guide projects. In Enterprise Guide metadata can be queried to identify email addresses. User credentials may not be established in Linux, so this scenario must be accounted for as well. The mail macro needs to first identify the operating environment, discover the associated email addresses associated with the user and then send then mail. The basic flow diagram of the email macro is below (Figure 1).

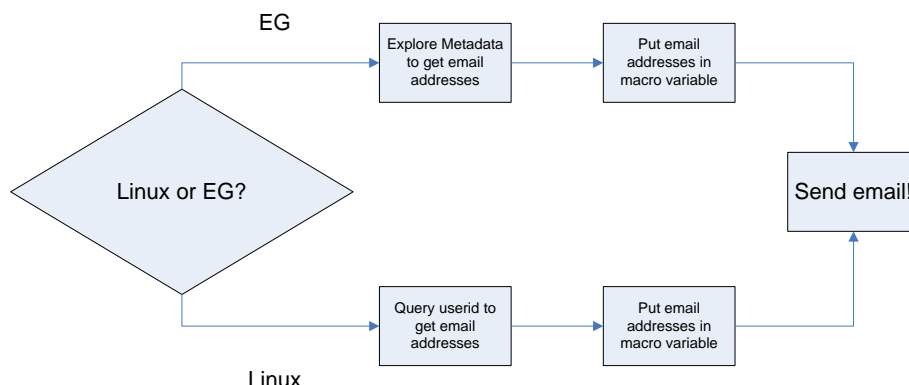


Figure 1. Flow diagram for email macro

### DETERMINING YOUR ENVIRONMENT

The first critical step is to determine in which environment the macro is executed. When a user signs into Enterprise Guide user credentials are established and can be used to determine the associated email addresses. When executing SAS code in batch mode the user may not have to establish user credentials, so we can't always rely on metadata to find our email addresses.

In the SAS environment the &SYSENV variable can be queried to determine if you are in batch mode or Enterprise Guide. We determined that &SYSENV is set to "FORE" when executing code in batch mode in Linux and "BACK" when running code through Enterprise Guide. The macro code is forked using an "if" statement

```
%if &SYSENV = FORE %then %do;
  %put In batch mode;
  /* Get email address in Linux */
%end;
%else %do;
  %put In EG mode;
  /* Get email addresses in EG */
%end;
```

## ESTABLISHING IDENTITY IN ENTERPRISE GUIDE

When user credentials are established in Enterprise Guide some environmental variables are set. The &\_metauser environment variable identifies the user and can be used to query the metadata server and determine the associated email addresses. To take advantage of this aspect of this macro your system administrator needs to setup user email accounts, which can be done through Management Console (Figure 2).

One very useful tool for manually exploring metadata trees is the Metadata Browser. This tool can navigate through metadata and identify where in metadata email addresses are stored. By proceeding from Person → Your User → EmailAddresses you can find your user email addresses, assuming that the admin has properly set them up (Figure 3).

In order to connect to and begin querying the metadata server we must first set some options:

```
options metaport=8561
  metaserver="talon2.ateb.com"
  metauser="sasadm@saspw"
  metapass="yourpasswd"
  metarepository="Foundation";
```

Once these options are set you can begin using metadata functions to navigate the metadata tree and obtain your target metadata element. In this case our target metadata element is the "Address" variable. Once obtained it will be stored in a data object and subsequently in a macro variable for later consumption. First, we set up the data step and obtain the metadata object corresponding to the user.

```
data emails;
  length uri address groupuri $100;
  call missing(uri, address, groupuri);
  nobj=metadata_getnobj("omsobj:Person?@Name = '&_metauser'",1,uri);
```

metadata\_getnobj will retrieve the Person object that has Name = &\_metauser. The second parameter tells the function to retrieve only the first matching object identified. The object obtained can be referred to using the uri variable. Nobj will hold the return value of this function, the number of records we retrieve. Nobj will be set to 0 if no matching Metadata records are found or to 1 if there is a single record. If a record is found the uri reference can be used to find associated objects using the metadata\_getnasn function.

```
if nobj=0 then put 'Person not identifiable.';
else do;
  a=1;
  grpassn=metadata_getnasn(uri,"EmailAddresses",a,groupuri);
```

E-mail	Type	Address
Phone	work	jason.baucom@ateb.com
Address	home	jasonbaucom@hotmail.com

Figure 2. Exploring Metadata using Metadata

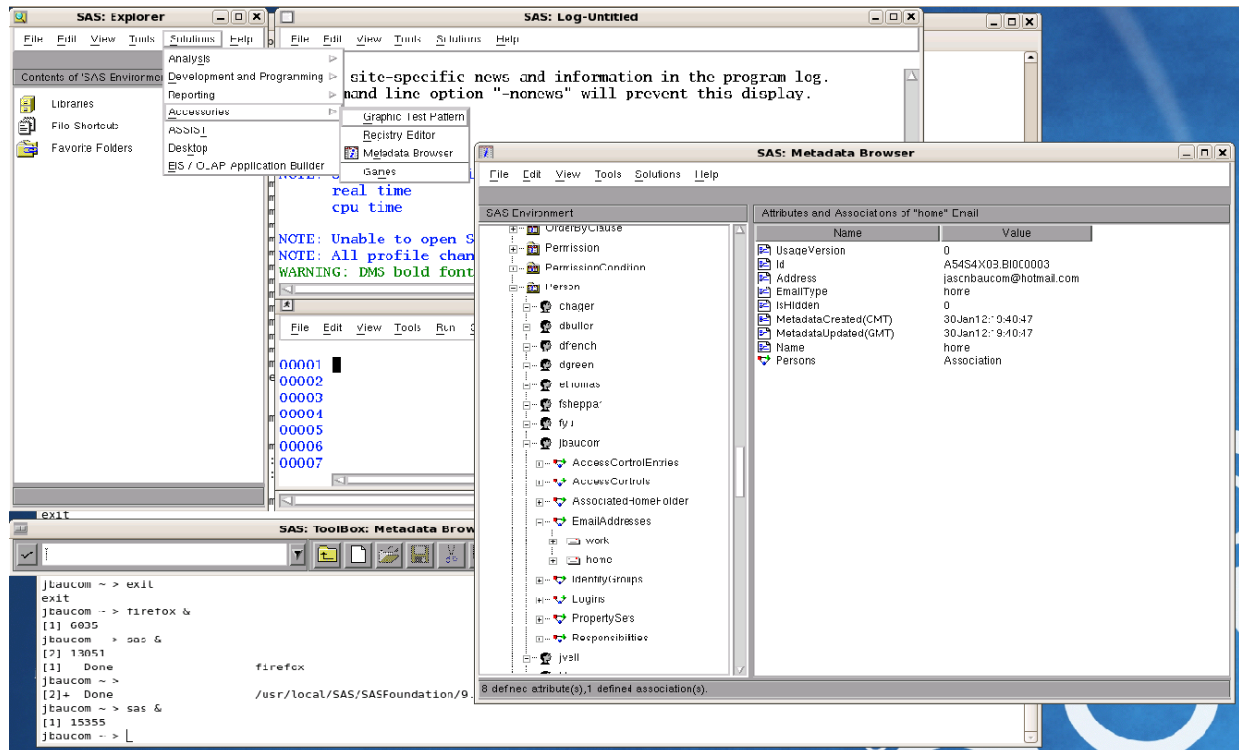


Figure 3. Exploring Metadata using Metadata

We are specifically interested in the "EmailAddresses" objects. A user might have multiple email addresses so the macro will need to loop over all possible email objects. The third parameter is utilized to identify which object to retrieve. We can loop over all email addresses using a variable as a counter. We initially look at the first EmailAddresses object associated with uri. If there is no email address the return code of metadata\_getasn will have the value of -3 or -4. Otherwise there is confidence that at least one email address exists. A default email address is used if no email address is found.

```
if grpassn in (-3,-4) then do; /* no email address */
  address="jasnbaucm@hotmail.com";
  output;
end;
else do while (grpassn > 0); /* loop over all available email addresses */
  rc=metadata_getattr(groupuri,"Address",address);
  put address=;
  output;
  a+1;
  grpassn=metadata_getasn(uri,"EmailAddresses",a,groupuri);
end;
```

The EmailAddresses object is stored in the groupuri object, so the address associated with that object can be extracted by using another metadata function.

```
rc=metadata_getattr(groupuri,"Address",address);
```

The metadata\_getattr function will place the value of the groupuri's "Address" attribute in the address variable. The address is stored in a dataset and the while loop continues in an attempt to identify any more email addresses. Once all email addresses associated with the user have been discovered our loop terminates with grpassn <=0 and the addresses are stored in a macro variable for later consumption.

```
proc sql noprint;
  select '||address||' into :emailVariable separated by " " from emails;
quit;
```

## ESTABLISHING IDENTITY IN LINUX

User credentials are not necessarily established in batch mode, so another method is required to determine email addresses. The Linux userid can be uniquely identified using the 'whoami' command. The result of a system command is stored in a macro variable for later use.

```
filename getuid pipe 'whoami';
data linuxusers;
    infile getuid lrecl=20 pad;
    input uid $20.;
run;
proc sql;
    select uid into :myID from linuxusers;
quit;
%put got &myID;
```

Since Linux accounts are not necessarily associated with a SAS account it is essential to establish a link between Linux userids and email addresses. This was hard coded in a data step.

```
data emails;
    length linuxUID $8. address $100.;
    linuxUID = "root"; address = "jason.baucom@ateb.com"; output;
    linuxUID = "dbuller"; address = "donovan.bullerwell@ateb.com"; output;
    linuxUID = "jbaucom"; address = "jason.baucom@ateb.com"; output;
    linuxUID = "jbaucom"; address = "jasonbaucom@hotmail.com"; output;
run;

proc sql;
    select '''||emailAddress||''' into :emailVariable separated by " " from
    emails where linuxUID = "&myID";
quit;
```

The end result is similar to our approach in EG, namely a macro variable that contains a list of email addresses. One potential drawback to this approach is that the macro will need to be actively maintained to reflect new users and email changes.

## TYING IT ALL TOGETHER AND SENDING THE EMAIL.

So far it has been determined which environment we are working in and the email addresses used in the active environment have been identified. The result of this effort is a list of email addresses stored in the emailVariable macro variable. Notifications may need to be sent to other email addresses, but this cannot be determined from metadata or Linux accounts, so this information must be passed into our macro by the user. The content of our message must also be passed to the macro as well. We wrap the contents of the macro with the following definition, passing a list of email addresses, subject and email content as parameters:

```
%macro mailteam(EmailList=, Subject=, Message=, Message2=, Message3=);
```

The EmailList parameter can contain a list of additional email addresses or be left blank. If there is no email associated with your account available in metadata you can manually enter your email address here. The Subject variable and up to three message lines can be used as well. Missing subject or message variables will simply be left blank in the outgoing email.

At the end of the macro we finally send our email:

```
filename mymail email to=(&EmailList %nrbquote(&emailVariable) )
    subject=&Subject;
data _null_;
    file mymail;
    put &Message;
    put &Message2;
    put &Message3;
run;
```

The "to" list is a combination of the emails identified from metadata or Linux and the list of email addresses we manually entered in the macro call.

The macro can be called in the following manner:

```
%mailteam(Subject="testing testing",Message="did it work?");  
  
%mailteam(EmailList="myotheremail@ateb.com", Subject="testing  
again",Message="did it work?",Message2="we need another line");
```

Using the EmailList variable can add additional non-identifiable email addresses to the message but it will not remove the default email addresses associated with the user account. This might be helpful if your admin has not assigned an email address to your account in Management Console or in the macro code.

## CONCLUSION

In conclusion a useful macro is presented that facilitates the sending of email messages in a shared code environment with minimal effort. It has the ability to send emails to addresses uncovered by querying metadata in Enterprise Guide, using system commands in Linux or by sending hard coded email addresses in the macro call. See appendix for a full code.

## REFERENCES

SAS Institute. "DATA Step Functions for Reading and Writing Metadata". Retrieved 08/01/2012. Available at <http://support.sas.com/documentation/cdl/en/lrmeta/60739/HTML/default/viewer.htm#a003091925.htm>.

## CONTACT INFORMATION

Your comments and questions are valued and encouraged. Contact the author at:

Name: Jason Baucom, PhD  
Enterprise: Ateb Inc.  
Address: 2600 Sumner Blvd  
City, State ZIP: Raleigh, NC  
Work Phone: 919-882-4990  
E-mail: Jason.baucom@ateb.com

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.

**APPENDIX 1. CODE LISTING**

```

%macro mailteam(EmailList=, Subject=, Message=, Message2=, Message3=);
  %let ourEmail=;
  %if &SYSENV = FORE %then %do; /* We are in batch/linux mode */
    filename getuid pipe 'whoami';
    data linuxusers;
      infile getuid lrecl=20 pad;
      input uid $20.;
    run;
    proc sql;
      select uid into :myID from linuxusers;
    quit;
    %put got &myID;

    data emails;
      length linuxUID $8. address $100.;
      linuxUID = "root"; address = "jason.baucom@ateb.com"; output;
      linuxUID = "dbuller"; address = "donovan.bullerwell@ateb.com"; output;
      linuxUID = "jbbaucom"; address = "jason.baucom@ateb.com"; output;
      linuxUID = "jbbaucom"; address = "jasonbaucom@hotmail.com"; output;
    run;

    proc sql;
      select '''||address||''' into :emailVariable separated by " " from emails
where linuxUID = "&myID";
    quit;
  %end;
  %else %do; /* We are in EG Mode */
    options metaserver="server.ateb.com"
      metaport=8561
      metauser="sasadm@saspw"
      metapass="sasadmpasswd"
      metarepository="Foundation";
    data emails;
      length uri address groupuri $100;
      call missing(uri, address, groupuri);
      nobj=metadata_getnobj("omsobj:Person?@Name = '&_metauser'",1,uri);
      if nobj=0 then put 'Person not identifiable.';
      else do;
        a=1;
        grpasnn=metadata_getnasn(uri,"EmailAddresses",a,groupuri);
        if grpasnn in (-3,-4) then do; /* no email address */
          address="jasonbaucom@hotmail.com"; /* default */
          output;
        end;
        else do while (grpasnn > 0); /* loop over email addresses */
          rc=metadata_getattr(groupuri,"Address",address);
          put address=;
          output;
          a+1;
          grpasnn=metadata_getnasn(uri,"EmailAddresses",a,groupuri);
        end;
      end;
      keep address;
    run;

    proc sql noprint;
      select '''||address||''' into :emailVariable separated by " " from emails;
    quit;
  %end;

```

```
filename mymail email to=(&EmailList %nrbrquote(&emailVariable) ) subject=&Subject;  
data _null_;  
    file mymail;  
    put &Message;  
    put &Message2;  
    put &Message3;  
run;  
%mend mailteam;
```