

Parsing Metadata Server Logs

Administrators may want to analyze their Metadata Server logs and programmatically take actions based on the parsed results. Note that the messages in the Metadata Server log have multiple sources, and the format of each line of the log is touched by several subsystems. The result is the messages may contain delimiters in some parts of the message. This makes it difficult to guarantee a consistent delimiter for specific sections of the message. The lack of a consistent delimiter increases the difficulty of writing a parsing program. The following information will facilitate parsing the Metadata Server log.

All log messages contain a date/time stamp and a connection number that uniquely identifies each client request. The rest of the line is a free-format message. The attached program will parse a Metadata Server log (or collection of logs if a wildcard is used in the INFILE statement) with separate data set variables for DATE, TIME, CONNECTN, and MSG. The program doesn't account for messages being split across multiple lines, so the Metadata Server should be run with **LINESIZE=MAX** set to use the program as-is. Depending on the end result being sought, more parsing could also be added. For example, the MSG often contains a client number, and the userid and domain with which the client connected to the server, in the form **nn:userid@domain:** which could be parsed from those lines that contain it.

The [first program](#) will parse one log file and create the four variables DATE, TIME, CONNECTN, and MSG. The [second program](#) parses all of the April 2005 logs in the subdirectory. The program identifies user id, connection status (accepted or rejected), connection number, peer address, peer port number and whether the connection was trusted or authenticated for New Connections. It also identifies user id and connection number for existing connections.

All messages are listed and can be analyzed for further parsing.

/* Simple Metadata Server log Parser program */

```
libname metalog 'C:\SAS\EntBIServer\Levl\SASMain';

options linesize=max;
data metalog.mslog;
/* Following path will get all logs for April 2005 */
infile
"C:\SAS\EntBIServer\Levl\SASMain\MetadataServer\logs\MetadataServer07Apr05.log"
    lrecl=32767 trunccover;
length date      8
        time      8
        connectn  8
        msg       $ 32000;
format  date      date11.2;
format  time      time9.;
length  record $   200;
informat record $char256.;
```

```

format record $char256.;
input @1 record;
drop record;
if ( length(_infile_) gt 32 ) then
  do;
    if ( substr(_infile_,1,2) eq '20' ) then
      do;
        msg = substr(_infile_,32);
        /* If the standard error about tcpSockRead is found just get next
record */
        if (msg="ERROR: The tcpSockRead call failed. The system error
is 'The connection was reset by a peer.'")
          then return;
        connectn = input( substr(_infile_,23,8),8.);
        date=input(scan(_infile_,1,':'),yymmdd8.);
        time=input(SCAN(_infile_,2,':'),time12.);
        /* determine if msg continues on next record */
        link checkmsg;
        /* output the current record */
        output;
        end;
      else
        do;
          if ( substr(_infile_,7,8) eq 'Log file' ) then
            do;
              put _infile_;
            end;
          end;
        end;
      end;
    return;
  checkmsg:
  do j2 = 1 to 100;
    drop j2;
    input @1 record @@;
    len=length(_infile_);
    drop len ;
    if (len ge 2) then
      /* If new record start return to main section */
      if ( substr(_infile_,1,2) eq '20' ) then return;
      /* if Log file record return to main section */
      if ( index(_infile_,'Log file') ) then return;
      msg=trim(msg)||left(_infile_);
      /* release the current record */
      input ;
    end;
  return;
run;
ods listing close;
filename htmlout temp;
ods html file=htmlout;
proc print;run;
ods html close;
ods listing;

```

/* Non-Simple Metadata Server log Parser program */

```
libname metalog 'C:\SAS\EntBIServer\Levl\SASMain';

options linesize=max;
data metalog.mslog;
  infile
"C:\SAS\EntBIServer\Levl\SASMain\MetadataServer\logs\MetadataServer27Apr05.log"
  lrecl=32767 truncover;
  length   date      8
          time      8
          connectn  8
          msg        $ 32000
          rstatus   $ 8
          user      $ 32
          ippport   $ 20;
  format   date      date9.;
  format   time      timel1.2;
  length   record   $ 200;
  informat record   $char200.;
  format   record   $char200.;
  input @1 record;
  userrec=0;
  drop record userm beg end connword;
  if ( length(_infile_) gt 32 ) then
    do;
      if ( substr(_infile_,1,2) eq '20' ) then
        do;
          msg      =      substr(_infile_,32);
          /* If the standard error about tcpSockRead is found just get next
record */
          if (msg="ERROR:  The tcpSockRead call failed. The system
error is 'The connection was reset by a peer.'")
            then return;
          connectn = input( substr(_infile_,23,8),8.);
          date=input(scan(_infile_,1,':'),yymmdd8.);
          time=input(SCAN(_infile_,2,':'),time12.);
          /* determine if New client connection record */
          userm=index(msg,"New client connection");
          if (userm) then
            do;
              userrec=1;
              beg=index(msg,'(')+1;
              end=index(msg,')');
              conn=substr(msg,beg,(end-beg));
              rstatus=scan(substr(msg,end+1),1);
              beg=index(msg,'for ')+3;
              user=scan(substr(msg,beg),1,'.');
              if (index(user,'trusted')) then
                trustedcon=1;
              else
                trustedcon=0;
              user=substr(user,index(user,'user ')+5);
              ippport=scan(msg,-1,' ');
```

```

        ipport=substr(ipport,1,length(ipport)-1);
    end;
    connword=scan(substr(msg,1,20),1,' ');
    if ((index(connword,':'))
        & not(index(connword,'NOTE:'))
        & not(index(connword,'ERROR:'))) then
    do;
        if (substr(connword,1,1)=':') then
            do;
                conn='';
            end;
        else
            conn=scan(msg,1,':');
            user=scan(msg,2,':');
        end;
        if (index(connword,'ERROR:')) then
            do;
                rstatus='ERROR';
            end;
        if (index(connword,'NOTE:')) then
            do;
                rstatus='NOTE';
            end;
        if (index(msg,'Client connection') &
            index(msg,'closed.')) then
            do;
                rstatus='Closed';
                conn=scan(msg,2,'()');
            end;
        /* determine if msg continues to new line */
        link checkmsg;
        output;
    end;
    else
        do;
            if ( substr(_infile_,7,8) eq 'Log file' ) then
                do;
                    put _infile_;
                end;
            end;
        end;
    return;
checkmsg:
    do j2 = 1 to 100;
        drop j2;
        input @1 record @@;
        len=length(_infile_);
        drop len ;
        if (len ge 2) then
            /* If new record start return to main section */
            if ( substr(_infile_,1,2) eq '20' ) then return;
            /* if Log file record return to main section */
            if ( index(_infile_,'Log file') ) then return;
        msg=trim(msg)||left(_infile_);
        /* release the current record */
        input ;
    end;
end;

```

```
end;  
return;  
run;  
ods listing close;  
filename htmlout temp;  
ods html file=htmlout;  
proc print;run;  
ods html close;  
ods listing;
```