



Startseite

# Yalan - Yet another Log Analyzer

15 October, 2005 - 19:07 — StephanFrenzel

Anhang

Größe

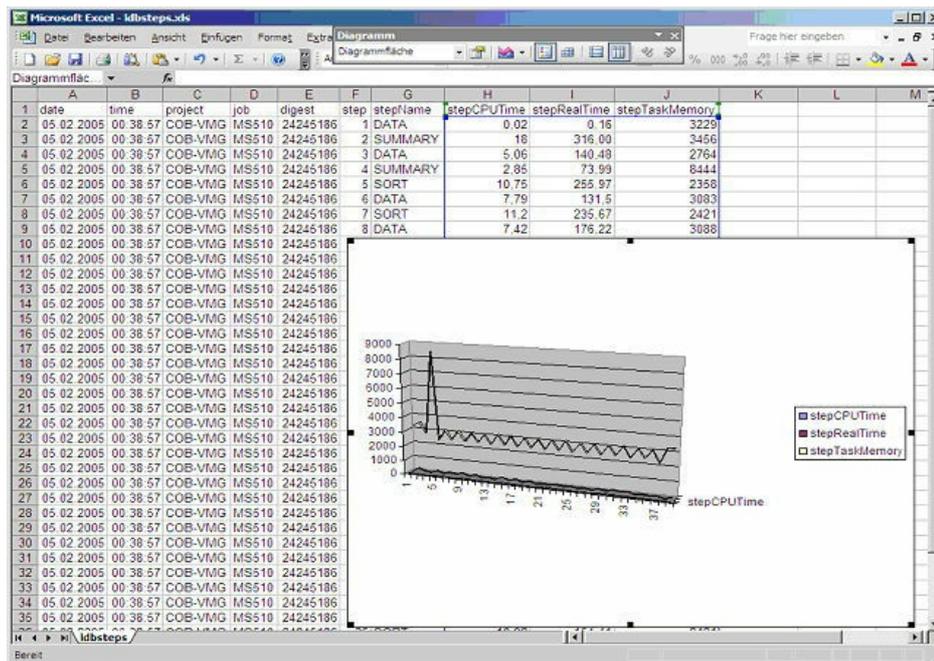
YalanIntro.pdf

121.64 KB

Yalan enables you to build a database containing information extracted from the logs of your sas jobs.

The informations collected by Yalan include:

- Informations about the jobs itself including date and time of execution,
- Informations about job steps (name of procedure / data step),
- Performance statistics on a step level (CPU Time, real time, memory, ...),
- Informations about data sets, database tables and external files read and written including the number of observations and variables,
- Error and warning messages.



Data from Yalan imported into Excel

## Getting Started

You do not have to change anything in your existing sas programs. The only thing you have to make sure is that your sas jobs write the logs to a file by setting the -ALTLOG option.

For example:

- Instead of  

```
sas -sysin c:/saspgm/myppgm.sas
```
- use

```
sas -sysin c:/saspgm/mypgm.sas -altlog c:/saslogs/mypgm.log
```

Yalan itself runs as a separate sas job that executes the macro "rdsStartYalan". "rdsStartYalan" contains all the configuration you have to provide:

```
%macro rdsStartYalan;

  /* Reset configuration;
  %rdsYalan (resetLogFiles);

  /* Where to look for new log files;
  %rdsYalan (addLogFiles, c:\saslogs, *.log);
  %rdsYalan (addLogFiles, c:\saslogs, *.saslog);

  /* In which time intervall to look for new log files;
  %rdsYalan (setPollingInterval, 15);

  /* Set the library (e.g. "c:/MyLDB") to write the log database to and (optionally)
     name the share server to access the log database (e.g. "sassrvt1");
  %rdsYalan (setLDBLibrary, c:\sasldb, sassrvt1);

  /* Yalan will execute the statements contained in the following file every time
     it polls for new log files. You may stop the Yalan Daemon by passing the
     following statement via this file: %rdsYalan (stopDaemon);
  %rdsYalan (setCommandFile, c:\sasldb\yalanCmds.sas);

  /* Start the Yalan daemon;
  %rdsYalan (startDaemon);

%mend;

options nomprint nomlogic nosource notes;
%rdsStartYalan;
```

As soon as a log file is written by one of your sas jobs to a file named in the "addLogFiles" call it will be read into the log database.

### Reading single Log Files

If you want to use Yalan to read single log files into the log database, use the macro "rdsLogReaderSAS".

```
libname lib1 'c:\MyLDB';

%rdsLogReaderSAS
  ( readLog
    , logFileName=c:\saslogs\saslog24665.log
    , logData=rdsLog
    , stepData=rdsStep
    , ioData=rdsIO
    , errorData=rdsError
    , dbLibName=lib1
  );
```

### Yalan on the MVS Platform

Yalan has been tested to run on MVS.

Here is an example of how "rdsStartYalan" might look like on MVS:

```
%macro rdsStartYalan;

    /* Reset configuration;
    %rdsYalan (resetLogFiles);

    /* Where to look for new log files;
    %rdsYalan (addLogFiles, JOB.DWHHU, *);

    /* In which time intervall to look for new log files;
    %rdsYalan (setPollingInterval, 15);

    /* Set the library (e.g. "c:/MyLDB") to write the log database to and (optionally)
       name the share server to access the log database (e.g. "sassrvt1");
    %rdsYalan (setLDBLibrary, dwh.sas.ldb);

    /* Yalan will execute the statements contained in the following file every time
       it polls for new log files. You may stop the Yalan Daemon by passing the
       following statement via this file: %rdsYalan (stopDaemon);
    %rdsYalan (setCommandFile, dwh.yalan.pgm(cmdYalan));

    /* Start the Yalan daemon;
    %rdsYalan (startDaemon);

%mend;

options nomprint nomlogic nosource notes;
%rdsStartYalan;
```

There are two aspects concerning code formatting and naming conventions you have to be aware of when installing Yalan on MVS:

- The source code of Yalan is not formatted to a maximum line width of 72. It uses the maximum possible length of 256.
- Yalan macro names are longer than 8 characters.

The best way to deploy Yalan on MVS is to allocate a PO library with LRECL 256, and proceed as follows:

- Copy all macros except rdsStartYalan in one member. (Let's call it "allYalan".)
- Copy "rdsStartYalan" in a separate member "rdsYalan". (This is the only member you will have to change.)
- Create a third member "cmdYalan" to pass commands to the running Yalan daemon.

When running the Yalan daemon on MVS you need to make sure that the filename SYSIN is available for IDCAMS which is called by Yalan. You therefore have to pass the code to be executed by SAS in another filename than SYSIN. Use a JCL like the following:

```
//DWHYALAN JOB (DWH,KKS,S777),MSGCLASS=X,MSGLEVEL=(1,1),CLASS=D,
//          TIME=(100,0),NOTIFY=&SYSUID
/*JOBPARM S=ANY
//YALAN EXEC SAS,OPTIONS='SYSIN=SASINPUT'
//SASINPUT DD *

OPTIONS S=256 ERRORABEND;

//          DD DISP=SHR,DSN=DWH.YALAN.PGM(ALLYALAN)
//          DD DISP=SHR,DSN=DWH.YALAN.PGM(RDSYALAN)
//          DD *
```

```
%rdsStartYalan;
```

```
/*
```

## Contact

For comments or questions on Yalan please contact:

Kybeidos GmbH  
Stephan Frenzel  
Moltkestrasse 27  
69120 Heidelberg  
Germany  
stephan.frenzel@kybeidos.de

---

- [Screenshot](#)
  - [Database Structure](#)
  - [Download](#)
  - [FAQ](#)
- 

[Screenshot >](#)

---

[Log in](#) or [register](#) to post comments

## Kommentare

---

### Support of LOGPARSE

27 October, 2005 - 15:42 — StephanFrenzel

It would be a good idea to optionally support integration of the [LOGPARSE](#) macro.

[Log in](#) or [register](#) to post comments

---

---