

Paper 172-2010

How to Transform a Technical Exercise into a Real Opportunity Taking Advantage of New Features That SAS® Developed: The True Challenge of a Migration

Grégory Becquembois, Pascal Maurice, GIE Decision Network, France

ABSTRACT

GMF (Garantie Mutuelle des Fonctionnaires) has expanded the use of its business intelligence system for all employees of its branch network through a commercial reporting portal (including SAS® Information Delivery Portal and SAS® Web Report Studio).

The new SAS® BI architecture uses the data generated under z/OS to deal with SAS® under AIX and present the data through SAS Web applications. The portability of the SAS system has saved expensive mainframe resources as well as increased the number of users. This helps to maximize the ROI of GMF's SAS BI Platform.

GMF has been a SAS customer for 30 years for metrology and 10 years for business. Until now, the architecture of GMF's SAS implementation was based on the use of the zOS platform to serve its numerous users. We migrated this configuration to AIX servers, so we are now ready to address the transition to SAS 9.2.

The true challenge of migration is transforming a technical exercise into a real development opportunity, while taking advantage of new features.

INTRODUCTION

We wanted to tell you this migration project because we thought it might interest many SAS customers

Companies that have a project of the same nature, on our side this is the second zOS to Aix migration project for two years and we know that there are two others starting in 2010, therefore a topical issue. In the field of insurance companies using the SAS system for many years, they often have a mainframe culture and the SAS system is often installed and used under zOS. Today, they are more and more thinking to migrate their SAS environment toward distributed platforms, and doing to dedicate their mainframe engines to their business applications.

But also companies planning to set up a decision system on decentralized platforms with operational data from a mainframe. To migrate, we had to start putting in place the foundations required for the proper functioning of the target platform, this step is not specific of a migration project.

We will try throughout this presentation to give you the best possible evidence in the hope they can save you time and help you in your work.

THE COMPANY

Mutual insurance since 1934, founded by civil servants for officials, who since 2005 is the parent company of the Group Covea with MAAF - MMA and also

- Mutuelle Générale
- Mutuelle nationale territoriale
- Groupement militaire de prévoyance des armées
- Mutuelle Intégrance
- Mutuelle générale de la police
- Mutuelle générale des hospitaliers
- Crédit Coopératif

For the GMF, mutualism is a state of mind based on solidarity, which is available both in its organization, contracts and relationships with its members.

Largest insurer of public officials (one in three employees).

Turnover for the group in 2008: 2 956 million, including 1831 non-life insurance, life insurance and 1125.

Result: 125 million euros

Equity: 2 670 million euros

Amount of assets managed: 19 350 million

Solvency Ratio: 3.03

- The 2nd French insurer by number of homes insured (2.4 million)
- And the 3rd in number of vehicles insured (2.7 million)
- 3 129 700 members and customers with at least 1 contract
- 667 642 subscribers Life
- 927 438 subscribers Legal Protection
- 6 014 employees
- 390 agencies

And fan of rugby!

THE SAS BI ENVIRONMENT

The SAS system was used for metrology with MXG from 1985, and then used for the implementation of a decision system, all installed on MVS.

Batch jobs cobol produce flat files, then batch SAS produce SAS datasets available to users via SAS / DMS interactively under TSO or in batch mode.

Development team has developed interactive applications in SAS / AF under TSO, with catalogs and SAS tables shared with SAS / Share

Some users have installed SAS on their Windows machines, in client-server mode with SAS / Connect.

In 2006, installation of the platform SAS BI under Aix to distribute dashboards within agencies with WRS, which has also make available through nfs the SAS datasets produced under zOS to the first SAS Enterprise Guide users.

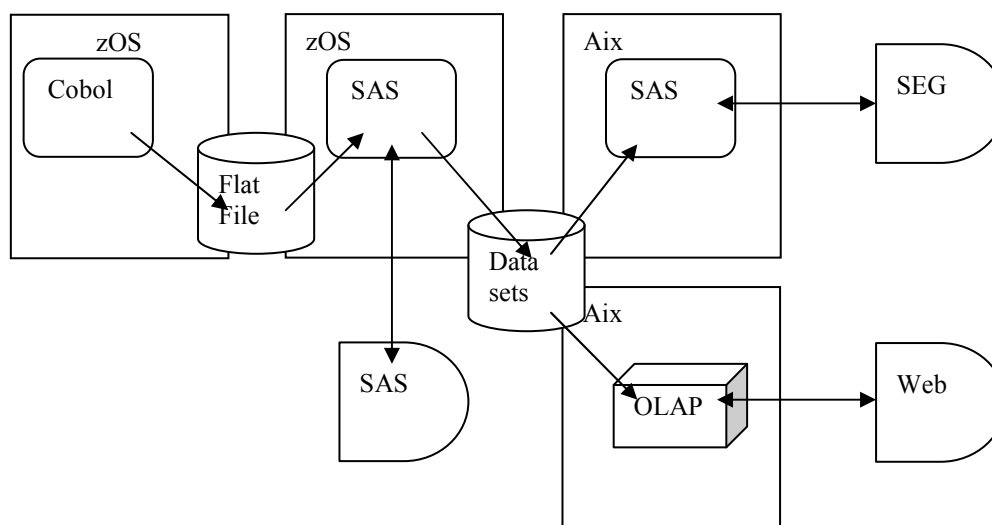


Figure 1. Architecture diagram

Some figures:

- Jobs in 1500 producing 800 tables
- 2 Terabytes of SAS tables on disk
- 350 users coders
- More than 3000 WRS & Portal Users

THE PROJECT

The deal was to migrate all the decision-making system on the AIX platform in order to recover the MIPS partition SAS under zOS. This will absorb the increase in resource consumption by enterprise applications without expensive upgrade of the mainframe platform.

The end of the SAS license under zOS is planned for December 2010. The project actually began in June 2009. It began with a study phase to make an inventory of the configuration: JCLS, SAS programs, SAS tables loading jobs. It led to a division into five steps:

1. The production step: starts to build the required environment for the receipt of the decision system once migrated.
2. The infocenter step: migration of Etl jobs
3. The business intelligence applications step: migration of the SAS / AF applications under TSO to .net plug-ins integrated within SAS Enterprise Guide.
4. The users step: Migration of users programs, SAS programs under MVS Batch or TSO, SAS programs in client-server mode and SAS Enterprise Guide projects.
5. The metrology step: migration of SMF collections made through SAS / ITRM.

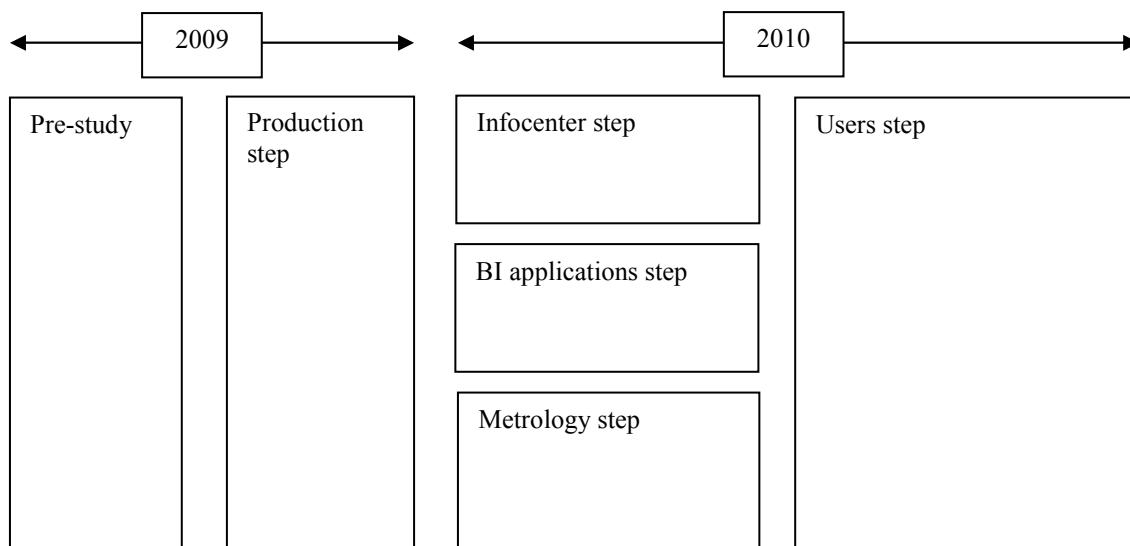


Figure 2. project macro planning

A project team was established with a project manager to coordinate the whole, and on guy responsible for each step.

THE PRODUCTION STEP

The production step must meet the expected needs of the following steps: infocenter, BI applications, metrology and users.

That means implement the essential components to a configuration based on the integration of mainframe and distributed systems. For each subject, we studied the adequacy of the solutions actually used to the new configuration, and decide what will be the best answer, taking into account the constraints of the project:

1. Hold the dead line of the project
2. Providing the same quality of service

What are the questions that we had to address:

HOW TO TRANSPORT DATA

Cobol jobs produce flat files under zOS, these files must be read by SAS programs in Aix. The elapse time of execution of the entire chain must not be longer than the time currently found if all runs under zOS. The different means of transport available are:

1. Sharing disk nfs
2. Tool transport pelican
3. IP protocols: HTTP, FTP.

For ftp, you have an alternative: add a transport step or use the engine ftp within the SAS filename statement.

The latter solution will be chosen for its simplicity of implementation and performance.

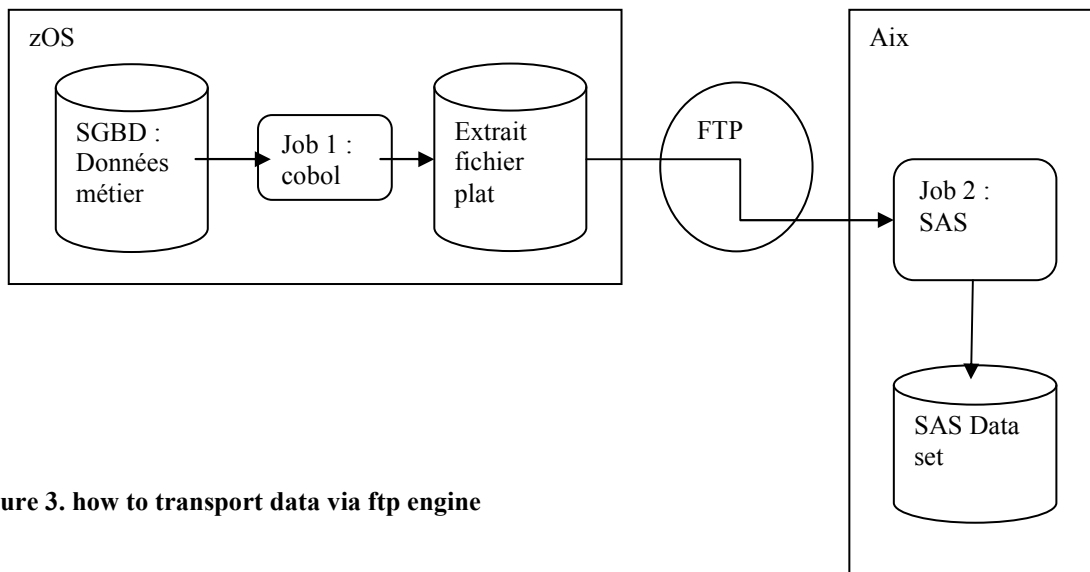


Figure 3. how to transport data via ftp engine

Original under zOS :

```

JCL => DD MONFIC DSN= ENV.DCLF.LIBREF(0)
SAS => Data work.matable_bis ;
SAS =>   Infile monfic ;
SAS =>   Input  var_char  $ebcdic10.
SAS =>           var_num_1  &num.3.
SAS =>           var_num_2  &PD.5.
SAS =>;
SAS => Run ;

```

Target under Aix :

```

STDIO=> Filename monfic ftp mach=hos_zOS 'ENV.DCLF.LIBREF(0)'
        lrecl=111 recfm=F ;
SAS => Data work.matable_bis ;
SAS =>   Infile monfic ;
SAS =>   Input var_char  $ebcdic10.
SAS =>           var_num_1  &num.3.
SAS =>           var_num_2  &pd.5.
SAS =>   ;
SAS => Run ;

```

Note:

The macro variables &num and &pd contains the informats used to read data in EBCDIC format ,display or packed decimal, these variables are set in the autoexec using the call to a macro initialization program that tests the OS under which runs the SAS program.

STDIO option allows to generate instructions and filename or libname by the script that starts the execution of the SAS program, and thus preserve the independence of naming files and directories that is the responsibility of production teams, SAS code is the responsibility of development teams.

The mach option in the filename statement can use the login and password stored in the .netrc file stored in \$HOME directory of the user submitting the job, and therefore not having to store this information in scripts or programs to address issues of security and centralized maintenance.

HOW TO APPLY THE STANDARDS

How to implement the standards that were developed for production under zOS and Windows.

Standards for naming jobs, files and directories environment (development, qualification, certification and production). The jobs are defined in Cortex, mainframe tools, and therefore constrained by length of 8 characters to name the objects.

The jobs are grouped by application, an application is the set of ETL jobs that produce SAS datasets associated with a BI application, or a functional branch owner.

Within an application, the jobs are grouped by jobset, a jobset contains the same type of jobs, supply jobs, jobs for archiving, and planning, for example, daily jobs, jobs weekly etc..

From jobset the names of DCLF (ID file or directory) are built, with which we will be able to produce cross-references for impact analysis, or problem solving.

On the Aix machines, the directory tree to receive scripts and programs has been implemented on the existing model for Windows platforms:

<i>Path</i>	<i>Content</i>
/ exploit / sas / <env> / scripts	Perl Scripts
/ exploit / sas / <env> / procs	SAS Programs
/ exploit / sas / <env> / macro	SAS Macro-programs

/ exploit / sas / <env> / parametres_histo	Historic parameters
/ exploit / logs / sas / <env> /	Logs

HOW TO SCHEDULE JOBS

We have to managed in the same chain, jobs executed under zOS COBOL and SAS jobs that run under AIX. Cobol jobs are planned by CA7. The alternative could also be to use Autosys on decentralized platforms (Windows and AIX). The choice was CA7, mainly because of the facility to integrate the zOS and Aix platforms. A master manages the CA7 job scheduling under zOS and AIX via a client CA7.

The jobs are first loaded into cortex in the production repository, they are called PCL, which will generate JCL, with DCLF declarations of files or directories. Afterward Cortex can be used to request cross-reference files * jobs, and therefore know the impact of changes to files and / or programs.

From Cortex, jobs are compiled for an environment (Qualification, Certification, Production), the variables used to construct the names of jobs and paths of files are then used to match each environment. For example, a job qualification will be prefixed JQ in qualification, HH in homologation, P1 in production; or the SAS datasets are stored in a tree starting with /qr1 in qualification / hr1 in homologation and /pr1 production.

In contrast, the chain structure should be adapted, jobs submitted in Aix are necessarily single-step, because the job must wait until after the return code from the previous execution of the job by the client CA7 in Aix.

Finally, we must implement the transition characteristics of files and / or directories to the script that will run the job under SAS Aix. We have structured these parameters the following scheme:

1. Number of input files
2. File by file characteristics: fileref, hostname, dsName, LRECL, RECFM
3. Number of output file
4. File by file characteristics: fileref, hostname, dsName, LRECL, RECFM
5. Number of libraries SAS input
6. Bookshop bookshop features: libref, dsName
7. Many bookstores Release
8. Bookshop bookshop features: libref, dsName
9. Name of the SAS program to run

For example, a program that will read a file to write to a SAS library:

1. "1"
2. "Myfile" "myhost" "PR1.DCLF.MONFIC" "\ (0\)" "288" "F"
3. "0"
4. "0"
5. "1"
6. "Mylib" "/pr1/appli/dclf/libref"
7. "My_prog.sas"

HOW TO BUILD HISTORICAL DATASETS

1. Job A ETL build the dataset in the current period
2. Job B fist level archiving copies the current table in the first level of the historic appending a time stamp depending on the frequency
3. Job C rename removes the suffix _bis from the dataset in the current period
4. Job X historization second level copy the current table in the second level of the historic appending a time stamp depending on the frequency

For example:

1. Job A creates the dataset my_data_set_bis
2. Job B copy my_data_set_bis in my_data_set_year_week
3. Job C rename my_data_set_bis in my_data_set
4. Job X copy my_data_set in my_data_set_year_month

The data is then copied at least once depending on how many levels of archiving.

Now:

1. Job A build the dataset in the current period in the first level history with a time stamp suffix depending on the period.
2. Job B archiving creates a link to the current level
3. Job X historization 2nd level mirroring the current table in the second level of the historic appending a time stamp depending on the frequency.

For example:

1. Job A created the table my_data_set_year_week
2. Job B created a link from my_data_set to my_data_set_year_week
3. Job C copy Job my_data_set in my_data_set_year_month

This way you save a job and a copy of the data set.

The script manages the archiving of historical depth which is supplied via a parameter file whose name matches the job that uses it and which contains for each table to historicize:

The current path of the table to copy, the table name to copy, the number of periods to keep, the periodicity of the history and finally the path of history in which we must copy the table, for example, to keep a history of 10 years:

```
/appli/DCLF/libref DATASET_NAME 10 A /appli/DCLF/libref.hist.a
```

The available frequencies are:

- Daily
- Weekly
- Fortnight
- Monthly
- Quarter
- half
- Annual

HOW TO DELIVER SCRIPTS AND SAS PROGRAMS

For web applications, svn and scm4all are used. To deliver the scripts, SAS programs and parameter files, we used only the scm4all tool. This tool generates records associated with an automatic version number in a first stage called build, afterwards delivery is only dropping the source files in the good directories.

The implementation of this tool called relatively little work (some days) and ownership of the interface, although not very friendly was made fairly quickly.

HOW TO SAVE AND ARCHIVE

The current system, SAS tables are stored on disk, but if they are not accessed by users, HSM will move on media cartridge. If a user requests access again to their data, HSM restore them without user intervention. Beyond the historical data stored on disk, SAS tables are stored on a cartridge through a job and proc copy. These archives are

accessible to the user only in batch mode. For data contract motor vehicles for example, data on disk for five years, then archived on cartridge for 10 years.

In the distributed world, the product used for backups and archiving is Netbackup. There is no equivalent of HSM.

For data produced by the ETL jobs, the datasets suffixed `_bis` enables secure treatment without having to use the system backup / restore.

For PDBs (Performance Data Bases) metrology, it is important to implement the ability to restore all files from a pdb in case of abend the job of collecting / archiving.

We therefore used Netbackup to save the PDBs during the day, making sure not impacting other processes active on partitions Aix.

Two modifications have been implemented:

1. Setting up a direct network connection between machines running Unix or NetBackup and SAN or files are stored in order to guarantee bandwidth to use real time.
2. Assign two drives and a pool cartridge application to ensure the recovery time in case of abended jobs.

In the longer term, the synchronization between treatments backup / restore and physical media will be via virtual tapes, which will optimize the use of the library by removing the dedicated readers. This solution was not considered immediately for budget reasons.

Finally, for services requiring high availability, such as metadata repositories, the solution adopted is the snapshot Aix, to quickly restore service and then the SAS application availability in case of abend of jobs that update metadata.

For archiving, to be SAS version independent we decided to store the data via the XML engine.

HOW TO SUBMIT USER'S SAS PROGRAM IN BATCH

The problem to be solved is that we do not have JES2 under Aix, and therefore we had to find a solution to offer equivalent functionality.

For the jobs that do not involve the mainframe, distributed generation uses the product Autosys (CA also), natural solution developed to enable users to submit batch SAS programs was to use this tool.

The submission of programs in batch mode is done via an interface developed in .net and available as a SAS Enterprise Guide plugin. The interface is a GUI used to send to the SAS session running on the AIX platform execution scripts parameter. These scripts provide the following functions:

1. Submit a batch SAS a parameter with the name of a config file, an autoexec file and name of the SAS program to run. The print and logs are stored in a directory of publication to be retrieved via HTTP protocol.
2. Application of state jobs from a user: give a list of jobs submitted with the state (pending, running, running, completed successfully, abended).
3. Cancelation of a submitted job: You can change your mind and cancel a job.
4. Request to stop a job running: can kill the process associated with a job running.

Beyond the fact that we have to offer at least equivalent to existing services, we may assume that using the batch should be less important. It was completely justified under zOS with a single point of entry under TSO, so if the user wanted to parallelize two jobs, one of these treatments has to be in batch mode, the user now has the ability to run multiple SAS Enterprise Guide sessions and so we can assume it will be less need for this feature.

HOW TO MEASURE THE QUALITY OF SERVICE

In the old configuration, it had no dashboard to monitor systematically the quality of service and the service definition was more a matter of oral tradition than a result of a formalized process. But to exchange the additional work generated asked to the users in this migration project, it seemed important to take this opportunity to introduce monitoring of quality of the service provided.

He had to start by defining a minimum:

1. Service associated with the ETL jobs: the datasets when they are made available at the beginning of each period.
2. Availability, and response time of the interactive sessions.

Then by re-shaping the Tivoli platform used by the production monitoring.

Then establish scenarios during the user step, with connection scenarios and submit a SAS program to recover the performance synonymous of the availability and response time.

Then construct the dashboards that include dates when SAS datasets are available, system availability and response time.

THE METROLOGY STEP

This step was the easiest of all, because the context was less complex, fewer jobs, fewer tables, fewer users, and the same questions already found within other metrology migration project that we had done at d other customers.

However it allowed us to prototype and validate the various options that had been decided to address the needs of the infocentre and users migration.

There was no need to convert the SAS programs that are the source of the solution MXG SAS / ITRM. We have to establish processes for recovering historical data stored in the PDBs ITRM and transport them in Aix.

The inventory shows us:

- The division into classical field: Jobs, RMF, CICS and IMS, DB2, RACF, MQSeries, etc..
- And a cutting by partition: partitions zOS production, certification, etc..

The intersection of the two divisions gives forty PDBs.

Given the performance observed in Aix (time of collection / reduction divided by five) we may be considered after the migration to reorganize these divisions, by keeping such as cutting by area and company to reduce the number of PDBs and therefore facilitate the construction of data marts dedicated to analysis and dashboards.

THE MIGRATION STEP

As you can see, this chapter is the smallest of this document, which reveals the true nature of this type of project when you migrate a SAS environment with a platform to another platform, the difficulty is not so much in the transformation of SAS programs, but the transition from a production environment to another.

What questions do we have to address:

1. Transport sources zOS to Aix
2. Transformation of SAS programs to adapt the instructions input, libname, filename, etc..
3. The \$ EBCDIC informat is equivalent to a \$ char, it was therefore necessary to manage sometimes removing characters left via the left function
4. Manage the difference of minimum length fields under zOS length numeric fields was specified, we have increased to their maximum (8)
5. Taking into account the different sort sequences and ASCII to EBCDIC

What are the stages of this migration:

1. Inventory
2. Selection of source to migrate
3. Getting source
4. Transformation
5. delivery

Validation is an element important in sizing the project in terms of time spent, we can eg consider converting hundreds of programs a day, but if we must validate this transformation through creation of a set of test and double and run comparison listing, you can validate only a dozen a day. The experience of this type of project shows that the transformation gives more reliable results than those generally expected, and consequently it is often possible to

save time validation.

Regarding the loading programs, the result produced is a SAS dataset, we have only to compare the datasets produced by the double run to validate the processing and hence would take only minimal time for validation.

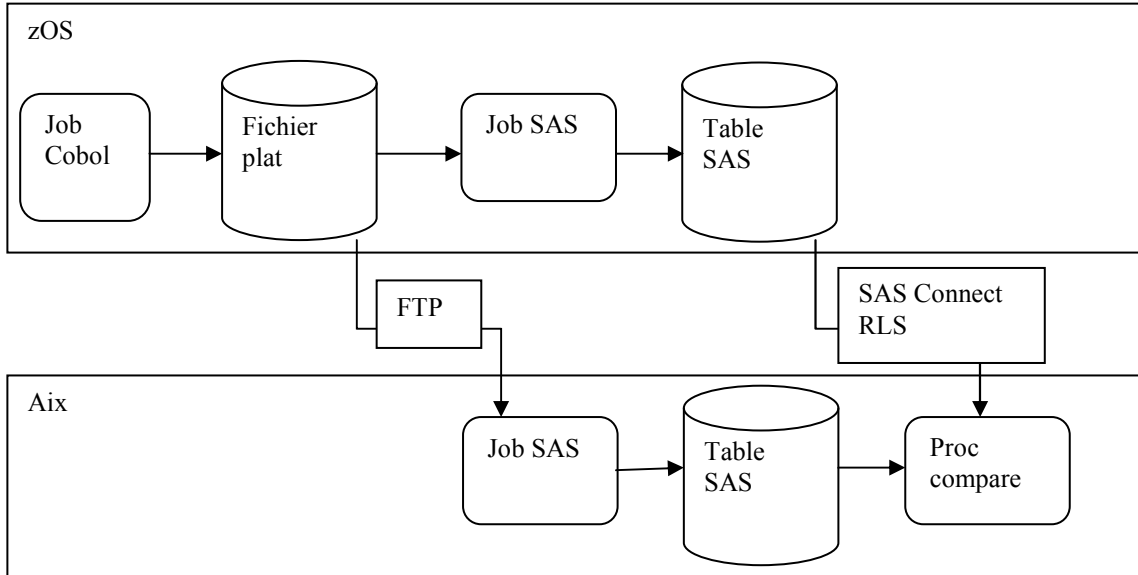


Figure 4. Validation schema

The proc compare options take into account the difference in management numeric precision of real variables between zOS and Aix, and also the difference of the lengths of numeric fields that has been increased.

The transportation was done via FTP, specifying the code page of source and target systems to maintain the national character education via ftp:

```
quote site sbd = (ibm1147, opened8859-15)
```

At least we have to replace the characters 'à' at the end of the instruction input by the character '@'.

THE USERS STEP

Of all the steps of the migration, the users step outlines was the less clear. Indeed, by definition, users have a set of tools but all solutions implemented by them to meet their business issues are never fully identified by the IT.

Thus, at the beginning of the project, regarding figures, we had to migrate a 15 years background of SAS programs for 350 users, that is to say 22 000 programs for some 3 million lines of code! By themselves, those figures can be a real budget problem...

Fortunately, thank's to our previous projects, we knew that most of those programs would not be migrated and we know the major risks of this type of project:

The Risk # 1: A systematic validation

Indeed, the main risk, in terms of costs, is related to the systematic validation (by users and IT) of all migrated programs. Trying to approved too many programs can quickly lead to a heavy workload (22 000 programs!) Without real benefits. With the additional difficulty that IT can't submit alone users's SAS programs (risk of update or delete important SAS Data sets for example....)

Early in the project, users still want to approve the most of their programs, sometime just for not losing time to scan all of them to know if they are still usefull. The reason to this is simple to understand: at this stage, migration gives them no more than an additional workload.

To overcome this initial difficulty it is important to identify as soon as possible the capital gains (benefits) that could be given to users and make it clear to them those new opportunities (taking account of unmet needs, optimization, automation, simplification of use ...). The addition of capital gains is essential to motivate users to enter in this type of project. Another unifying factor can be to include users in little POCs and benchmarks on the target environment.

Once users integrated to the migration project and motivated by capital gains it's easier to obtain from of them a work of synthesis on their huge initial capital of SAS programs. This synthesis work will limit migration and validation work and will allow to all (users and IT) to focus on implementations of new features.

Risk # 2: users's availability

Another key point to integrate in this type of project is the fact that, unlike IT poeples (private joke, i am one of them !), users have a real job. Seriously, they can't work in full time on a migration project and if you try to force them, all your efforts to find capital gains and to integrate them in the project could be quickly destroy. And a users migration project without users's participation is clearly an impossible mission (keep cool, this document will not be destruted in 10 sec.. ;-)).

That means that you have to schedule your time according to users's availability. Sometimes, it is quite difficult to wait but...

For example, for all GMF's SAS users, 6 months were necessary to achieve the writing of papers (list of all usefull sas programs). Less than 1000 programs were checked by users on the initial capital of 22 000 programs. Yes, Six months is a long period, but actually nothing regarding to years needed to validate all initial programs. Of course, during this elapsed period all of the gains have also been identified, tested and validated by users. In fact, this 6 months period is the necessary time to federate all users in the project... and all these before to work really on the migration...

Summary of the working method:

- Create a working groups (by type of job for example)
- Organize an information meeting
- Identify key users (interlocutors)
- Work hard with key users to have a good understanding of SAS usages in each jobs (Interaction with other tools (eg Microsoft Excel),Needs in terms of resources (CPU, Disk ...),
- Identification of handling non-productive and tedious
- Identification of capital gains and extra SAS programs features to implement in the new environment (files transfert, SAS FSP usages,...)
- Test and validate capital gains with users and IT
- Obtain from users papers with usefull SAS programs to migrate
- Organize training on new tools (eg SAS Enterprise Guide, Add-in for MS Office)
- Migrate programs
- Validate, with users, transformations in the new environment
- Switch-users in the target environment
- Organize support to help users in this new environment

Important: one more time, during all the steps it is essential to inform users about the progress, problems encountered, once again this part of the migration must be a common project and not only an IT project.

RESSOURCES USED

	Project Tracking	Expertise	Development
Pre-study	45	75	74
Production step		27	9
Infocentre	15	80	68
Metrology	15	75	52
Users	35	120	100
Total D / h	110	376	303

GMF in turn has two people dedicated to the production during the project period, and we have no value for the rewriting of intelligence applications.

It should add the licensing of agents used on platforms SAS (scm4all, CA7, Autosys, Tivoli)

CONCLUSION: READY TO GO

This migration project which should be a technical exercise has finally resulted in:

- The consolidation of production in Aix.
- Better service
- Monitoring the quality of service
- The renovation of environmental decision
- A better experience for users

It is now ready to be migrated to 9.2, see you next year?

CONTACT INFORMATION

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

Name: Pascal Maurice
Enterprise: Decision Network
E-mail: p.maurice@decision-network.eu
Web: www.decision-network.eu

Name: Grégory Becquembois
Enterprise: Decision Network
E-mail: g.becquembois@decision-network.eu
Web: www.decision-network.eu

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.