

## **Setting Up Winning Conditions for a Migration of SAS® from a PC Environment to a Server Environment**

Mathieu Gaouette, Videotron S.E.N.C.

### **ABSTRACT**

Whether it is a question of security or a question of centralizing the SAS® installation to a server, the need to phase out SAS® in the PC environment has never been so common. On the surface, this type of migration seems very simple and smooth. However, migrating SAS® from a PC environment to a SAS® server environment (SAS® Enterprise Guide®) is really easy to underestimate. This paper presents a way to set up the winning conditions to achieve this goal without falling into the common traps. Based on a successful conversion with a past employer, I have identified a high-level roadmap with specific objectives that will guide people in this important task.

### **INTRODUCTION**

The server infrastructure centralizes and optimizes the use of resources with regards to SAS® and offer a wide range of new benefits to the end user while allowing some tighter security. At first glance, it sure looks like a move where everyone wins. To a certain extent, this is true. However, there are some pitfalls that can be disastrous to a migration project unless taken into consideration. Knowing about these hurdles in advance is important in order assess and mitigate them.

This paper's objective is to identify these major obstacles and give some insights on how to tackle them. Project managers working on such projects will obviously gain from this information. End users could also benefit by using that knowledge when planning tests for their own migration projects to ensure a smooth transition.

### **CREATING A SOLID BASE**

Underestimating the task in a project can result in major setbacks. In my recent experience, this type of migration project was somewhat underestimated. I am convinced that it was no exception to the general rule. This type of assignment simply is prone to this.

It is therefore crucial to have a strong base for the project. In this case, it starts with the root: the objective. Migrating to a server infrastructure can be initiated for a number of valid reasons such as a desire to reduce costs, optimize performance or enhance the security. Make these motivations be known to all participants of the project. Doing so will prove key when the hurdles are encountered and some people start challenging the project itself.

Communication is more than an asset, it is a requirement. More specifically change management must be considered. As you will see in the next section, migrating operations to a server is no trivial task. As a result, you will impact the habits of your end users. Good communication channels must be established and used to keep in touch with our SAS® users. Newsletters, touchpoint calls, workshops; these are all good ways of achieving that. The goal is to get good updates and overall feedback from your users.

In addition, you will have to decide whether or not to keep a few desktop licenses as a fallback plan in case you would have to revert in some cases to the desktop. This is a safe strategy. It is important to keep in mind that these licenses are an exception and users having them should be aware of it. Moreover, no new development should be performed while using local installs. Keeping about 10% of the licenses at the end of the first year is a good rule of thumb. Of course the number of different groups using SAS® and the number of power users in total can affect this proportion.

### **GETTING KNEE-DEEP IN**

Whether you will be personally involved in the actual conversion of code or not, you should be aware of the problems that will be encountered, taking on this task. These are laid out in order that they could very well be encountered but it is no guarantee that it will play out like this.

## CHANGING THE OS

Migrating from an OS running on a desktop to a different OS running on a server is going to produce lots of surprises. Of course there is the operating system itself (ex: Windows, Linux, UNIX ...) but we also must keep into consideration whether it is a 32bit OS or a 64bit OS. Passing from 32bit to 64bit one while keeping the same OS still provide challenges.

### Moving to a UNIX environment

Moving to a UNIX environment (or similar such as IBM's AIX or Linux) is a common practice. It often allows easier server management and better security. This move is heavy in consequences for the end user. One of the main impacts of this is the lack of a Microsoft Office capacity on the server, such as Microsoft Office or Microsoft Access Connectivity Engine (ACE). This means that barely no Excel or Access files can be created or read from the windows server.

The smoother alternative to allow these interactions with Excel and Access files is to deploy a small windows server that runs SAS PC files server®. This server does not need to have base SAS® installed nor does it need high processing power. This infrastructure will allow SAS® sessions under any OS to communicate with the PC files server engine to act as a translator to windows files.

What follows is an example of the syntax used to export while using a distant PC files server machine:

```
PROC EXPORT DBMS=EXCELCS DATA=sashelp.cars
  OUTFILE='\\networkdrive\networkfolder\cars.xlsx' REPLACE;
  PORT=9621 ;
  SERVER='yourSASPCFS';
  SHEET='data';
RUN;
```

With SAS® 9.4 (M1), a new experimental feature has been introduced to produce native Excel files through ODS: ODS EXCEL. This feature can replace some export procedures to Excel. Being experimental, I suggest that you do not take for granted that the conversion will be covered by this technique.

Another great alternative is the use of the ODS tagset ExcelXP. This tagset creates actual XML files. However, you can create the file with an 'xls' extension so that it will be opened within Excel. There are plenty of examples of such reports online. Mind you, ODS usage might imply some rework if the original solution was not already using ODS reporting. Also, this technique only addresses the output to Excel and not the input as it is not its purpose.

Lastly, one viable solution is to leverage the SAS Add-in to Microsoft Office®. To put it simply, the add-in allows users to run encapsulated SAS® codes known as stored procedures and get the result displayed in Excel (or other Office documents). It also allows users to read or import data from SAS® libraries. This implies a paradigm shift as instead of having SAS® users producing and distributing reports, you have the report consumers actually refreshing their reports themselves. Such a change has to be well directed as it could imply a larger number of people actually using the SAS® infrastructure. It is a practical option for a limited number of report consumers. For large groups, another solution more easily scalable should be considered.

### Passing to 64bit

Most of the conversion problems will come from formats of SAS® artifacts such as datasets and catalogs. These objects created under 32bit will have to be converted/migrated to the 64bit world. For these, plenty of resources can be found online to help.

When it comes to interacting with Office formats such as Excel, the solutions are not as simple as an artifact conversion. Running on a desktop, the end user has access to a variety of DBMS engine related to Excel, such as XLS and EXCEL. The limitation with a 64bit system is the "EXCEL" DBMS. This engine is not supported under a 64bit windows system. A work around is to use the DBMS XLSX or EXCELCS (used by PC Files Server). Although the XLS engine also works, it is better to avoid it as it is

an older format. For Access, there is a similar limitation with the ACCESS engine. The alternative is ACCESSCS (also a PC Files Server DBMS).

When using the PC Files Server format, it is important to know that behind the scenes, SAS® does not use Microsoft Office's libraries but rather use the Microsoft ACE engine. This engine has some limitation when it comes to deleting or altering files. The limitation comes in when you attempt to delete or replace a sheet that has more than 65536 lines. In this case you get a cryptic error. One of these errors is the following:

```
ERROR: CLI execute error: [Microsoft][ODBC Excel Driver] Cannot expand
named range.
WARNING: File deletion failed for _IMEX_.MYSHEET.DATA.
ERROR: Export unsuccessful. See SAS Log for details.
NOTE: The SAS System stopped processing this step because of errors.
```

**Output 1. Error message found in log while trying to replace data with SAS PC files server®**

One way of avoiding this situation is obviously to always stay below the 65000 lines mark. With a fixed report or list, it might be possible but in most cases, growth will slowly make the number of rows increase over time. In these cases, the use of a report template could avoid the problem. Using a model allows you to have a blank report that you copy over when you produce your updated report. Once copied over, you simply export to the given empty sheet. As the limitation with the ACE engine affects the deletion or alteration of a large sheet, you avoid this by only exporting once every time.

## **DESKTOP VS DISTANT SERVER**

When operating on a desktop, SAS® is king of the hill. Users have access to a lot of files and resources that are not necessarily accessible from the server side. Often, these situations should not be because they pose a security threat but nevertheless, they should be addressed as they are part of existing components to migrate.

### **Local drives**

It is quite common to see users make use of local drives such as Window's "My Documents" as a source or destination for files. In a secure server environment, SAS® does not have access to user's local Windows drives. In most cases, these issues are easy to work around. For shared files, they should be moved to a shared location available from the server.

For private files, ensure that you provide your users with an alternative whether it is a private folder on the server itself or a location available over the network. Mind you these situations are not common. You should only tackle these privacy issues if they really apply to your business.

### **Shell escape**

SAS Display manager® users (also referred to as SAS/PC®) have the possibility of running system commands from within SAS®. These commands are often referred to as "x commands" because of the name of the option that allows their use. With this feature, the user has the opportunity to run any command that could be ran from a windows command prompt (or shell on a UNIX OS).

To someone that does not work with SAS®, it might look hazardous but it is in fact a very helpful tool when it comes to automating SAS® code. People will often use it to copy/rename files and create directories to name only two. There are various ways of using this feature with the three most commons being:

- X statements
- Filename PIPE use
- Systask statements

With a Server infrastructure, you will have to decide whether or not you want to enable this feature. Regardless of your decision, you should definitely consider the current usage of this feature among your users. Taking out this possibility will impair some of them. As a result, some existing solutions might have to be reworked from scratch to avoid the use of x commands. As some of the alternatives might not even involve SAS® programs, you should be really cautious with that avenue to avoid major impacts on the project.

With a secure server, allowing x commands should not be a security risk. If a user can execute dangerous commands from a shell escape in SAS®, it has to be because his userid has these rights. X commands do not provide elevated rights. For that reason, I don't think that x commands should be prohibited. Taking them out does not solve security holes, it simply helps hiding them. On the other hand, I would really advise developers to limit their use as much as possible.

Note that allowing x commands will not solve all of your problems if users are using desktop specific applications through the command line, like encryption or archiving programs. In such cases, you could simply deploy the same applications on the server. That however is assuming that your server's operating system is the same.

If the OS is different, then you might have to find alternatives. The most common usage of an external software is probably winzip (or the older pkzip). A free alternative is the open source 7zip software. It also allows command line calls. If you have a UNIX server, you could also use the standard gzip and gunzip applications.

## **DDE**

These three letters will haunt you during that project unless you assess the situation right early on. DDE stands for Dynamic Data Exchange. It is a method of interprocess communication that was introduced in 1987 for windows 2.0. It has since been replaced by a lot of newer technologies and Microsoft openly recommend users to avoid DDE when possible.

This method allow users to execute a wide variety of operations within Excel that would not be possible without it. You can fill/clear cells, create/delete sheets, run Excel macros and much more. DDE is a windows method and is not available on a different OS. Moreover, you can only use DDE with an Excel which resides on the server itself. This method can't be used over PC files server.

Because of the possibility of installing Office on a windows server, the biggest challenge is obviously when moving to a UNIX server. It would be nice to have a magic solution that applies to every DDE use but there is no such thing. The only proper way of tackling that hurdle is to take each code and revisit the solution itself. In most simple cases, the use of PC files server can address the use of DDE. In the more advanced cases, a possible rework could involve the use of the SAS add-in for MS Office®.

## **Implicit use**

A desktop install of SAS® is much more than just the SAS Display manager® software, it is also the SAS® engine itself. This installation behaves just like a server in almost all regards. As a result, a SAS/Enterprise guide® user can connect to this server in a similar way as he would connect to a distant server. In the software, the server will be listed as "Local" in the server list. Obviously a code developed to work on a local server will need to be modified in order to work on a server just as if it had been developed with SAS Display manager®.

Some users will also use batch processing in order to run SAS® code initiated by a scheduler or simply to achieve simplicity of execution (ex: A windows batch file that can be double-clicked). As with the previous example, the affected codes will require the same kind of attention as the other SAS/PC® codes.

## **OTHER CONSIDERATIONS**

Another important point is the access to an smtp server to send emails. Usually, the servers will support this feature because of its use for system alerts. If your security team does not want smtp ports opened on servers then you will need to size the amount of work that will be required to develop alternative solutions for all these SAS® codes that send emails.

For that problem or just as an added tool, NAS can be considered in order to allow SAS® to create files/reports that are accessible from a desktop (via a network drive). SFTP servers can also be used. A good way of replacing a report being sent over email is to transfer the report to a destination accessible by the target users and then send an email to inform them of the availability of the given report. In case of a network drive, the direct link can even be provided in the email in order to allow the user to simply click on the link to open the report.

## CONCLUSION

Moving your SAS® processing to a server certainly has its load of challenges as I have shown here. Having a proper grasp of the task at hand truly is an asset here. End users, especially the more technical ones will surprise you with their home made solutions. Odds are you will even encounter situations not listed here. If you have followed my advices and have built a solid foundation, you will get through.

Make sure that you involve the end users. They must be a part of the project rather than being spectators. This is especially important for the end user's managers as they will be your strongest allies.

And finally, keep in mind that in most cases, users have in hands solutions that require very little housekeeping (ex: they change a few parameters and hit "run"). Proposed alternative solutions should be as easy and as simple, for them to get onboard. Use workshops to discuss possible solutions or alternatives. Good change management and agility will be your glue to get all of these winning conditions aligned together.

## REFERENCES

Wikipedia. "Dynamic Data Exchange." Available at [https://en.wikipedia.org/wiki/Dynamic\\_Data\\_Exchange](https://en.wikipedia.org/wiki/Dynamic_Data_Exchange).

Hemedinger, Chris. "The top gotchas when moving to 64-bit SAS for Windows." The SAS dummy blog. May 1, 2012. Available at <http://blogs.sas.com/content/sasdummy/2012/05/01/64-bit-gotchas/>.

SAS institutes. "SAS(R) 9.4 PC Files Server: Installation and Configuration Guide." Available at <http://support.sas.com/documentation/cdl/en/pcfsicg/67728/HTML/default/viewer.htm#titlepage.htm>.

Hemedinger, Chris. "Experimenting with ODS EXCEL to create spreadsheets from SAS." The SAS dummy blog. August 29, 2014. Available at <http://blogs.sas.com/content/sasdummy/2014/08/29/experimenting-with-ods-excel-to-create-spreadsheets-from-sas/>.

## CONTACT INFORMATION

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

Mathieu Gaouette  
Videotron s.e.n.c.  
[mathieu.gaouette@videotron.com](mailto:mathieu.gaouette@videotron.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.