Migration + upgrade = happy users,

but how many days downtime?

Nigel Montgomery, Novartis Pharma AG, Basel, Switzerland
David Garbutt, BIOP, Basel, Switzerland

Abstract

In 2009, Novartis started the 'GPSII upgrade' project which included SAS® migration from v8.2 to v9.2, version controlled system (ClearCase®) upgrade and server relocation and virtualization. This affected approximately 900 users from within Statistics, Programming, Data Management, Database Programming and CRF Development.

This paper discusses how we determined some key differences between SAS® v8.2 and v9.2 and additionally how we managed the scheduling and migration of approx. 600 projects including approx. 3 TB of SAS data.

We will share the SAS® v9.2 differences from v8.2 we found during this project, along with other key findings throughout the project.

The way that can be told is not the true way

Lao Tzu, Tao Te Ching

1. Introduction

The origins of this GPSII upgrade project within Novartis date back to early 2007, when only a SAS® upgrade was being considered. This original project was put on hold due to the fact that management were considering further upgrades on other software and hardware and were considering if it was feasible to implement all these upgrades at the same time. Another reason for the delay was that there was potential for considerable downtime for the everyday user.

In late 2008 / early 2009, Novartis decided to proceed with the now newly named GPS II upgrade project, incorporating a SAS upgrade, a version controlled system upgrade, a server relocation and virtualization along with other upgrades such as an SPLUS® upgrade and a change in project access rights process.

This paper will give an overall description of the GPSII upgrade project and

what it entailed. Particular focus will be on the differences between SAS v9.2 and v8.2, the scheduling and migration of the large volume of data (with minimal downtime), how we created an automated process and the lessons learnt from this project, good and bad.

The end result was a thorough process that enabled 3.5 (IT) FTEs to migrate and convert the 3.5 TB (550 projects) in six and a half months as scheduled.

1.1. Business Coordination

As with most non-clinical projects, the GPSII upgrade project was originally under-resourced both from an IT and Business perspective. The *business* in this context, refers to everyone who works with ClearCase® , SAS®, SPLUS® etc. on Statistics, Programming, Data Management, Database Programming or CRF Development. Original plans had one business coordinator working at approximately 20% of the time on this project in order to coordinate fully with IT and relay information to the business as and when needed. After careful consideration from the original business coordinator and as the project grew in size, it was deemed that we would need a business coordinator for at least 50% of their time. In the end, the business coordination was shared between three people with their total time making up the total required. The Business Coordinator role would use their business experience on all the software that was being updated and give their input into how best to deal with the tasks and issues that arose during the GPSII upgrade project.

Additional business assistance was deemed necessary, due to the fact we had approximately 600 projects to migrate, with some projects having up to as many as 200 studies, coming from many different therapeutic areas. We therefore assigned the 'Migration Champion' role to a suitably qualified and experienced business member on a therapeutic area and Novartis site (East Hanover, Basel, Horsham, Hyderabad, Tokyo and Shanghai) basis, thus resulting in some 20 migration champions being assigned to cover all the business areas, both therapeutic and geographical. These migration champions were incorporated into the larger GPSII upgrade team and were given the necessary information and training they needed in order for them to perform their role.

The business coordinators and migration champions had many roles within the overall project and one of these roles can be seen in Figure 12 on page 20. Further roles and responsibilities will be touched on later in the paper.

1.2. The old and new environments

The migration project was separated in to two parts: new server set-up, upgrade and configuration was executed first and the second part was to populate the new server with studies and convert current dataset versions to SAS V9. Consequently for clarity we will try to use the term *migration* when referring to the re-location of SAS files and programs and the term *conversion* when referring to to the process of changing the format of a SAS (data) file object.

The old and new configurations are diagrammed in Figure 1 and Figure 2.

There were some problems with the old environment and the goal was to solve these issues. To improve performance and reliability we

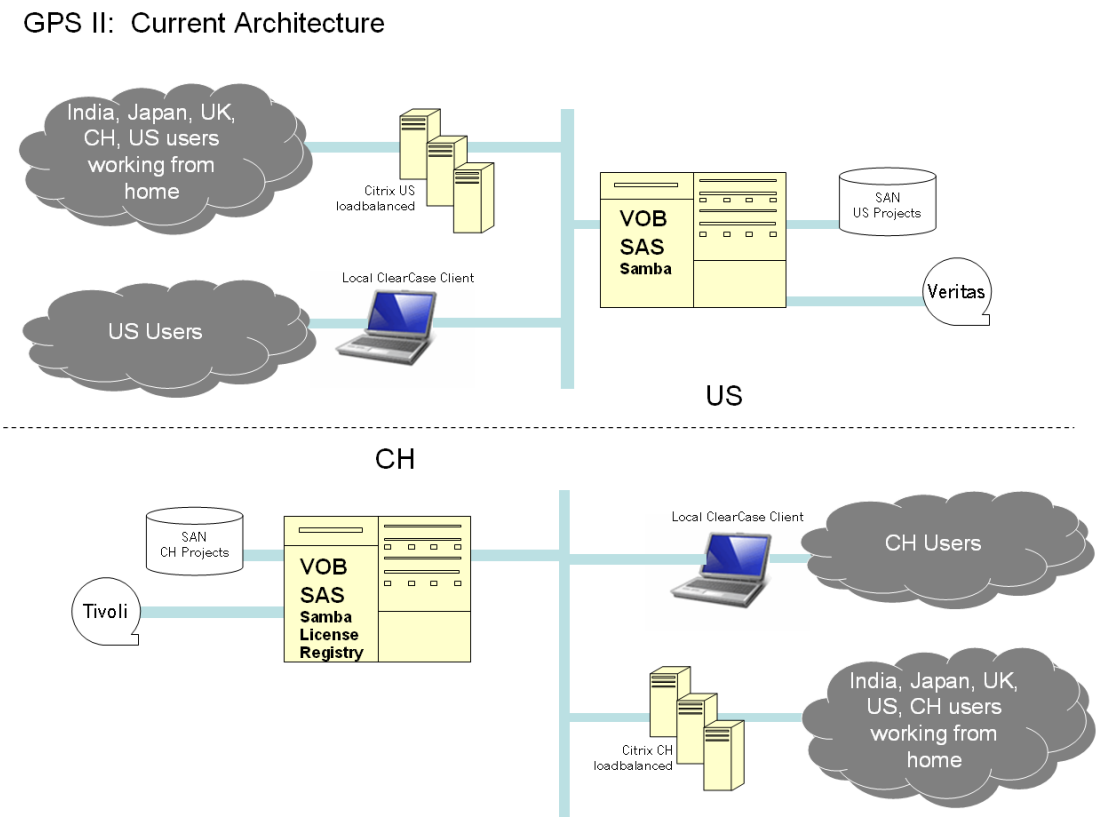


Figure 1: The original system architecture

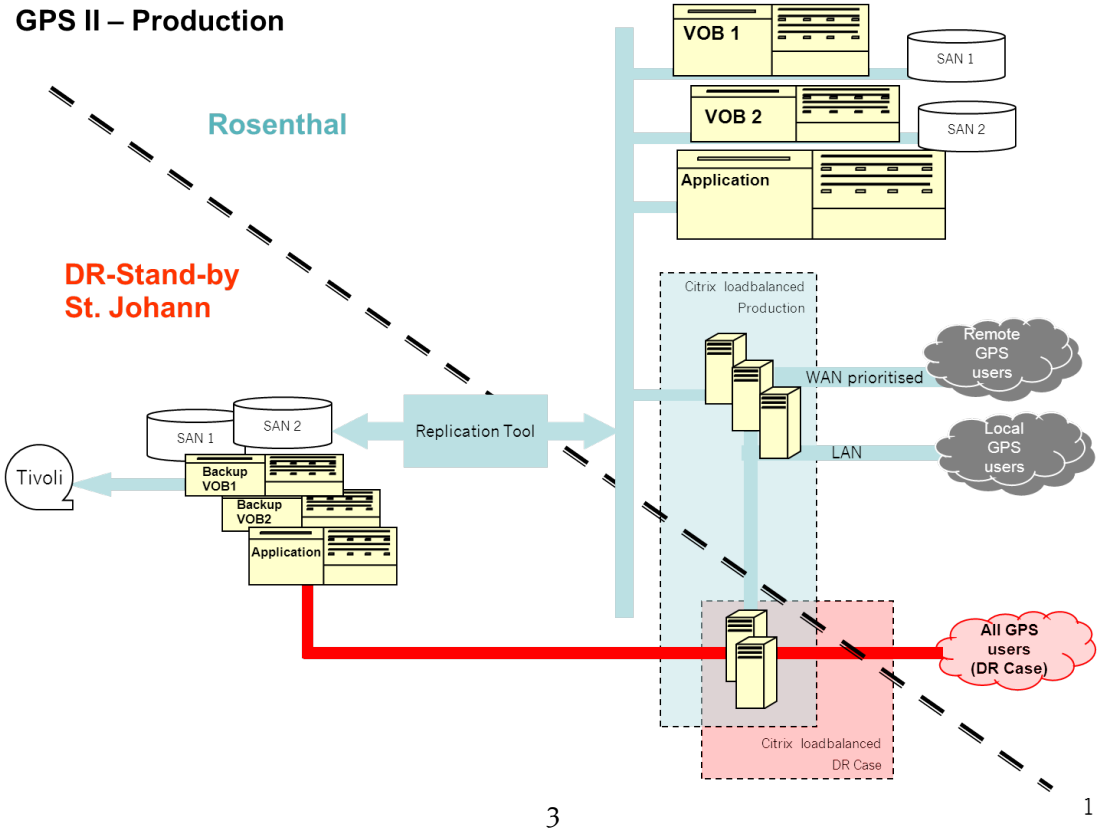


Figure 2: The new system architecture

1. consolidated all servers to Basel and
2. moved all user access to Citrix
3. added Disaster recovery servers (that continually replicating the production environment using ClearCase Multisite from Rational)
4. removed the ClearCase local (PC) client we had used previously.

The projects and their data were distributed between Basel & US so the US projects had to be moved across sites as part of their migration and conversion. The Basel projects also needed to be copied to the new server but this was a copy within the same subnet. This difference meant the process for US projects was more time consuming with the copying to new server taking at least 10 times longer than for the local copy. This factor was important in the development of the conversion process.

The term VOB (Versioned Object Base) used in the diagrams is the ClearCase name for the store for programs , output and data, all of which are versioned. It is equivalent to the repository as used by CVS, Subversions and Perforce. In the original creation of GPS it was decided to use a separate VOB for each combination of compound and indication. A VOB is like a database — it has to be moved as a whole and, for example, locked before backup to maintain internal consistency. In our conditions the VOB was chosen as the unit of migration. In each VOB there are separate studies (up to ca. 200) each of which can have several directories with SAS data. We adopted a policy that studies would be treated as units to be converted once. This implied that if a VOB was converted but one study failed then that one study could be fixed, re-converted and the VOB then declared done. If that study could not be corrected and converted then the whole VOB would be restored to it's pre-conversion state and rescheduled for a later date when the issue had been resolved.

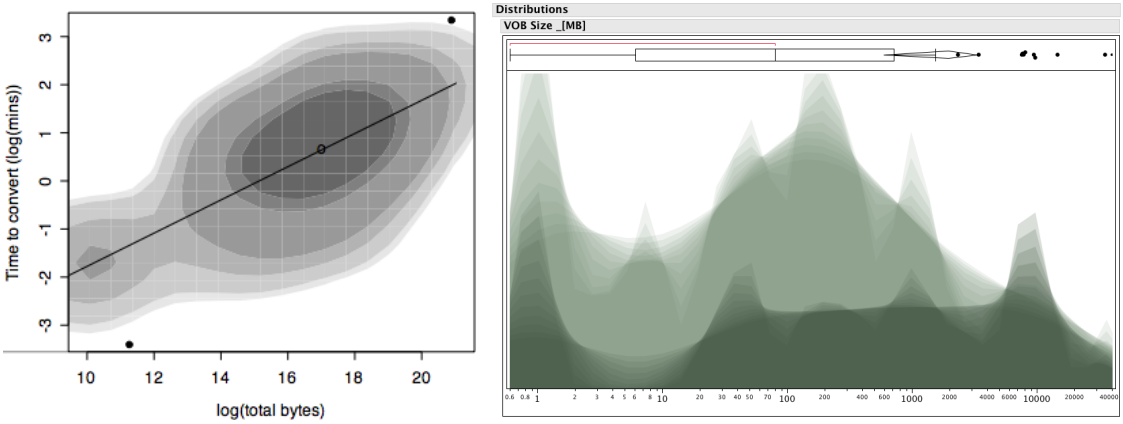
Tests on the performance of the conversion script and the distribution of VOB sizes are to be found in Figure 3 on the next page. These were important in the planning because performance variation was highlighted and it was also clear the load would be too high to complete in a single period of downtime for the system.

2. SAS conversion planning

2.1. First considerations

Planning a SAS conversion is not technically simple because of the range of products and file types used by the SAS system. Some key points to focus on and get right for the conversion plan are:

1. Decide the unit of migration (one dataset?, one directory?, one study?, one project?, one server?, all our SAS files?). A key question is what is the need in the business for roll back if issues are experienced with the conversion?
2. develop an automated, robust, validated process for each migration unit with reliable timings
3. make a complete catalog of the migration units, size, location, owner
4. create a test area with copies of 'typical' migration units copied from the production environment



(a) Times to convert vs VOB Size

(b) Distribution of VOB sizes to be migrated (active VOBs shown darker)

Figure 3: Conversion times and VOB sizes

5. make a thorough survey of what is to be converted in terms of datasets (versions?), catalogs, graphics, item stores, views etc.

6. create a detailed schedule based on realistic timings that determines when each unit will be migrated and is therefore unavailable for use

7. determine the policy to be adopted on retaining old V8 datasets. If you already have the data in a version control system then this becomes much easier.

The details of what can be used and what must be converted are dependent on several factors including the Operating system and hardware architecture in use. There are many types of SAS objects not all of which are convertible. In our case files needing manual intervention were mainly catalogs, views and item stores.

2.2. The conversion script

For the conversion part of the project a validated PERL script written by Thotwave `http://www.thotwave.com/` was used to convert a whole VOB. It was driven by a text file containing a list of what was to be converted and a code to include or exclude that element. It could also be used to convert a single study and was clever enough not to convert already converted studies. It fully utilised the version control system and would only finalise a conversion if the converted files compared equal to the old files. In such a case it would not check the new files in but undo the checkouts. It was decided that only the current versions of data sets would be converted and the last version 8 and first version 9 datasets would be labelled in ClearCase.

The conversion script was an essential part of the conversion project that was already done and will not be discussed further in this paper. One aspect affected the process design and that was the fact that the conversions program could fail

for various reasons (datasets with lowercase names, spaces, other illegal characters, too long file names, changed filetypes, Version 6 datasets, checked out files, etc). Because all such cases could not be located in advance the conversion program had been designed with a scan function which would run a dummy conversion up to the point that SAS would read the data and then check the log for errors. These errors were then consolidated into special logs for the scanning function. Afterwards scanning errors in need of correction could be fixed by users before the live migration. It was important to do this work in advance because if the actual conversion could not proceed quickly any hope of making a fast, one run conversion would be gone and the time frame for conversion could easily become 7-8 days not 1-2. This work needed to be done soon enough that the Program Programmers had time to correct the issues and so it was decided to schedule the scan of each VOB four weeks before the scheduled migration date. At the same time a scan for checked out files was made and the list emailed to those users having checkouts. The aim being to catch files that had been checked out a long time ago because clearly current work would be checked back in the day before migration.

2.3. Datasets and catalog format

The picture is complicated in our case because although we were staying on the same operating system and hardware (AIX, IBM Power P6) SAS only supported a 64 bit version in V 9.2 and so the upgrade is classed by SAS as a platform change. This made catalogs unreadable, datasets unwritable and forced a conversion.

There is confusion about what a '32–bit' application means in the context of SAS. Could this somehow affect the precision of SAS results? Will there be rounding problems with numeric variables in datasets?

Data step and calculations done within SAS procs are always done with double precision real numbers — which is 64–bit (8 bytes). Numeric variables stored in SAS datasets are 8 byte by default — other lengths are only used following a length declaration in the SAS program.

If this is true how could a 64–bit dataset differ from a 32–bit one? The answer is not pretty — it is that the dataset header has 64–bit integer numbers (for counts like number of observations) in it instead of 32–bit. Consequently on 64–bit platforms the maximum number of observations in a dataset is $2^{(64-1)}$ rather than $2^{(32-1)}$. However this difference merely in the header means a SAS operation is needed to convert SAS datasets. This can be PROC COPY or PROC MIGRATE.

There are other file types that cannot be converted and these include those based on catalogs such as format catalogs, graphics catalogs, item stores (written by PROC TEMPLATE) and compiled SAS macros and AF frames. These types of objects must be recreated from their source program using the appropriate PROC. We decided these files would be listed for the users as part of the scan program so they could organise the conversion of these objects once they were on the new system.

To develop PROC MIGRATE code now you can use the PROC MIGRATE calculator see section 8 on page 31

2.4. Roll-back and the unit of migration

SAS does its best to detect what engine should be used for a directory if the default is given. Certainly in version 6 and 8 this can cause problems if some data sets are one version and some are another. For example, you have two datasets `A.ssd` and `B.sas7bdat` (V6 and V8 respectively). Now a SAS program accessing A first will not see dataset B and vice versa. This behaviour can lead to some desperate calls to support. Because of this behaviour we had long had the policy to keep only one version of datasets in a directory. Usage of SAS 9 on a SAS V8 library will not allow writing to the V8 data (this is a cross platform migration remember) and will therefore create a SAS V9 dataset. This would result in mixed version directories unless we could force all files to be re-written together. Because our normal programming process mandates a program per analysis dataset this could be *not* be guaranteed. This and issues of catalog compatibility lead us to decide that the unit of conversion should be the study and the unit for operating would be the project. This means we would schedule and work with projects but within a project we would migrate whole studies. So if there was a problem with a single dataset within a study that would not convert then that study would be rolled back to V8 and the next study treated as a separate case. We would then go back diagnose and fix the issue and then re-migrate the failed study. Once all the studies were successfully done we would release the project back to the users.

We decided that in the case of a VOB where a study was not migrated within the deadline the the users should choose to revert the whole VOB, the VOB bundle, or continue on the new server while the issue with a single study was fixed.

Reading on SAS file formats

There is a portion of the SAS support website devoted to migration and also documentation of PROC MIGRATE or a data step. section 8 on page 31 has some relevant sources of information.

3. How much downtime?

To answer this question we have to develop a conversion process and test and measure it on real data. There is a related question which is how long will each project be unavailable during conversion? As we have seen projects differ widely in size both total size and numbers of studies and indications which makes it impossible to pluck a figure out of the air and then meet that target.

The conversion process had three levels (from bottom up):

1. the SAS PROC MIGRATE to migrate and validate the new content (managed with a PERL script)
2. Migrating content to new servers (and subsequent updating synchronisation)
3. Conversion Process level managing the copying, testing for user checkouts, checking for non-convertible files, and all the other work associated with actually scheduling and performing the conversion

Each of these levels has its own procedures and organisation and scheduling.

A sensible procedure for the first level is given in Figure 4.

```
Create work area on (test) server
For each MU DO
    find all directories containing SAS objects
    For each directory DO
        (assume many libraries per Migration Unit)
        make a temp dir
        if 32 -> 64bit conversion is needed make remote connection
           to a V8 server then
        %before,
        proc migrate
          to temp
          from original location,
        %after / %em
        fi
    end
    validate copy process:
    if any directory unsuccessful then
        mark dir for later diagnosis and repeat proc migrate
    else if directory was moved OK then
        overwrite V8 library with the contents of the temp location
    fi
end
when whole MU is moved OK then
zip all log files and store in place agreed with Data owners
delete all temp areas for the MU
fi
```

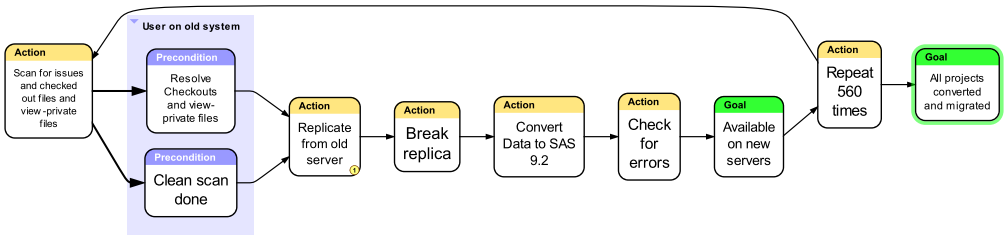
Figure 4: Migration Procedure

PROC MIGRATE and the `%before`, `%after` and `%em` macros are documented on the SAS web site see section 8.

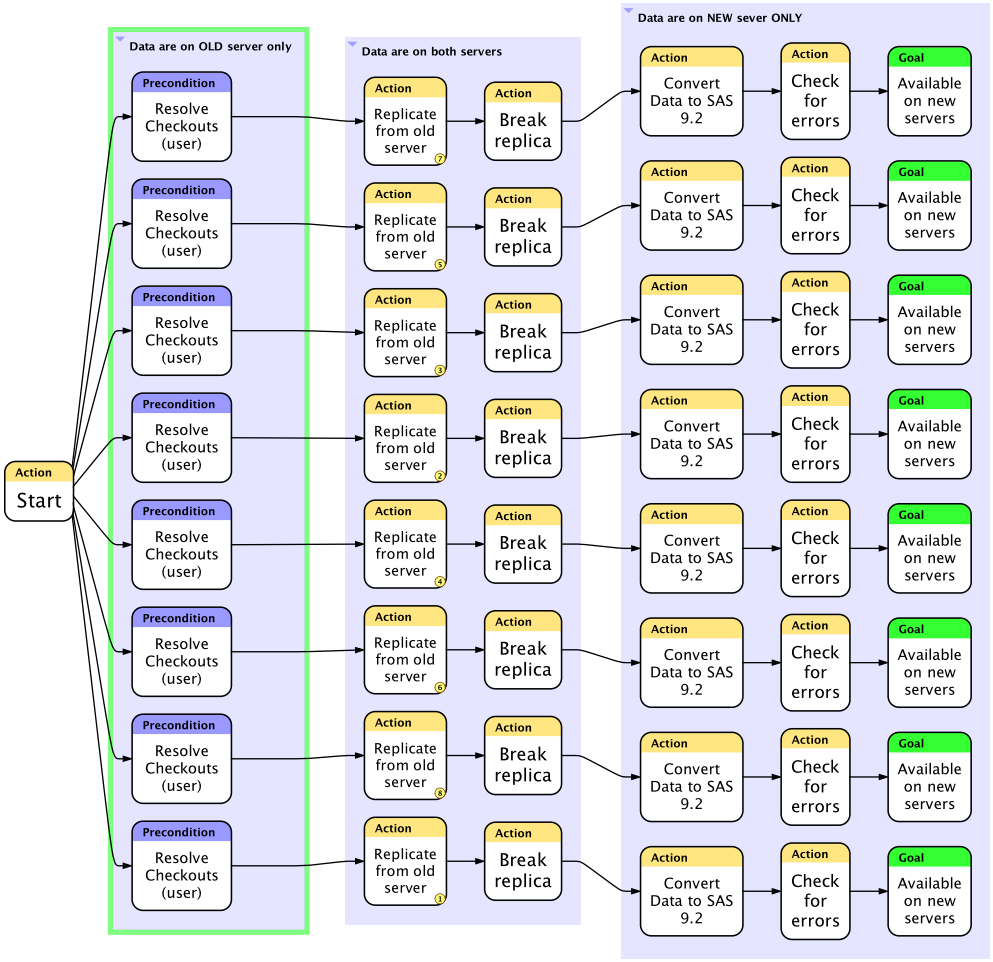
3.1. A migration project as a process

It is useful to distinguish between management of a single project and a production line. A production line uses the same processes with different material and where the problem is to maximise flow (or output minus waste). Generally speaking similar techniques can be used for these two problems although the acronyms are all different as are the trade offs between investment and fast execution.

This migration and conversion was essentially a one off process "to be done once" needing to be planned and costed in enough detail to be enable reliable execution because of its potential to disrupt operational systems. It is inevitable that it would therefore be a process with a lot of preparation and specific tools. The approach we adopted was to maximise the automation as much as possible and to do this using programming tools that were easy to write and powerful enough to create file lists, email users, and so on. This enabled us to reduce the administrative and communication burden on the converters while maintaining the process we had agreed. Some example scripts are in the Appendix.



(a) Process version 2 first version



(b) Eight MUs being migrated in parallel. If the Replication steps can be extracted and done up-front the migration time can be drastically reduced

Figure 5: Process version 2

3.2. Developing the conversion process

It ended up taking a significant amount of work (4 people for 5 months) and not 1.5 people for two months as originally planned. It began with a sketched process needing "a little work" to fill in details.

3.2.1. How did the process seem to start with?

Process version 0

This effectively documented the migration script call only — 5 pages. Note that, correct though it is the commands below allow any naming convention for the input text file and output logs. Clearly in a large effort it is going to be essential to have this and standard names for working directories where all relevant files can be kept. These conventions can be enforced by using a wrapper script to name everything so this was the first script to be written.

1. scan: `nohup sas9migrate_prescan.pl --studylist=myscanlist.txt > myscanlist.out 2>&1 &`
2. migrate: `nohup sas9migrate.pl --studylist=mylist.txt --migrate > mylist.out 2>&1 &`

Process version 1

This version was a powerpoint slide and can be seen on the preceding page. A simple linear procedure of 5 steps - repeated 560 times what could be simpler?

What does order mean in this list of tasks and in this diagram? The order these processes happen has a big effect. For example, we originally planned to

1. do the SAS migration first (on the old server) and then
2. replicate and
3. then break the replica.

The disadvantage of this method would be that most of the processing would be done on the busiest servers and none on the (empty) new servers; therefore this method was rejected and the order refined.

Process version 2

So we refined the process further and further using the information from the timings and experiments.

Notice that the third version looks about the same length but it is not because actually the replication has been done for all VOBs before starting. Once the replication is set up then the new system is kept in step with the old hourly.

3.2.2. Version 3 - measurable

After developing the second version we again tested using realistic amounts of data and from each site and were able to estimate the project total duration. The calculations were complex and done with the planning spreadsheets in Section Figure 9 on page 15

With a migration of this size it is important to consider variability as well as mean durations. We can calculate a rough estimate like this using the PERT[1] approximations to take account of the variability in project size and complexity.

We do not actually have to make distributional assumptions in this case because we know the distributions of size and locations of the full population we want to migrate. Nevertheless I have used the PERT method here because it is accurate enough to illustrate the point and in

[1] http://krypton.mnsu.edu/~tony/courses/609/PERT/tech.html

Version 1
— 1 PP slide

1. Scanning (one time exercise completed before migration)
2. Resolve scanning problems by business
3. Replication with scripts
4. Migration
5. Inform PP about start
6. Run last replication and lock projects on old server
7. Check for errors in log file
8. Run migration script for bundle of VOBs / projects
9. Check log files
10. Unlock VOBs and
11. inform VOB owner if successful or not

Process Version 2
—- 13 pages

1. Get Lists Of Users Per Group For The Target VOBs
2. Inform All Users of the Vob That Migration is due
3. Check If Vob is in A TA Group
4. Replicate Vob From The Release 1 To Release 2 Servers
5. Re-protect The Vob Replica On Rel 2
6. Scan Vob For Potential Migration Problems
7. Inform Migration team CIS team Of The Vob Scan Results
8. Wait For OK From CIS migration team that target VOBs are clean
9. Lock VOB on old server
10. Last search For Checkouts And View-private Files
11. Force A Final Vob Replication
12. Inform GPS Support Final Vob Replica Has Been Made
13. Transfer Mastership To Release 2
14. Prevent back-replication To The Old System
15. Re-scan Vob For Potential Migration Problems
16. Check For New Migration Issues
17. Migrate To Sas Version 9
18. Apply Appropriate Access Right As Saved In Step 5
19. The Target VOBs Are Now Available On The GPS II Rel 2
20. Run Script To Make It Available For Modesim
21. Fix Issues Till Migration Is Complete
22. Communication After Migration
23. Inform MT CIS of successful Migration
24. Inform User About Issues During Migration
25. Inform User About Fallback To Old System if needed
26. Inform All Users Notified In Step 1

Process version 5
— 45 pages

1. Check Sas Labels And Sas Attributes Are Created In New Vob
2. Inform All Users Having Access To The Vob That Migration Is Upcoming
3. Create And Distribute List Of Check-outs Per Target Vob Set
4. Cancel / Undo Last Checkouts
5. Scan Vob For Potential Sas Migration Problems
6. Lock Release 1 And Run The Final Vob Synchronisation
7. Import Final Vob Changes To Release 2
8. Checks Before Breaking Replica
9. Migration Abort Deadline
10. Break Replication From Old To The New Environment
11. Remove Any Remaining Checkouts From The Source Server
12. Last Scan Before Sas Migration
13. Check All VOBs To Be Migrated Have The Correct Permissions
14. Final Checks Before Running Sas Migration
15. Migrate To Sas Version 9
16. Check SAS V9 Migration Logs For Issues
17. Assess Reasons For Failed (Study) Migrations
18. Remove Checkouts Left By An Incomplete Migration
19. Rerun Failing Studies
20. Apply Appropriate Access Rights
21. Release VOBs For Access
22. Make Vob Available For Modesim
23. Make Any Special Groups Available From Citrix Servers
24. Inform Users Migration Of The Project Is Completed
25. Execute Fall Back To Release 1 Data
26. Precursor Setup And Replication Activities

Figure 6: Versions 2,3 & 5 of the migration process. Some steps from V3 have been moved to a separate process not listed here and the process for CNC bundles is also in another document

fact even given the exact sizes there was still massive variability in times for replicating, breaking replicas, and running the SAS migrations. The figure of 231 weeks is very high but is calculated as if the projects could only be migrated one at a time in sequence. If we had infinite resources of people, computer power and network capacity we could migrate all projects in parallel. When executed in parallel the total time needed is not the average time for a MU but the time the longest one takes — approx. 1.6 weeks. In this migration there were three projects with over 100 GB of data.

To make a more realistic calculation we have to take constraints into account such as the number of projects that must be migrated over a weekend.

Constraints: throughput per day, people trained to run the migration, network, time to close up production while migrating.

To calculate the duration we can use the PERT method and that gives some idea of the total duration After applying the constraints we can estimate the total duration in calendar months.

The time estimates showed the process would take a year for two people and reasonable assumptions about how much we could do per week. So it was agreed we would have extra people to help but they could not be available right away and the replication step was not only very variable (depending on size and network load) it took too long to get a batch completed in 48 hours.

3.2.3. Version 4 - ripe for refactoring

Having created a workflow with Excel and documented it with pencil diagrams and a Word document of 30 pages we started dry runs of version 3 and discovered all kinds of practical problems. We ended by redesigning the process almost completely.

The simplifications we made meant taking parts of the process and separating the operations to do them into parallel streams — we changed the linear nature of the process we had created. I suspect we got into that problem because the tools we used *imposed* a linear structure when actually there was not one. At some level we were aware of that, but once listed and linearised (as must be done in a document or a set of spreadsheet rows) thinking is prevented made to seem superfluous. In a perfect environment this might not matter because there would be time to reconsider. In real life this can be a problem.

> *Why should we spend time drawing diagrams when we have to write programs!*

So we looked at the whole process in our document and I have diagrammed it in Figure 5 but showing just 8 projects of the 560. I have added grouping boxes that show where the data are at any one time. Data starts only on the old servers, then for a while it resides on both and the old server remains the master copy. Then the replication relationship is broken and the old copy locked to prevent changes during conversion, and after that the SAS migration starts.

Here there was clearly a chance to make all the replication steps up front (main-

[2] an easy to use tool for charting and brainstorming because all objects are easily joined and layout is completely automatic. Http://www.flyinglogic.com/

Step	Min - Max	Duration (minutes)	Expected duration, sd	CV
Replicate	30 – 600	60	145, 95	38
Break Replica	10 – 600	60	142, 30	22
Convert to SAS	30 – 720	89	178, 115	64
Check, Redo	1 – 720	60	160, 120	75
Correct permissions	40 – 500	50	123, 75	62
Total (one replication)	80 – 3100	310	743, 207	68
Total (560 cases) (weeks)	25, 975		231	

Table 1: Table of estimates for duration of data migration, from Webcalc.net

taining synchronisation) and then later (as scheduled) do the rest of the migration.

Figure 7 shows the revised migration process diagrammed with FlyingLogic[2].

This is the process we started working with, the replication was treated as an operational change and the migration document modified to reflect the changed process. Now we had removed a major source of delays and uncertainty the total duration looked much better.

3.2.4. Version 5 - up and running

We re-tested some conversion timings only to find a conversion still took too long and in addition the extra team members requested were not on board fast enough (35 days late).

Refactor

We re-examined our high priority projects and projects that needed weekend migration due to users' workload and realised that this list was not very long. It also defined the minimum time the migration could take. We could therefore allocate those bundles to weekends and use that as a skeleton around which the other projects had to fit into. We looked more carefully and realised that a large number of projects were not in active use. Why were we spending effort scheduling inactive projects when it obviously did not matter when they were done. And more importantly there was really no need to fix in advance when they were done. We could decree that these inactive projects would *not* be converted on a schedule (they are not being used, right?).

So we decided to create a simplified conversion process for dormant projects and this enabled us to make two streams of work that could run in parallel. One low priority and shiftable and the other fixed and to be carefully managed. This enabled us to always have background work available if there were delays to the main conversion, and also to temporarily suspend the CNC work if we needed extra work on the mainstream. This increased utilisation of our time and protected us from schedule deviations to some extent.

This cut our project list for scheduling to about 300.

Process for inactive projects

This new process meant they were closed up (no check-ins remaining), copied to the new server, their permissions restored and left as SAS V8.2 files. This would suffice for re-running old outputs. If new data was to be added or if the analysis data sets needed to be rebuilt then a con-

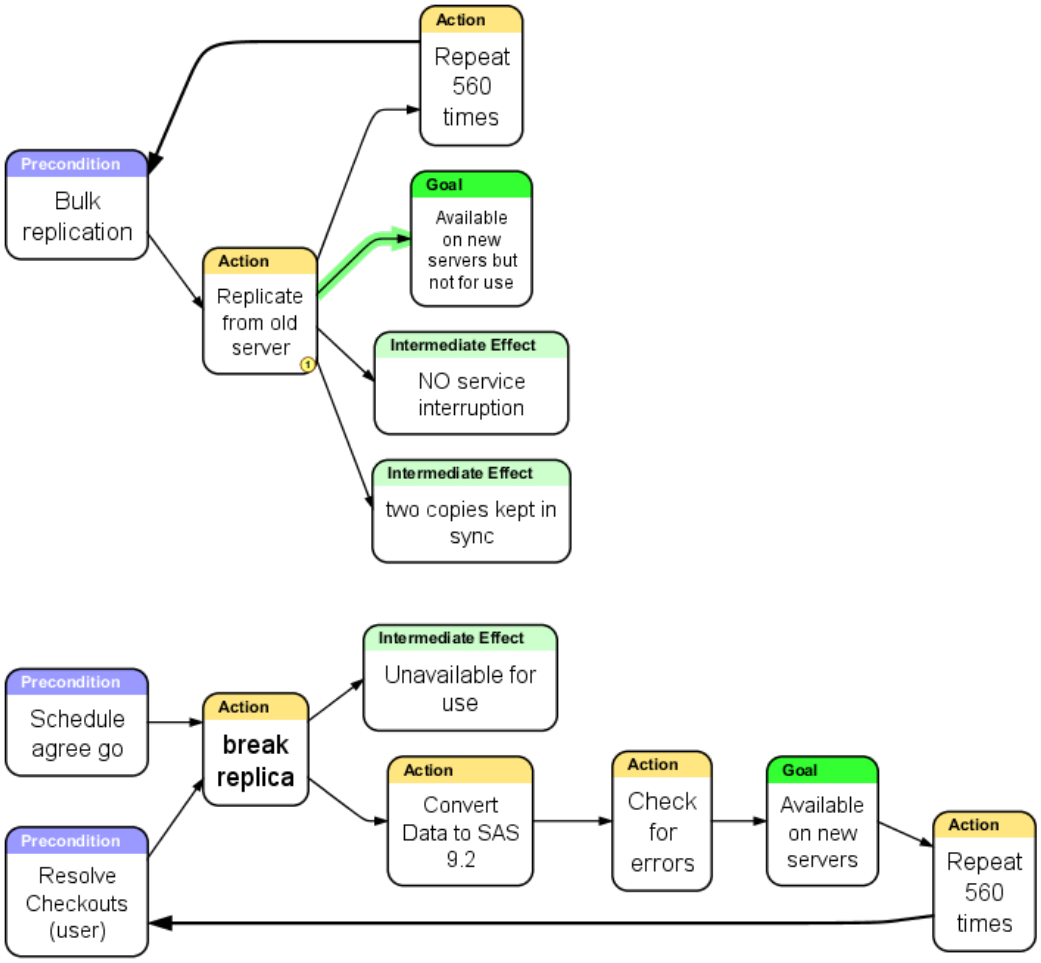


Figure 7: Migration Process Version 3 after split into replication and migration streams.

version could be done. We therefore created a special process that could convert a project on the new servers that could be requested by users as a service request at any time in the future. We called these CNC (Copy, No Convert) projects.

This slight scope shift was the vital last step towards creating a viable process and timetable.

4. When will my project be unavailable?

Given the scale of this migration, the scheduling of the projects was never going to easy, as we found out. It was obvious to us that the scheduling of some projects or groups of studies was going to be easier scheduled than others due to activity on the projects, more specifically studies on that project at a certain time. In order for the scheduling process to work and

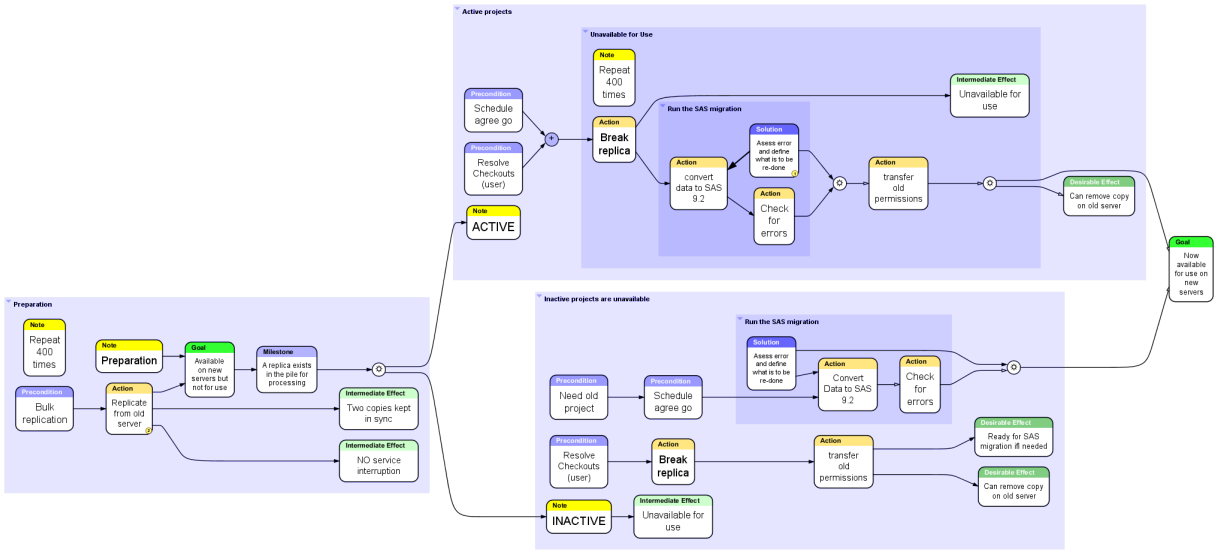


Figure 8: Processes Version 4 with three streams and now dependent on project type

	TA	PP Site	Migration Champion	Program Programmer/ VOB owner (Responsible for IQ Sign off)	Program Statiscian/ VOB owner	VOB Bundeling (project level)	VOB (Indication)	Migration Treatment C=Copy Only M=Full Migrate X=Special	Active VOB	Active Project	VOB Size [MB]	VOB Size [GB]	Size VOB Bundle [GB]	#VOBS	#VOBS Spec Protect	#Studies	#Studies per VOB spec. Protection	#Groups per Vob spec. Protections	# Studies in CPV per VOB	Scanning Effort (h)	Scanning Duration (h)	Mig Effort (h)	Mig Duration (h)	Migrator
2	IID	BS	Pradeep Shetty	Philippe Le Claire	Karthinathan Thang	Project 1	Indication 1	M	Active	Active	7832	7.6	8.54	3	0	42			1	1	2	2	3	Y
3	IID	BS	Pradeep Shetty	Philippe Le Claire	Karthinathan Thang	Project 1	Indication 1 restricted	M	Active	Active														
4	ONCOLOGY	IN	TBC	Suoy Mondal, Karthik	Sundaramoorthy	Project 1	Indication 2	M	No VOB	Active		0.0	8.54	3	0	42				1	2	2	3	
5	IID	BS	Pradeep Shetty	Philippe Le Claire	Karthinathan Thang	Project 1	Indication 3	M	Active	Active	913	0.9	8.54	3	0	42			1	1	2	2	3	
6	IID	BS	Pradeep Shetty	Philippe Le Claire	Karthinathan Thang	Project 2	Indication 1 restricted	M	Active	Active														
7	NSO	TBC	Stephan Mynhardt	Suoy Mondal, Karthik	Sundaramoorthy	Project 3	Indication 1 restricted	C	Unknown	Unknown														SS
8	IID	EH	Pradeep Shetty, Stephan M	Tina Chen	Yanming Yin	Project 4	Indication 1	M	Active	Active	9691	9.5	9.50	5	0	25			1	2	3	2	4	VN
9	IID	EH	Pradeep Shetty, Stephan M	Tina Chen	Yanming Yin	Project 4	Indication 1 restricted	M	Active	Active	37	0.0	9.50	5	0	25	1	1		2	3	2	4	VN
10	IID	EH	Pradeep Shetty, Stephan M	Tina Chen	Yanming Yin	Project 4	Indication 2	M	Active	Active	30	0.0	9.50	5	3	25				4	4	4	5	VN
11	IID	BS	Pradeep Shetty, Stephan M	Tina Chen	Yanming Yin	Project 4	Indication 2 restricted	M	Active	Active	1	0.0	9.50	5	3	25	1	1		4	4	4	5	VN

(a) Part I Contact, size and criticality information

	TA	PP Site	Break Replica & Sync Due	Proposed Start Date	Proposed End Date	Proposed Week #	Proposed Month	Downtime (days per migration)	Weekend	Migration Complete	Actual Date Completed	Actual Week # Completed	Critical in June 2009	Critical in July 2009	Critical in Aug 2009	Critical Sep 2009	Critical Oct 2009	Critical Nov 2009	Critical Dec 2009	comment
2	IID	BS		Tue, 04/08/09	Wed, 05/08/09	32	Aug	2	No	Yes	Wed, 12/08/09	33	non-cr	non-cr	non-cr	non-cr	non-cr	non-cr	non-cr	Vob on GPS server and PP/PS assigned
3	IID	BS		Tue, 04/08/09	Wed, 05/08/09	32	Aug	2	No	Yes	Wed, 12/08/09	33	non-cr	non-cr	non-cr	non-cr	non-cr	non-cr	non-cr	Missing from sheet - added on 19.8.09
4	ONCOLOGY	IN		Tue, 04/08/09	Wed, 05/08/09	32	Aug	2	No	Yes	Wed, 12/08/09	33	non-cr	non-cr	non-cr	non-cr	non-cr	non-cr	non-cr	VOB not existing on server
5	IID	BS		Tue, 04/08/09	Wed, 05/08/09	32	Aug	2	No	Yes	Wed, 12/08/09	33	non-cr	non-cr	non-cr	non-cr	non-cr	non-cr	non-cr	Was lost from schedule? - Re added 1.7.09 - AMS
6	IID	BS		Tue, 04/08/09	Wed, 05/08/09	32	Aug	2	No	Yes	Wed, 12/08/09	33	non-cr	non-cr	non-cr	non-cr	non-cr	non-cr	non-cr	Missing from sheet - added on 19.8.09
7	NSO	TBC	Wed, 28/10/09	Thu, 29/10/09	Fri, 30/10/09	44	Oct	2	No	Yes	Thu, 29/10/09	44								Missing from sheet - added on 16.10.09
8	IID	EH	Mon, 26/10/09	Tue, 27/10/09	Wed, 28/10/09	44	Oct	2	No	Yes	Tue, 03/11/09	45	critical	critical	critical	critical	critical	non-cr	non-cr	Vob on GPS server and PP/PS assigned
9	IID	EH	Mon, 26/10/09	Tue, 27/10/09	Wed, 28/10/09	44	Oct	2	No	Yes	Tue, 03/11/09	45	critical	critical	critical	critical	critical	non-cr	non-cr	non-critical time period
10	IID	EH	Mon, 26/10/09	Tue, 27/10/09	Wed, 28/10/09	44	Oct	2	No	Yes	Tue, 03/11/09	45	critical	critical	critical	critical	critical	non-cr	non-cr	Vob on GPS server and PP/PS assigned
11	IID	BS	Mon, 26/10/09	Tue, 27/10/09	Wed, 28/10/09	44	Oct	2	No	Yes	Tue, 03/11/09	45	critical	critical	critical	critical	critical	non-cr	non-cr	non-critical time period

(b) Part II Showing feedback on migration timing, proposed and actual dates

Figure 9: The Migration Schedule Spreadsheet

to work well, we needed feedback from those people who knew best of when a project or groups of studies would be best suited for migration.

For this, the core team, led by the project manager, produced a clear and succinct email communication that would be sent to the Program Heads of each therapeutic area, Statistics and Programming — the most numerous group of SAS and GPS II users — and managers of others areas involved (Data Management, Database Programming and CRF Development) in order for them to get feedback on all of their projects from within their area of expertise. This email communication detailed the following:

- Project scope,
- Project timeframe,
- Migration plan and scope,
- Expected downtimes,
- Risks,
- Required actions before migration

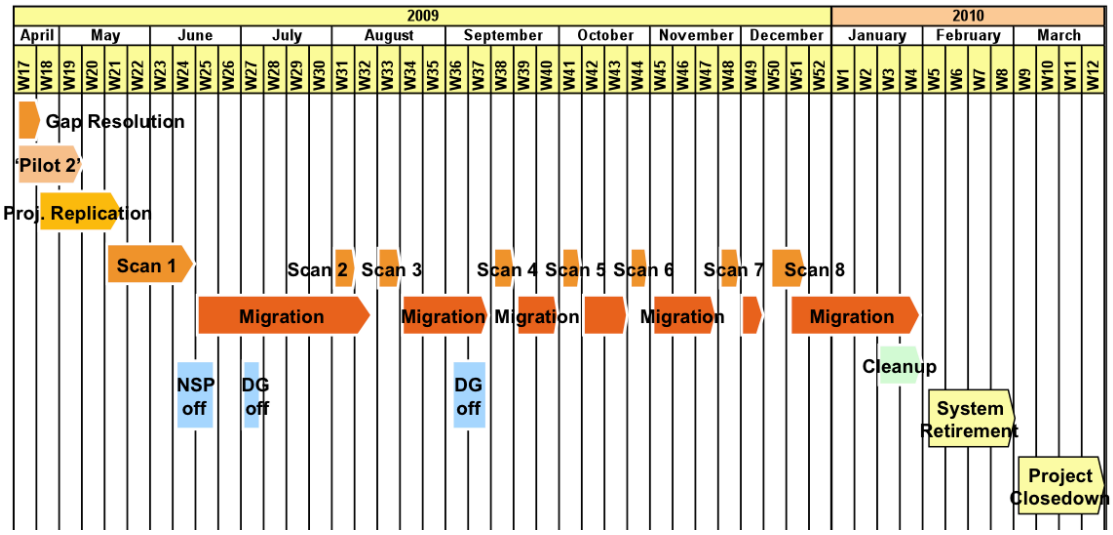
One of the key pieces of feedback we required per study, was if there was a time period that it could NOT migrate and if not, then when was the best time for it to migrate. Within the communication email we had stressed there was a 6 month time window in order to complete the whole migration (Jun 2009 – Nov 2009). Needless to say, as always, we got a few awkward teams stating that no time within that period was a good time. After direct discussion with these teams by the project manager and business coordinators, we were able to come to an agreement that suited all concerned.

The migration schedule document type that was used was Excel as this was deemed the most user friendly. This document was a living document (updated daily) and stored in a central location for everyone to read and use as needed (hence the need to be user friendly). This document was the key document for the migration team in the whole migration process and the central point of reference for key details. A sample of the spreadsheet can be seen in Figure 9 on the preceding page.

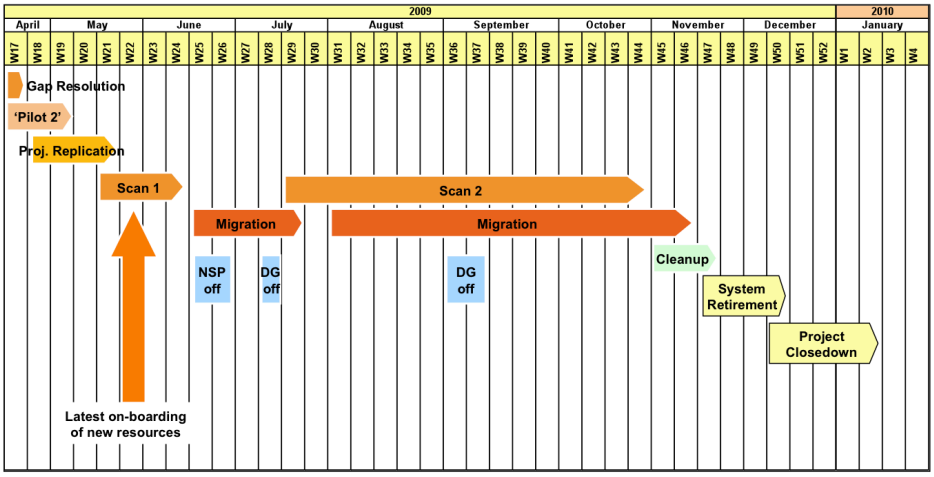
With the aid of the business coordinators, migration champions and managers, the team were able to class each project or groups of studies into bundles and also define which bundles were VIP ones (bundles will be referred to from this point onwards). A VIP bundle was defined as one that had studies close to submission and were deemed highly important to Novartis management. In order to minimise disruption from the daily business work, VIP bundles would generally migrate at the weekend and therefore have less disruption to the business teams during normal working hours.

With the information received from each of the business areas concerned, the project manager was able to put together a draft migration schedule for the migration period (the 6 months mentioned above), together with other relevant details. Some of these factors were known to affect the duration of the conversion process and others were there to record the status. A few are listed below:

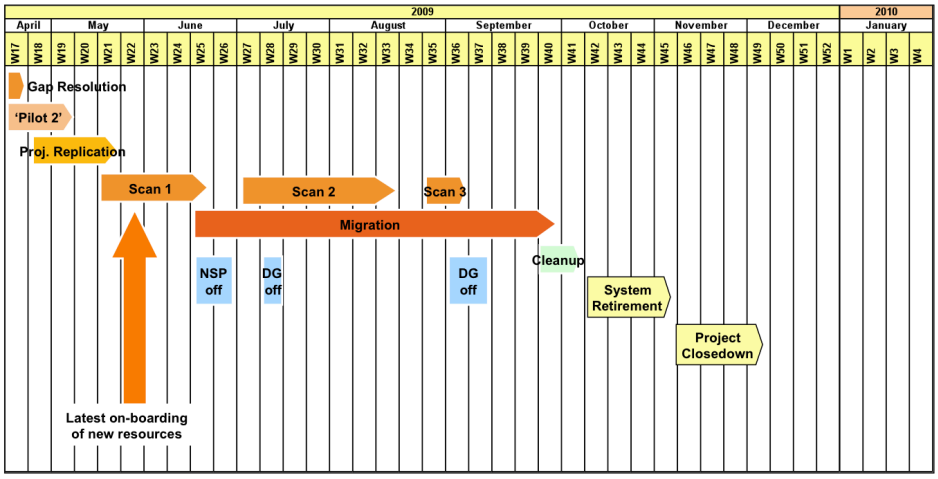
- Bundle migration per week,
- Start and end date of migration,
- Categorisation of bundles (VIP 1 to 3, Inactive or Unknown),
- Weekday or weekend migration,



(a) Team of two (1.4 FTE) and replication — 9 months



(b) Team of three — 7 months



(c) Team of four - ends October — 4 months

Figure 10: Duration estimates for the migration under three scenarios

- Migration type (CNC or F),
- Percentage migrated by therapeutic area,
- Name and location of key contacts (Project Statistician and Project Programmer) per bundle,
- Name of Migration Champion assigned,
- Critical months / weeks identified (if any),
- Expected downtime per bundle,
- Migration status,
- Requests for reschedule,
- Symlinks (automatic links setup from study to study or one project to another)

4.1. Scheduling methods and considerations

4.2. VIP Bundle meetings

We classified bundles into different categories dependent on whether these were seen as high priority by Novartis management and whether these were close to submission. As part of the process plan development, we had performed several dummy migrations for a number of bundles, including one VIP bundle. As part of this dummy run, we had sat together with the VIP bundle team in order to discuss the steps involved and what would be required of them. We received the feedback from this team, that such a meeting was highly beneficial for them, highly reassuring and an opportunity for them to ask any questions directly to us. Therefore we decided that , because of the importance of those bundles with a VIP 1 to 3 category it would be good to have similar meetings with the appropriate project teams. This enabled us to clarify any outstanding issues or questions the teams might have had and also to describe the process we had developed. In particular the sign off process was described. Some teams were uneasy about signing off the migration before being able to access their files but as the process description made clear

1. the datasets were guaranteed to be the same
2. a cancellation and re-migration was aways possible. Although it meant abandoning all work on the new platform, which as time passed became more and more of a factor. In the end no projects chose this option because of SAS issues or other incompatibilities.

4.3. Migration preparation document

In order to make the migration process as smooth as possible, the business coordinators put together a document for use by the Project Statistician and the Project Programmer (the key contact people for each VOB bundle). This document gave them guidance on what they needed to do to enable a smooth and easy migration followed by smooth start-up on the new system post-migration.

Pre-migration tasks involved manual checking for situations known to cause issues with the migration process and then remedying these (checked out files, files not under source control, etc.), from within ClearCase®

	ID	Question	Answer
2	Q-001	Are there any procedural differences between sasv8.2 and sasv9.2?	Project team has discovered some default settings have changed eg proc lifetest and have inquired with SAS Institute to get a list of any other such changes. List of changes already found by team can be found in GPS IMAN page under 'Knowledge database for identified SAS (code) issues'.
3	Q-002	Can the migration schedule be changed at short notice	Yes, obviously a project may have due dates changed or receive ad-hoc requests which may result in the need for migration date changing, this can be done with managers approval. The migration board will meet bi-weekly and update / communicate the updated schedule
4	Q-003	Will sasv8 still be available on new environment?	Yes, but will be only available via GMF and Unix Please submit IMAN ticket to get details how to implement update: GMF CSV file to be adapted to change compiler • default = SAS calling SAS 9.2 • to be changed to SAS 8 executable • User is asked to get in touch with DI team, as this re-run should be done only in very few cases • DI team is currently verifying restriction to prevent access to SAS 8 executable, by assigning users to specific groups
5	Q-005	Where can I find an overview which project have been already been migrated	Ongoing status can be found in GPS IMAN page under 'GPS II Release 2 Project Migration Schedule' see column titled "Migration Complete"
6	Q-006	What is a migration champion?	An appropriate person has been allocated per TA per site, that will be used as a first point of contact during the migration period. This is the migration champion.
7	Q-007	How are symlinks over projects handled?	All symlinks will be taken over to new environment. The unit of migration is the project, thus if symlinks within a project then there should not be an issue. However, if across projects, then if possible , make them migrate in similar time otherwise the link will be tentatively unavailable until next project migrated). All symlinks will be taken over to new environment, though it is definitely worth checking that these are working properly.

Figure 11: Part of the FAQ document

Post-migration tasks were tasks such as checking all personnel had performed the training for the new systems (accessing the new environment, SAS v9.2 training, SPLUS training etc), confirmation that restricted areas had been protected correctly, etc. This document also contained links to useful migration material such as the migration schedule (as discussed in section 4 on page 14 , FAQ document (discussed in the next section), and known SAS code differences given in section 6 (pages23–29).

4.4. Frequently asked questions (FAQ)

In the early stages of GPSII upgrade project we began a frequently asked questions document so business personnel could see if their question had already been asked and perhaps get immediate questions answered. As expected, what began with only few general questions in the early stages of the project, finished with a comprehensive list of approximately 100 questions. Questions varied from specific software questions to practical questions about the process.

- Are there any procedural differences between SAS v8.2 and SAS v9.2?
- Can the migration schedule be changed at short notice?
- Should my CRO be told to hold back on transfers until my migration is done?

An extract from the FAQ can be seen in Figure 11. The FAQ document was stored in a central location for everyone to read and use as needed and was updated on a regular basis or as and when needed. The use of this FAQ document, along with other related documents, can be seen diagrammatically in Figure 12 on the following page.

4.5. Pre / Post Migration Issue Management

From as early on as the initial pilots performed to obtain benchmarks, it became clear that we had to put in place a system for migration issue management, otherwise the business coordinators and migration champions would soon become inundated with numerous requests for help. As you can see from Figure 12 a number of documents and roles were used to develop the issue management process, whereby the business user would have a

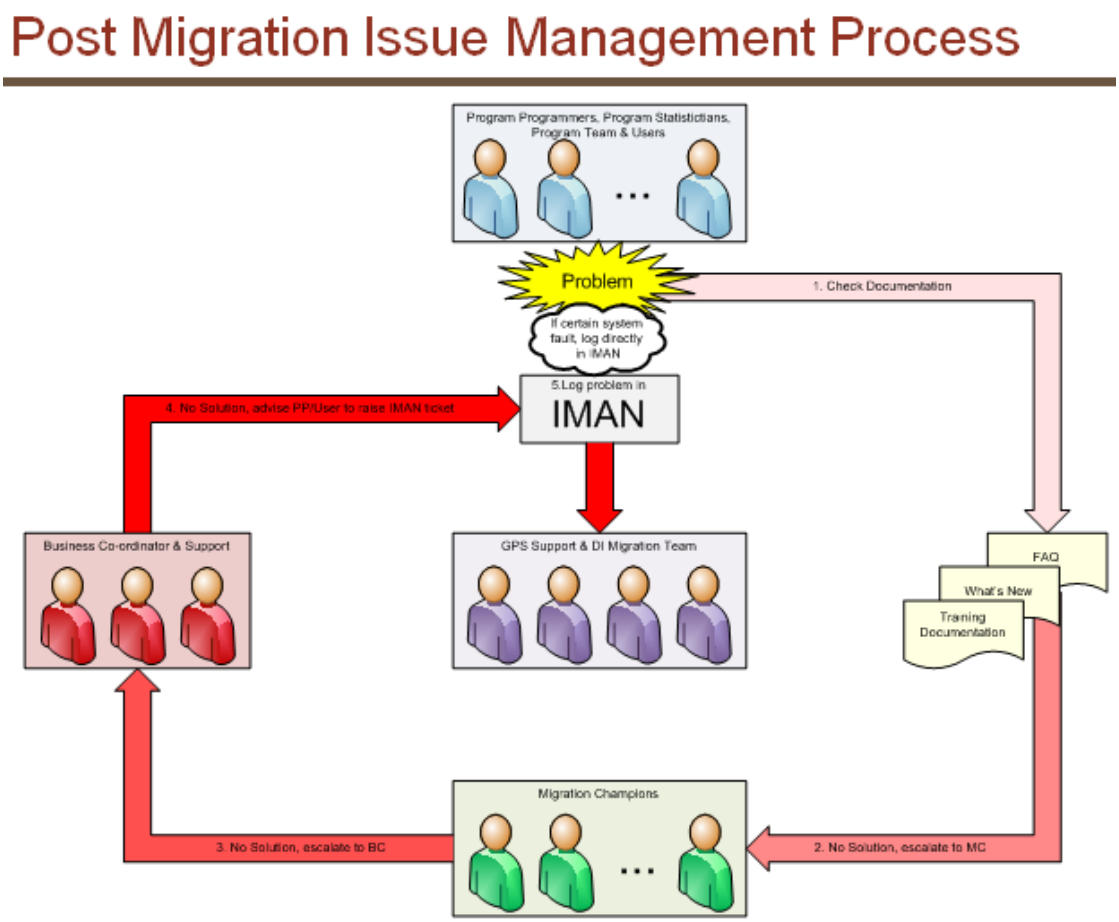


Figure 12: Post-migration Issue Management

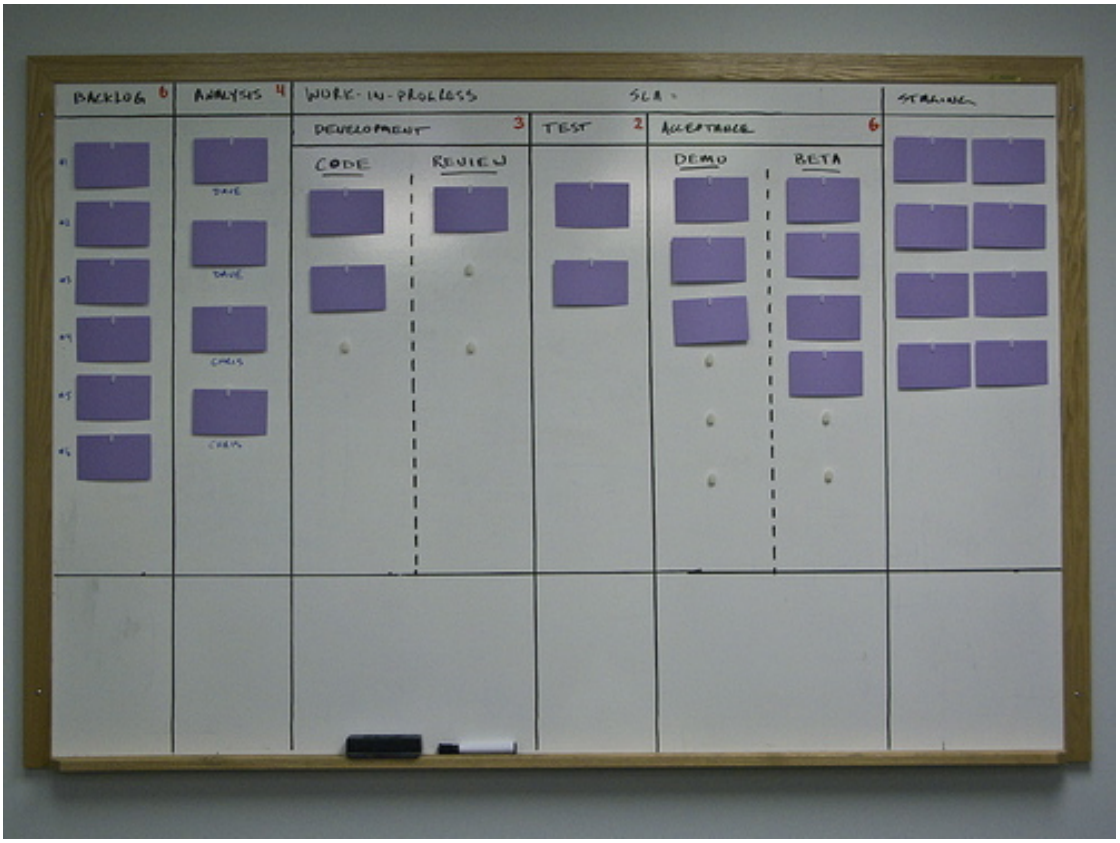


Figure 13: A kanban board (not ours, but similar and neater)

clear path in which to follow up their issue or query.

We originally planned to use this process this for post migration issue management, but it soon became clear that this applied for all issues whether pre, during or post migration.

Firstly, the business user should access their issue and access whether the issue is a system fault or otherwise. If a system fault, the issue should be logged immediately into IMAN (the Novartis Incident MANagement system). If the issue is not deemed a system fault, then the business user should firstly check the available documentation on IMAN e.g. FAQ, Training documentation etc. If the issue is still not solved after checking the documentation, then the next port of call should be the migration champions. The migration champions attended the larger GPS II upgrade meeting on a weekly basis and were updated with any new issues than arose and where therefore up to date on the issues being discovered. Should the migration champions be unable to solve the issue, then they would refer this to the business coordinators to try to get the issue resolved. If still unsolved at this point, an IMAN ticket would be lodged by the business user.

We adopted this systematic approach of addressing migration issues because we wanted to reduce the number of IMAN

tickets for issues that already had been addressed and for which a solution had been found. We also wanted to spread the work load from questions on issues across all resources (documentation, migration champions, business coordinators and IMAN). As new issues were discovered we updated the documentation. The FAQ document was updated on average once a week.

5. How is it going?

This is perhaps the commonest question we heard for the the six months we were running the migration. You need to know the answer both for yourself and for your customers.

We tackled this question in two ways. First we kept the schedule updated and added a dashboard showing various measures of progress. The main one of these was a cumulative graph of migrations done plotted vs date with also a line for expected number. This summary was updated and published for all users to see.

For ourselves we kept a whiteboard with columns for Done, scanning, next week, this week, next week. On this board each VOB bundle had a post-it note with its name and other information. In our Monday weekly meeting we allocated each VOB bundle to one or more people and as each VOB bundle worked through the conversion process we moved it on the board. then the next week we emptied the done column and added the new VOB bundles for next week. We all found this method very useful and I think the only change I would make would be to keep all the done stickers somewhere as a reminder of how much work we got through. This technique is used a lot in the Kanban[3] approach to process management. See example in Figure 13 and a discussion about using Kanban[4] for monitoring a project's progress.

6. Finding differences between SAS v8.2 and SAS v9.2

There are three areas of difference to be aware of when making a SAS conversion between SAS versions.

SAS code changes This is when a SAS program has to be changed (rewritten) because a language feature is changed or removed. An example is the use of the `Do over` loop deprecated since version 6 and removed from the documentation in version 8. In some future SAS version programs with this construction will no longer work.

SAS behaviour changes This is where the program is unchanged but has a different effect. This can be caused in several ways, the most likely are changes to the names or values used for variables in output datasets, changes to table names in ODS, and changes to default options in statistical procedures that affect parameterisations or the order of parameters.

SAS data format changes SAS usually 'refine' the proprietary format of datasets of catalogs at major releases. Since the filetype change

[3] http://www.agileproductdesign.com/blog/2009/kanban_over_simplified.html
[4] http://www.personalkanban.com/pk/primers/you-cannot-yell-at-a-board-with-stickies-on-i

between V6 and V8 the filetypes have not changed and this makes it harder to know instantly if a directory has been converted or what SAS version a given file is.

For some time this was a subject generating more heat than light and with added FUD spread by SAS's inability to provide a comprehensive list of changes that could affect existing code.

Code and behaviour changes Through the dummy run that was performed on a key Novartis project (this was of substantial size) the team were able to establish some key differences between V8.2 and V9.2. This, together with research on the internet, we were able to make an initial list of known differences. Obviously, the more migrations that were performed, the more differences we could expect to find. We looked into this issue and concluded that without a reliable list of changes we could search for in code we had to adopt an approach of "convert but test".

We found some differences and these are listed and discussed in the section below on SAS code. Compared to previous SAS version transitions and migrations I (DG) have worked with (V5.18 -> V6, V6 -> V8, V8 -> V9, MVS to AIX and VMS to AIX) this was the easiest so far as far as SAS version changes go.

This key differences list was stored in a central place (like the migration schedule, FAQ document etc) and updated as needed and readily available for the business to reference and use. The issues we found are described on pages 23–29. We are sure this falls short of a complete list of the differences, but it is derived from 4Tb of studies so it certainly covers the SAS areas we use extensively. We were also able to provide the business with a scanning tool to search their project for the known differences that we'd found. `checkkeyword.sas` was stored in a central location under version control for everyone to use at their discretion.The code uses a txt file, `checkkeyword_metadata.txt` (stored with the program) where parameters like which directories to scan and what to scan for were stored and could be updated. When additional differences were found they were added here.

6.1. SAS code changes

6.1.1. Change in the file name function / option

The second argument of FINFO is a text string requesting particular details. The string needed to get the name of a file has changed in V9.2.

```
/* SAS code in SAS V8.2 */
Function FINFO(file-id,File name)
```

```
/** SAS code in SAS V9.2*/
Function FINFO(file-id,Filename)
```

6.1.2. Extra %then after %else

When using `%if %then %else` statements, a "lagging" `%then` cannot be used.

```
/* SAS code in SAS V8.2  */
%if &Stephen="useless"
%then %do;
    %put &Stephen is useless;
%end;
%else %then %do;
    %put &Stephen is not useless;
%end;\columnbreak
```

```
/** SAS code in SAS V9.2 */
%if &Stephen="useless"
%then %do;
    %put &Stephen is useless;
%end;
%else %do;
    %put &Stephen is not useless;
%end;
```

6.1.3. The macro in operator

The `in` operator is now available in macro `%if` tests. It can be spelled out or the shortcut # character can be used. The separator between alternative values is <space> by default but can be changed by a SAS option. This is a powerful addition to the macro language. However it creates a new issue if you could have a # character in your strings. You can turn off the `in` operator with a SAS option or modify your `%if` tests. See http://support.sas.com/kb/35/591.html

```
options minoperator;
%macro test;
   %let qq=#;
   %put &amp;qq;
   %let qqq=&amp;qq;
   %let a=#; %let b=in; %let c=;
   %if &amp;b EQ &amp;b
   %then %put the # is in '&amp;qq' and &amp;qq is &amp;qqq;
%mend test;

165  options minoperator;
166  %macro test;
167     %let qq=#;
168     %put &amp;qq;
169     %let qqq=&amp;qq;
170     %let a=#; %let b=; %let c=;
171     %if &amp;a EQ &amp;a
        %then %put the # is in '&amp;qq' and &amp;qq is &amp;qqq;
172  %mend test;
173  %test
#
ERROR: Operand missing for # operator in argument to %EVAL function.
ERROR: The macro TEST will stop executing.

175  options nominoperator;
176  %macro test;
177     %let qq=#;
178     %put &amp;qq;
179     %let qqq=&amp;qq;
180     %let a=#; %let b=; %let c=;
182    %if &amp;a EQ &amp;a
```

```
      %then %put the # is in '&amp;qq' and &amp;qq is &amp;qqq;
183  %mend test;
184  %test
#
the # is in '&amp;qq' and # is #
```

6.1.4. Proc phreg syntax change

```
/* SAS code in SAS V8.2 */
ods trace on;
ods listing close;
proc phreg DATA=data1 ;
    model time*&cens(1) =
            treats paysd
            numrlp2n exdsc1nb/
            ties=exact ;
    ods output
    ParameterEstimates=stat1
    ParameterEstimates=hazard_1
        (keep= ESTIMATE HAZARDRA
            STDERR variable
        where=(upcase(variable)=
            "treats"))
    ParameterEstimates=pcox_1
        (keep=PROBCHIS variable
            rename=(PROBCHIS=P__1)
        where=(upcase(variable)=
            "treats")
        );
run;
ods listing ;
ods trace off;
```

```
/** SAS code in SAS V9.2 */
ods trace on;
ods listing close;
proc phreg DATA=data1 ;
   model time*cens(1)=
            treats paysd
          numrlp2n exdsc1nb/
          ties=exact ;
   ods output
    ParameterEstimates=stat1
    ParameterEstimates=hazard_1
        (keep= ESTIMATE HAZARDRA
        STDERR parameter
        where=(upcase(variable)=
            "treats"))
    ParameterEstimates=pcox_1
        (keep=PROBCHIS parameter
        rename=(PROBCHIS=P__1)
        where=(upcase(variable)=
            "treats")
        );
run;
ods listing ;
ods trace off;
```

6.1.5. ODS output

If used with crosstabfreqs in v9.2 then the output has changed from v8.2 to v9.2.

```
/* SAS code in SAS V8.2 */
proc freq data=data_a.da_ident;
    where itt=1;
    table trtreg1c*dcnrsn5c*sbjdcn1c
        / missing norow nocol;
    table trtreg1c*dcnrsn1c*sbjcmp1c
        / missing norow nocol;
    ods output CrossTabFreqs=freqs1
        CrossTabFreqs2=freqs2;
run;
```

```
/** SAS code in SAS V9.2*/
proc freq data=data_a.da_ident;
    where itt=1;
    table trtreg1c*dcnrsn5c*sbjdcn1c
        / missing norow nocol;
    ods output CrossTabFreqs=freqs1;
run;
```

```
/* second alternative */
proc freq data=data_a.da_ident;
    where itt=1;
    table trtreg1c*dcnrsn1c*sbjcmp1c
        / missing norow nocol;
    ods output CrossTabFreqs=freqs2;
run;
```

6.1.6. ODS statement change for proc lifetest.

Set up some data first

```
data test;
    do i = 1 to 100;
        censor = int (ranuni(1));
        if censor = 0 then
        time = 30* ranuni(1);
        trt = int (2* ranuni(1));
        bygrp = int (3* ranuni(1));
        output;
    end;
    drop i;
run;
proc print data=test (obs=10);
run;
proc sort data=test;
    by bygrp;
run;
```

PROC LIFETEST code in SAS V9 reuses the variable name to name strata variable

```
/* SAS code in SAS V8.2 */
ods output LogHomCov =LogRankVar;
proc lifetest data=test method=km
    /* conftype omitted */;
    time time * censor(1);
    strata trt;
run;
proc print data=logrankvar;
    var rowname _0 _1;
run;
```

```
/** SAS code in SAS V9.2 */
ods output LogrankHomCov =LogRankVar;
proc lifetest data=test method=km
    conftype=LINEAR ;
    time time * censor(1);
    strata trt;
run;
proc print data=logrankvar;
    var trt _0 _1;
run;
```

6.1.7. Proc Genmod change

In 9.2, GENMOD was changed to search directly for the value of the dispersion parameter. In prior releases it searched for the log of the dispersion. In some cases, searching for the log value is more successful and that appears to be the case here. Use the option LOGNB in the MODEL statement to do the search for the log value.

The results in 9.2 match the 8.2 results and avoids the warning.

```
/* SAS code in SAS V8.2 */
proc genmod data=ARLNUM1_20;
  class tgp2_3 cou1a;
  model arlnum1=tgp2_3 cou1a
        numrlp2n exdsc1nb
      / dist=nb link=log
        offset=lnday        ;
  lsmeans tgp2_3 /cl;
  estimate "2-3" tgp2_3 1 -1/exp;
ods output estimates=estout20;
run;
```

```
/** SAS code in SAS V9.2*/
proc genmod data=ARLNUM1_20;
  class tgp2_3 cou1a;
  model arlnum1=tgp2_3 cou1a
        numrlp2n exdsc1nb
      / dist=nb link=log
        offset=lnday lognb;
  lsmeans tgp2_3 /cl;
  estimate "2-3" tgp2_3 1 -1/exp;
ods output estimates=estout20;
run;
```

6.2. SAS behavioural changes

6.2.1. Validvarname in data _null_ is enforced

The validvarname option now will apply to all variables – even in null data steps.

```
/* SAS code in SAS V8.2  */
Options validvarname=V6;
data _null_;
    if 0 then
    set work.ctf_chofile
        nobs=ctf_chofile;
    call symput('ctf_chofile',
        left(put(ctf_chofile, 8.)));
    stop;
run;
```

```
Options validvarname=V6;
/** SAS code in SAS V9.2*/
data _null_;
    if 0 then
    set work.ctf_chofile
        nobs=ctf_chof;
    call symput('ctf_chofile',
        left(put(ctf_chof, 8.)));
    stop;
run;
```

6.2.2. Multiple lengths warning

The warning

```
(WARNING: Multiple lengths were specified for the variable _RTFTEMP by input data set(s).
          This may cause truncation of data.)
```

is new with SAS v9.2. But it is a known SAS issue and does not affect programming. In SAS 9.2 TS2Mo it can be turned off with a new option. This warning changes the

http://support.sas.com/kb/31/850.html

return code from SAS to 4 which could require changes if you are checking such codes on your system.

6.2.3. Lifetest confidence interval

The default transformation for confidence intervals changed from LINEAR to LOG-LOG, it seems like one needs to use the SURVIVAL statement with CONFTYPE=LINEAR to reproduce SAS V8.2 results.

```
/* SAS code in SAS V8.2 */
proc lifetest DATA=example;
    time time*event(0);
    survival OUT=outdat
;
run;
```

```
/** SAS code in SAS V9.2 */
proc lifetest data=example
    time time*event(0);
    outsurv=outdat
conftype=LINEAR;
run;
```

And another example.

```
/* SAS code in SAS V8.2 */
proc lifetest data=indset
        method=km
        timelist=4 6 12 18
            reduceout
        outs=outkmest
;
    time pfs1_1n*pfs1_cs(1);
    strata trtreg1c;
run;
```

```
/** SAS code in SAS V9.2 */
proc lifetest data=indset
        method=km
        timelist=4 6 12 18
            reduceout
        outs=outkmest
        conftype=LINEAR ;
    time pfs1_1n*pfs1_cs(1);
    strata trtreg1c;
run;
```

6.2.4. Proc Lifetest output variable in logcov dataset

Name of the variable that is being created in the logcov dataset of proc lifetest.

```
/* prepare a dataset for the test */
data test;
   do i = 1 to 100;
      censor = int (ranuni(1));
      if censor = 0 then
         time =  30* ranuni(1);
      trt = int (2* ranuni(1));
      bygrp = int (3* ranuni(1));
      output;
   end;
   drop i;
run;
proc print data=test (obs=10);
run;
```

```
proc sort data=test;
    by bygrp;
run;
```

Output variable changed.

```
/* SAS code in SAS V8.2 */
ods output LogHomCov =LogRankVar;
proc lifetest data=test method=km
;
   time time * censor(1);
   strata trt;
run;
proc print data=logrankvar;
  var rowname _0 _1;
run;
```

```
/** SAS code in SAS V9.2 */
ods output LogrankHomCov =LogRankVar;
proc lifetest data=test method=km
    conftype=LINEAR ;
   time time * censor(1);
   strata trt;
run;
proc print data=logrankvar;
  var trt _0 _1;
run;
```

6.2.5. SAS Graph changes

For some examples of plots losing parts of the plot have been reported. See http://www.listserv.uga.edu/cgi-bin/wa?A2=ind0901b&L=sas-l&F=&S=&P=48487 according to this report the changes to default margins on devices can cause this. Another possibility is using OFFSET in a LEGEND statement with absolute units (e.g. cm) rather than percentages.

6.2.6. SAS data format changes

Style cellcontents seems to have disappeared

```
/* SAS code in SAS V8.2 */
STYLE line FROM cellcontents /
```

```
/* SAS code in SAS V9.2  */
STYLE line FROM cell /
```

7. Lessons to learn

Of the many things the GPSII upgrade team did right we obviously did have a number of things that could have been done better. I have bulleted the majority of these below.

- Do not under-estimate the resources needed to perform the task at hand. Only make your resource estimations after thoroughly evaluating what's within the scope of the project.
- Ensure the communication channels between the different departments are set-up and being used effectively and efficiently. This should be driven by the project manager.

- Ensure the users follow all given directions, instructions and take all recommended training, this will avoid causing additional work for the project team. An example would be not following the migration issue management instructions and going direct to a migration champion or business coordinator when the issue had already been answered in the FAQ document.
- Make sure all business areas are consulted for input into timelines, proposed go-live times etc etc. An omission of a key business area would likely lead to re-work and loss of time.
- No matter how thorough your initial plan is, this plan will almost certainly need updating as the project proceeds. Hence, it is vital that the project team add to and improve the initial plan, as and when needed throughout any process. This could be seen from our initial 5 step plan that in the end was a 45 page document.
- No process should ever be expected to 'run like a machine' from the start. You will always encounter unexpected scenarios that will the planned process had not been developed to deal with. Only through repeated testing, dummy runs or actual 'real-life' runs will the developed process become more stable.

Vital things we got right.

- We scheduled using business constraints and took variability (e.g project size) into account
- Schedule in some slack every week for catchup after issues.
- Do not start your migration until you have a very good detailed process developed and tested. A sheet of paper with an idea of what steps to do is *not* sufficient.
- Do not be afraid to radically change your process if there are time issues, question everything you do and if it is really needed, look for parts that can be done in parallel. Look for differences in your MUs that can be exploited (e.g. inactive projects)
- Do dry runs until your times match your projections
- Build an accurate calculation of migration time taking into account project size,
- Accurate data on how long a migration unit takes to migrate are more important than keeping a commitment you made 6 months before testing began. Even stones are not written in stone (they can be sanded).
- Automate your process as much as you can with *ad hoc* scripting tools for fast coding, easy redesign, monitoring and notifications

- It is easier when you have the datasets in a VC system. Is it worth putting them in one just for this project? Probably.
- Throughly document your process and all its elements (scripts, files,...)

8. References

SAS migration focus area http://support.sas.com/rnd/migration/ and documentation of PROC MIGRATE.
http://support.sas.com/documentation/cdl/en/proc/63079/HTML/default/viewer.htm#p1kv04orx2cy03n1urntor37gzkz.htm and validation issues in the Pharmaceutical industry are covered here: http://support.sas.com/rnd/migration/planning/validation/. A full list of SAS filetypes to look for when scanning a file system is here: http://support.sas.com/rnd/migration/procmigrate/execution/special_findingnotmvs.html
There is an introduction to precision issues in http://support.sas.com/techsup/technote/ts654.pdfand http://support.sas.com/techsup/technote/ts230.html. Another useful account is Montgomery (2008) http://www.lexjansen.com/phuse/2008/cs/cs08.pdf

Contact information

We would value your comments and questions on this paper.
Please contact either of the authors at:

David J. Garbutt, Principal Consultant,
BIOP AG,
Centralbahnstrasse 9,
CH – 4051 Basel
Switzerland
Work Phone: +41 61 227451
Fax: +41 44 390 3722
david.garbutt@biop.ch
LinkedIn profile

Nigel Montgomery, Principal Statistical Programmer, Novartis Pharma,
Lichtstrasse 35,
CH – 4056 Basel
Switzerland
Work Phone: +41 61 3246180
Fax: +41 61 3243039
Nigel.montgomery@Novartis.com
LinkedIn Profile

Acknowledgements

DG I would like to thank my family who were very tolerant for the year I migrated away seemingly every night.

NM I would like to thank my wife for being so tolerant during the GPS II upgrade project. I'd also like to thank my fellow Business Coordinators Morgane Emeury and Mako Araga for the great teamwork and spirit shown throughout the project, without this it would have been a lot more difficult.

Both of us would like to thank our colleagues on this project that is Norbert Spengler and the GPS team Hermann Klee , Jean-Yves Robo and Ulrich Knappich. Special thanks are also due to Raymond Stritt and Alan Solomon of IMS Health who were our project managers and helped created a realistic process and then created and shepherded to completion a viable schedule and project plan. The list would be incomplete without mentioning our colleagues who joined us in June 2009 to create a well-oiled migration machine: Sachin Shinde (Cognizant) and Venkatraman Naganathan (Novartis).

A. Appendix

A.1. Maximal process automation with scripts

The scripts written were used in several stages and all version controlled and validated.

Fifteen scripts were written and documented in the first two months of the project. They formed the basis of the process and provided lots of file management functions as well as breaking replication relationships, capturing group memberships, composing emails to users with individualised content.

Names: email_co.sh, email_err.sh, email_mig_fb.sh, email_mig_ok.sh, email_users.sh, migr2sas.sh, migrvob.sh, mkgrouplist.sh, mkls.sh, mkr2vob.sh, mkuserlist.sh, mkwo.sh, modesim_rel.sh, scanNrep.sh, scansum.sh, scanfor.sh.

They shared information written to text files such as list of Program programmers and Program Statisticians, VOBs in the bundle, users in groups and their email addresses.

A.1.1. What type of file is this?

Unix has a utility for checking the type of file by looking at the content, the file command. It uses a special file (called, informally, the magic file) with diagnostic rules. A magic file with SAS support is here `http://ftimes.sourceforge.net/Files/XMagic/magic`. The typeit (Korn shell 93) script checks what kind of a SAS file a file is by looking inside rather than just looking at the filetype.

```
#! /usr/bin/ksh93
# type a sas binary file
```

```
#
# DJ Garbutt - 11 Feb 2009
# nb run with korn shell 93 to get indexing
# get first 2048 bytes
dd bs=1024 obs=1024 count=2 of=typeit_temp_$$ cbs=1 if=$1 >/dev/null 2>&1
# convert nulls to spaces
tr '\0' ' ' <typeit_temp_$$ | tr '[:cntrl:]' ' ' | tr '\012' ' ' >typeit_temp
    _$$_out
# read into variable - turn off record separators then restore them
oldIFS=$IFS
IFS=''
contents=$(< typeit_temp_$$_out)
IFS=$oldIFS
realname=$(basename $1)
oldapp=${contents:0:3}
# SAS v6 ds
if [[ $oldapp == SAS ]]
then
   print " SAS 6 ? file says [$(file -m ~bin/SAS.magic $1|awk -F':' '{print $
       2}')]"
else
   app=${contents:84:3}
   alsotype=${contents:88:4}
   name=${contents:92:32}
   filetype=${contents:156:8}
   version=${contents:216:16}
   if [[ $version == 9* ]]
   then
   platform=${contents:224:60}
   else
   platform=${contents:228:60}
   fi
   # put in a variable & print to remove trailing spaces from variables
   message="$realname or $name is a $app $alsotype $filetype (${version%%[A-
       Za-z_]*}, $platform). file says [$(file -m ~/bin/SAS.magic $1|awk -F
       ':' '{print $2}')] "
   print $message
fi
# tidy up temp files
rm -f typeit_temp_*
```

Sample results (the files data01 - data09 are not really SAS datasets but text files named with a type of .sas7bdat).

```
> apply 'ksh93 typeit.sh %1 ' /vob/PROJTEST/PROJTEST-STUDY14/data/analysis/data*.sas7b@(dat|cat|vew)
data01.sas7bdat or is a \(, :\). file says [ ascii text]
data02.sas7bdat or is a \(, :\). file says [ ascii text]
data03.sas7bdat or is a \(, :\). file says [ ascii text]
data04.sas7bdat or is a \(, :\). file says [ ascii text]
data05.sas7bdat or is a \(, :\). file says [ ascii text]
data06.sas7bdat or is a \(, :\). file says [ ascii text]
data07.sas7bdat or is a \(, :\). file says [ ascii text]
data08.sas7bdat or is a \(, :\). file says [ ascii text]
data09.sas7bdat or is a \(, :\). file says [ ascii text]
data30.sas7bdat or DATA30 is a SAS FILE DATA \( 8.02, RS60:02M0AIX \). file says [ SAS 7+ data file]
```

```
data31.sas7bdat or DATA31 is a SAS FILE DATA \( 8.02, RS60:02M0AIX \). file says [ SAS 7+ data
data32.sas7bdat or DATA32 is a SAS FILE DATA \( 8.02, RS60:02M0AIX \). file says [ SAS 7+ data
data33.sas7bdat or DATA33 is a SAS FILE DATA \( 8.02, RS60:02M0AIX \). file says [ SAS 7+ data
data34.sas7bdat or DATA34 is a SAS FILE DATA \( 8.02, RS60:02M0AIX \). file says [ SAS 7+ data
data35.sas7bdat or DATA35 is a SAS FILE DATA \( 8.02, RS60:02M0AIX \). file says [ SAS 7+ data
data36.sas7bdat or DATA36 is a SAS FILE DATA \( 8.02, RS60:02M0AIX \). file says [ SAS 7+ data
data37.sas7bdat or DATA37 is a SAS FILE DATA \( 8.02, RS60:02M0AIX \). file says [ SAS 7+ data
data38.sas7bdat or DATA38 is a SAS FILE DATA \( 8.02, RS60:02M0AIX \). file says [ SAS 7+ data
data39.sas7bdat or DATA39 is a SAS FILE DATA \( 8.02, RS60:02M0AIX \). file says [ SAS 7+ data
```

This is a useful approach because for transport files the filetype is not fixed by SAS but in a file statement meaning anything can be used. Also people sometimes rename a file by changing the filetype to `.sas7bdatold` and other tricks.

Other scripts possible: email, list checkouts (create CSV and email), output comparator?

A.1.2. Send list of checkouts by email to users with checkouts

The migration would fail with checked out datasets and versions not checked in are also not synchronised across hosts so they would be lost on the new servers. we therefore scanned for checkouts and emailed the users accordingly. This script shows how that was done.

```
#! /usr/bin/ksh
# SAS migration checkouts scanner,emailer
# DJ Garbutt - 9 Mar2009
#
#
dir=$(basename $PWD)
for files in *.txt
do
   name=${files%.*}
   set -A vobs $(cat $files | tr '\t\r' ' ' | awk '$1 ~ /\+/ {print $2} ' )
 # regenerate list of checkouts
   print "\nListing checkouts for vobs in $files ( ${vobs[*]} )"
   time apply '/home/tec_cla1/bin/lsco %1 ' ${vobs[*]}
   # add them all together to one file for emailing
   touch ./${name}_co_$$.csv
   for vb in ${vobs[*]}
    do
      touch ${vb}_co.csv ${vb}_tec_co.csv
      cat ${vb}_co.csv >> ./${name}_co_$$.csv
      cat ${vb}_tec_co.csv >> ./${name}_tec_co_$$.csv
    done
    sort -t';' -f -k 2 -k 3 -k 1nr -o ${name}_co_$$.csv ${name}_co_$$.csv
    sort -t';' -f -k 2 -k 3 -k 1nr -o ${name}_tec_co_$$.csv ${name}_tec_co_$$.csv
# mail users scan is done.
  # get list of user ids and hence email addresses to send it out
   apply 'lsuser -a gecos %1' $(gawk -F'|;' '{ print tolower($2)}' ./${name}_co_$$.csv | sort -u ) |
        gawk -F'[,;]' '/mailto/ {sub(/mailto/,"",$2);print $2 }' >| ./${name}_checkouts.email

#Construct a multipart mime message.
# First encode file with uuencode to get icon in lotus notes
/usr/bin/uuencode ./${name}_co_$$.csv ${name}_checkouts.csv >|${name}_checkouts.csv
# Construct message: latest AIX mail inserts mime header & footer above/below text
if [[ -s ${name}_checkouts.csv ]]
then
/bin/mail -s "List of checkouts in the VOBs of $name" -c "$( < ../mtdi.email) $( < ../mtcis.email)" \
   $(cat ${name}_checkouts.email ppps.email mc.email | sort -u ) \
   <<EOF

Dear CIS migration co-ordinator,

  please find attached a CSV file with all Checked out files found in a pre-migration scan of the $name VOB.


  The log file should open directly as a CSV file from within Lotus Notes.

  To convert the file to xls - follow these steps:
    1) select all of column A
```

```
    2) from data menu select 'text to columns'
    3) in the wizard that pops up select
    4) delimited format with
    5) vertical bar "|" (or ;) as the separator.
    6) You may need to select all columns and set data| filter | autofilter
    6) save the file as xls wherever is convenient for you

  The CSV file is sorted by age (of checkout) and user.
  The header for his file is :
  Age(days);User;view; File path ;Version;Status;Labels

We also have lists for individual VOBs, please ask if you want any of them.

best regards, your SAS Migration team,

Norbert Spengler and Dave Garbutt
GPSII SAS Migration Team

PS This file includes the following VOBs with checkouts: (number, name)

$(wc -l ${name}*_co.csv | grep -v 'total' | gawk ' $1 > 0 {sub(/_co.csv/,"",$2);printf "%5i \t%s\n", $1,$2}' )


$(< ${name}_checkouts.csv)

EOF
else
/bin/mail -s "No checkouts currently in the VOBs of $name" -c "$( < ../mtdi.email) $( < ../mtcis.email)" \
   <<EOF

Dear CIS migration co-ordinator,

  we have looked for checkout in the vobs:
  $vobs

  and found none.
  Please contact us if this contradicts your knowledge of the current state of these VOBs.

best regards, your SAS Migration team,

Norbert Spengler and Dave Garbutt
GPSII SAS Migration Team


EOF

fi

# Now email details of checkouts for tec users (files are created in lsco
/usr/bin/uuencode ./${name}_tec_co_$$.csv ${name}_tec_checkouts.csv >|${name}_tec_checkouts.csv
#print "$? from uuencode"
#ls -l *_co_$$* *checkouts.*

# Construct message: latest AIX mail inserts mime header & footer above/below text
#
/bin/mail -s "List of checkouts for tec_cla1 (etc.) in the VOBs of $name" \
   $( < ../mtdi.email) <<EOF

Dear DI migration co-ordinator,

  please find attached a CSV file with all Checked out files for Admin and Support team accounts found in a pre-
       migration scan of the $name VOBs.
  These files need to be checked in before the migration can proceed.

  The log file should open directly as a CSV file from within Lotus Notes.

  To convert the file to xls - follow these steps:
    1) select all of column A
    2) from data menu select 'text to columns'
    3) in the wizard that pops up select
    4) delimited format with
    5) vertical bar "|" (or ;) as the separator.
    6) You may need to select all columns and set data| filter | autofilter
    6) save the file as xls wherever is convenient for you

  The CSV file is sorted by age (of checkout) and user.
  The header for his file is :
  Age(days);User;view; File path ;Version;Status;Labels


best regards, your SAS Migration team,

Norbert Spengler and Dave Garbutt
GPSII SAS Migration Team
```

```
PS This file includes the following VOBs with checkouts: (number, name)

$(wc -l ${name}*_tec_co.csv | grep -v 'total' | gawk ' $1 > 0 {sub(/_tec_co.csv/," tec users",$2);sub(/_co.csv/,"
     normal users",$2);printf "%5i \t%s\n", $1,$2}' )



$(< ${name}_tec_checkouts.csv)

EOF


done
# tidy up
  mv -f ./${name}_checkouts.csv ./${name}_checkouts.mime
  mv -f ./${name}_co_$$.csv ./${name}_checkouts.csv
  mv -f ./${name}_tec_checkouts.csv ./${name}_tec_checkouts.mime
  mv -f ./${name}_tec_co_$$.csv ./${name}_tec_checkouts.csv

  # print summary
print "\nTotal number of checkouts found: "
wc -l ./${name}*_checkouts.csv
print "\t No. of users emailed: "
wc -l ./${name}*_checkouts.email
```

The script lsco is called from the above script to create a CSV file listing all checkouts.

```
# /usr/bin/ksh
# prepare a list of current checkouts for VOBs matching $1
# split file by user name - tec_cla1 & support to one file - others to normal
# revised to supply view context if there is none
view=${CLEARCASE_ROOT:-/view/tec_cla1_view}
cleartool lsco -all -areplicas -fmt "%Ad;%u;%Tf;%En;%f;%Vn;%Nl\n" ${view}/vob
    /${1}/ |
   sort -t';' -f -k 2 -k 3 -k 1nr |
   gawk -F'[|;]' -v filen="./${1}_co.csv" -v filetec="./${1}_tec_co.csv" '
      BEGIN {OFS=";" ; $2 = tolower($2) }
       {
          $2=tolower($2);
          if ( $2 ~ /tec_cla1|roboje1|knappul2|kleehe1|ctreview|spengno2|
              garbuda|tec_clu1|chbss/ )
             file=filetec;
          else
             file=filen;
          print $0 > file
          }'
```

The awk fragment

```
BEGIN {OFS=";" ; $2 = tolower($2) }
 { $2=tolower($2);
    if ( $2 ~ /tec_cal1|roboje1|knappul2|kleehe1|ctreview|spengno2|garbuda|
        tec_cul1|chsuer/ )
    file=filetec;
    else
    file=filen;
print $0 > file
 }
```

A.1.3. Checking permissions

We needed to confirm they were exactly as before to ensure cases like special groups and different protections for an individual directory were not inadvertently missed. A copy of the original permissions in place was captured at the time of replication and moved to the new server. Then this script was run after the permissions had been restored.

The script uses several useful tricks: HERE documents substitute parameters present including command pipes, and the `join` command can make a matched list from two lists of directories with their permissions. This takes the hard work out of comparing long lists of directories.

```
join  -v 1 -v 2  -o 1.2 2.2 0  -e '--?--'  before.lst after.lst
```

This command prints a listing with three columns: the old group assigned, the new group and the directory referred to. (groups beginning with Z are special groups protecting areas with unblinded information).

```
NSO NSO CFTY720D/CFTY720D2309_DMC
NSO NSO CFTY720D/pool/pool_001
NSO NSO CFTY720D/pool/pool_002
NSO NSO CFTY720D/pool/pool_003
ZFTY720D ZFTY720D CFTY720D/pool/pool_004
ZFTY720D ZFTY720D CFTY720D/pool/pool_005
```

If the first two columns are not equal then the permissions are not the same and some action is needed. This tool acts as a quality check for the people setting the permissions *and* as a document to be signed off for the users as a final release. The procedure was to leave the top level directory set to a group with no normal users, run this script which emails a copy to the users recorded as responsible, then when a PDF of the file is returned signed (digitally) change the top level directory to the normal group (in this case it would be NSO). The first section also serves as a customised instruction list for setting the permissions.

Other versions of the `join` command can be used to get a list of all cases where the first and second column are not equal to highlight in the email.

```
#! /usr/bin/ksh
# SAS migration scanner,emailer
# DJ Garbutt - 9 Mar2009
#
#add get email address for PP to add to email
# July 3-5 2009 - DJG. (incortp cahnges made after FTY feedback
# added join of the full.lst files to make comparison easier.
# prints checks to terminal, highlighting if permissions are Ok or not.
# outputs a test .sh file as experiment
# revised version to use info from new gps_projlist_vob.sh
# fewer fields per line and no files from servers without the VOB
# 9-Aug-09 DJG Added tests at end - eg grep check \for CAdmPHCA,
# also it is adapted for revised gps_projlist_vob.sh which now lists vob level as well.
dir=$(basename $PWD)
# define list of normal vobs for use in REs --- may need updating if a new TA group is created
normgroups='/SANDOZ|CVM|RESP|MPSPEC|NSO|ONC|IID|legacy|CPV|CUsersPH|CPV_US|CPV_CH/'
```

```
typeset -r PRG=$(basename $0)
USAGE="usage: $PRG [[-d] | [-a] [-t] ]
     Email final IQ to PP etc and calc current permissions on old and new for comparison
        optional flag -d (dummy run) means results only emailed to Norbert & Dave
        -a (ctual) means mail to PP /PS & MC
        -t (est) means only emailed to Dave
       Use a dummy run before setting permissions on the vobs to check, and
       again afterwards to check all is done before sending to PP/PS.
"
# actually default is "dummy = true - ie send only to Norbert & dave
dummy=true
while getopts ":dat" arguments
do
  case $arguments in
     d) dummy=true ;;
     a) dummy=false ;;
     t) dummy=true
           test=true ;;
     \?) print $USAGE
        return 1
        ;;
      *) print "$OPTARG is not a valid switch"
        print $USAGE
        return 1
        ;;
  esac
done
shift $(( $OPTIND - 1 ))

for files in *.txt
do
  # .... .........................for every file ending in .txt in the current directory
  name=${files%.*}
  # get a Vbar sep voblist
  voblist="$(print $(gawk -F'[ \r\t]' ' $1 ~ /\+/ && $2 !~ /\// {vob[$2]=$2} END {OFS="|"; for (g in vob ) print g
       } ' *.txt ) | tr ' ' '|' )"

      # mail users IQ details
      if [[ $(< $files) != *\+* ]]
      then
        continue
        # no VOBs marked with a + in this txt file
      fi
      # 3) confirm the permissions using the gps_projlist_vob.sh script for each VOB.
      # execute Uli's script to get current permissions of servers for every VOB marked with a + in the current txt
            file
      # nb ignore the cleartool errors from servers where the VOB does not exist (one old server or the other)
       gawk -F'[ \r\t]' ' $1 ~ /\+/ && $2 !~ /\//{
              print "----------------- Getting protections for "$2" on all systems";
               system("/home/tec_cla1/bin/gps_projlist_vob.sh " $2 );
              print "\n"}' $files
 # now process all the VOBs in this txt file and make an email for each one
      if [[ -s ${dir}-all_grps.lst ]]
      then
         rm -f ${dir}-all_grps.lst
      fi
      if [[ -s ${dir}-special_grps.lst ]]
      then
         rm -f ${dir}-special_grps.lst
      fi
      if [[ -s ${dir}-normal_grps.lst ]]
      then
         rm -f ${dir}-normal_grps.lst
      fi
      if [[ -s ${dir}-grpstudy.lst ]]
      then
         rm -f ${dir}-grpstudy.lst
      fi
      #wc -l *-special_grps.lst
      # new - no explicit loop on vob - use wild card on full.lst file names instead - only need to look at old
           server
       gawk -F'^' -v normgrp=$normgroups -v dir=$dir \
             -v text=$voblist '\
            length($2) > 0 {print $1"|"$2 >> dir"-grpstudy.lst" };
            length($2) > 0 {if ( $2 ~ normgrp ) grp[$2]=$2 ; else spgrp[$2]=$2
               }
              END { nng=0 ;
                   nsg=0 ;
                   for ( item in grp ) {nng++ ; print item >> dir"-normal_grps.lst" } \
                   for ( item in spgrp ) {nsg++ ; print item >> dir"-special_grps.lst"}
                   print "\nFor \n" text " vobs\n there are " nng " normal groups and " nsg " special groups\n"
                  }'\
```

```
       *_full.lst_*@(21|51)*
wc -l *_grps.lst
wc -l *-grpstudy.lst
wc -l *_full.lst_*@(21|51)*
cat *-special_grps.lst


        # append the full.lst files created by the above script $$_old_full.lst
        cat *_full.lst_@(ichn21.eu|usehux51.na).novartis.net >>$$_old_full.lst
        cat *_full.lst_chbsux0113.eu.novartis.net >>$$_new_full.lst
        # create upto date lists of groups/ special groups from full.lst for email
        # the orig lists are on old servers and anyway could be out of date
        # This new way to do it is fail safe - any groups incorrectly failing the programmed test for = normal
            group
        # are still included as special (and therefore seen)
       #/view/trigger_view/vob/util/wdcs/clt4wdcs/data/full.lst


      # now take accumulated lists across all target vobs and reduce to only unique values
      cat ${dir}-special_grps.lst | tr -d ',' | sort -u -o ${dir}-special_grps.lst
      sort -u -o ${dir}-normal_grps.lst ${dir}-normal_grps.lst
      # concatenate the the special and normal groups together
      cat ${dir}-special_grps.lst ${dir}-normal_grps.lst >| ${dir}-all_grps.lst
      # Construct message: latest AIX mail inserts mime header & footer above/below text
      # to- $(< ppps.email)
      if [[ $dummy == true ]]
      then
        if [[ $test == true ]]
        then
        mailinglist="david.garbutt@novartis.com"
        else
            mailinglist="$( < ../mtdi.email) "
            fi
      else
      # mailinglist='-c "$( < ../mtdi.email) $( < ../mtcis.email) $(< mc.email) $(< ppps.email) "'
       mailinglist=$( < ../mtdi.email)
      fi
# concatenate the files from each vob together, do not forget to sort on fields to get a good join later
cat *_full.lst_@(ichn21.eu|usehux51.na).novartis.net | sort -t^ -k1,1 -k2,2 | tr '^' ' ' >|before-$$.lst
cat *_full.lst_chbsux0113.eu.novartis.net | sort -t^ -k1,1 -k2,2 | tr '^' ' ' >|after-$$.lst
# head *-$$.lst

/bin/mail -s "$dir_Project_perm GPS II Migration - Project Permissions for project $name ($dir) " \
   $mailinglist << EOF

Project Permissions Verification for Project $name ($dir) Created : $(date +'%Y-%m-%d')

Dear Project Programmer(s) and Program Statistician(s),


The migration process has been completed and the permissions changed for the vobs listed below with a + sign:

$(cat $files)

The following normal groups were found to be in use for this VOB set:

$(cat ${dir}-normal_grps.lst | tr -d ',' | sort -u | tr '\n' ',' | sed 's/,[ ]*$//')

Including the following special groups:

$(cat ${dir}-special_grps.lst | tr -d ',' | sort -u | tr '\n' ',' | sed 's/,[ ]*$//')

 Where there is more than 1 special group (i.e. non-C group or TA group) in use for this VOB set, review and
       approval by the PP/PS
 is also required (section 4).

I have executed the instructions stated in the GPS II Data Migration Work Instruction (GPSII_SP_WKI001) and applied
       all appropriate group
assignments.


|_____ (DI Signature)

As part of the migration process DI have applied the special groups to the directories (studies or pool areas) as
      they were
configured on the GPSII Release 1 server.

The default group for the vobs( studies and pool areas, util, etc.) will be the appropriate TA group, or B-group if
       the VOB is restricted.
the following steps have been executed successfully:

1) Added the TA and special groups to the VOB group list.

cleartool protectvob -add <ta_group and/or specialGroup(s)> -del <all_old_groups> <vob-storage-pathname>
```

```
2) Set the default protections for other top level directories for each target VOB

Any WDCS directories protected with special permissions will be listed here:

$(for vobs in $voblist ; do if [[ -r /vob/$vob/gps_migr/wdcs/${vob} ]]
    then print "$( ls -ld /vob/${vob}/gps_migr/wdcs/${vob}* 2>/dev/null |
  gawk -v normgrp=$normgroups 'BEGIN {OFS=", " } $2 !~ normgrp && ! ( $2 ~ /^B/ && $1 ~ /restricted/ ) { print "
        cleartool protect -chg",$4" -recurse -chmod 770 " $9} ')"
  fi
  done)


3) Set the protection for special group directories in each VOB where needed.
  (number is line number in ${dir}-grpstudy.lst )

$(gawk -v normgrp=$normgroups -v dir=${dir} ' length($2) > 0 && $2 !~ normgrp && ! ( $2 ~ /^B/ && $1 ~ /restricted/
      ) {print "("NR") cleartool protect -chg "$2 " -chmod 770 -recurse /vob/"$1 } \
   {print " cleartool protect -chg "$2 " -chmod 770 -recurse /vob/"$1 > dir"-perm.sh" }' before-$$.lst)

4) The status protection of studies on the Rel 1 and Rel 2 servers is as follows:

Permissions before and after the migration to GPS II Rel 2 ( old group, new group, vob/study or pool/pkpd area) :

$(join -a 1 -a 2 -1 1 -2 1 -o 1.2 2.2 0 -e '--?--' before-$$.lst after-$$.lst | gawk '{ printf "%-9s%-10s%s\n", $1,
      $2, $3 }' )

Note that this list will now contain the permissions at the vob level as well as each study/pool/wdcs directory.
      The VOB level permissions
should be different from the old server until the VOBs are released to users.
If only these directories belong to CAdmPHCA then this document can still be signed.

Any studies only on Rel. 1, or only on Rel. 2 will be listed here (missing permissions are marked by --?--)

$(join -v 1 -v 2 -o 1.2 2.2 0 -e '--?--' before-$$.lst after-$$.lst | gawk '{ printf "%-9s%-10s%s\n", $1, $2, $3 }'
      )


PP/PS :
   Please compare the lines above and verify that each study or pool on GPSII release 2 is protected by the same
        group as on the GPSII release 1 server.

All Special groups allocated as expected. I authorise VOB level access to be applied to enable authorised users to
     access the data.

|
__ (signature)


                                                    _____________ (date)

If some special groups are not assigned correctly contact us immediately.

Thank you for your prompt response to this email.

If there is more than one PP/PS then only one needs to review and sign this confirmation.

Your DI Migration team

Dave Garbutt and Norbert Spengler


EOF

done


difs=$( join -v 1 -v 2 -o 1.2 2.2 0 -e '--?--' before-$$.lst after-$$.lst | gawk '$2 !~ /CAdmPHCA/ { printf "%-9s
     %-10s%s\n", $1, $2, $3 }' )
[[ -s $difs ]] && print "\n$PRG\t Mismatches in study lists!! Fix and Do not send.
$difs
"
 difs2=$(join -a 1 -a 2 -1 1 -2 1 -o 1.2 2.2 0 -e '--?--' before-$$.lst after-$$.lst | gawk ' $2 != $1 && $2 !~ /
      CAdmPHCA/ { printf "%-9s%-10s%s\n", $1, $2, $3 }')
[[ -s $difs2 ]] && print "\n$PRG\t Mismatches in permissions!! Fix and Do not send, unless it is only VOB level
     directories.
$difs2
"
# with RE to get dirs with CAdmPHCA or CUsersPH on study level or below (ie they have ^string/string/[^] before
     CAdmPHCA|CUsersPH )
grep -q -E '^.+\/[A-Z0-9_a-z]+[/^]+CAdmPHCA|^.+\/[A-Z0-9_a-z]+[/^]+CUsersPH' after-$$.lst
(( $? == 0 )) && print "\n$PRG\t Permissions on 113 still contain CAdmPHCA or CUsersPH. Fix and do not send."

# tidy up temp files
```

```
rm -f $$_old_full.lst $$_new_full.lst before-$$.lst after-$$.lst mangle-$$.awk
print -- '---------------------------' "sent email for project $name in $dir"
```

Here is a sample output file (some line breaking has been preserved by adding a full stop at the start of a line).

```
T:\SAS_Migr\CABC123_CH\email_message.lst Mittwoch, 3. Juni 2009 14:58
Project Permissions Verification for Project (CABC123_CH) Created : 2009-06-03
Dear Project Programmer(s) and Program Statistician(s),
The migration process has been completed and the permissions changed for the vobs listed below with a + sign:
+ CABC123
+ CABC123D
+ CABC123D_restricted
# when looping over txt files

.
The following normal groups were found to be in use for this VOB set:

.
NSO

.
Including the following special groups:
BABC123D,ZABC123D
Where there is more than 1 special group (i.e. non-C group or TA group) in use for this VOB set, review and approval by the PP/PS is also requ
|
|
|
|
|
|
|_____________________________________________________ (DI Signature)

.
As part of the migration process DI have applied the special groups to the directories (studies or pool areas) as they were configured on the

.
1) Added the TA and special groups to the VOB group list.
cleartool protectvob -add <ta_group and/or specialGroup(s)> -del <all_old_groups> <vob-storage-pathname>
2) Set the default protections for other top level directories for each target VOB
Vob: 1 CABC123
cleartool protect -chg $def_group -recurse /vob/CABC123/util
cleartool protect -chg $def_group -recurse /vob/CABC123/pool
cleartool protect -chg $def_group -recurse /vob/CABC123/gps_migr
cleartool protect -chg $def_group -recurse /vob/CABC123/lost+found
cleartool protect -chg $def_group /vob/CABC123/*

.
Vob: 2 CABC123D
cleartool protect -chg $def_group -recurse /vob/CABC123D/util
cleartool protect -chg $def_group -recurse /vob/CABC123D/pool
cleartool protect -chg $def_group -recurse /vob/CABC123D/gps_migr
cleartool protect -chg $def_group -recurse /vob/CABC123D/lost+found
cleartool protect -chg $def_group /vob/CABC123D/*
Vob: 3 CABC123D_restricted
cleartool protect -chg $def_group -recurse /vob/CABC123D_restricted/util
cleartool protect -chg $def_group -recurse /vob/CABC123D_restricted/pool
cleartool protect -chg $def_group -recurse /vob/CABC123D_restricted/gps_migr
cleartool protect -chg $def_group -recurse /vob/CABC123D_restricted/lost+found
cleartool protect -chg $def_group /vob/CABC123D_restricted/*

.
Any WDCS directories protected with special permissions will be listed here:

.
3) Set the protection for special group directories in each VOB where needed.
(number is line number in CABC123_CH-grpstudy.lst )
(30) cleartool protect -chg ZABC123D -recurse /vob/CABC123D/pool/pool_004
(31) cleartool protect -chg ZABC123D -recurse /vob/CABC123D/pool/pool_005
(33) cleartool protect -chg ZABC123D -recurse /vob/CABC123D/pool/pool_007

.
4) The status protection of studies on the Rel 1 and Rel 2 servers is as follows:

.
Permissions before and after the migration to GPS II Rel 2
( old group, new group, vob/study or pool/pkpd area) :
NSO NSO CABC123D/CABC123D1201E1
NSO NSO CABC123D/CABC123D1201_DMC
NSO NSO CABC123D/CABC123D2101
NSO NSO CABC123D/CABC123D2102
NSO NSO CABC123D/CABC123D2105
```

```
NSO NSO CABC123D/CABC123D2106
NSO NSO CABC123D/CABC123D2107
NSO NSO CABC123D/CABC123D2108
NSO NSO CABC123D/CABC123D2109
NSO NSO CABC123D/CABC123D2110
NSO NSO CABC123D/CABC123D2113
NSO NSO CABC123D/CABC123D2201E1
NSO NSO CABC123D/CABC123D2201E1_DMC
NSO NSO CABC123D/CABC123D2301E1
NSO NSO CABC123D/CABC123D2301_DMC
NSO NSO CABC123D/CABC123D2302E1
NSO NSO CABC123D/CABC123D2302_DMC
NSO NSO CABC123D/CABC123D2306
NSO NSO CABC123D/CABC123D2306_DMC
NSO NSO CABC123D/CABC123D2309E1
NSO NSO CABC123D/CABC123D2309_DMC
NSO NSO CABC123D/pool/pool_001
NSO NSO CABC123D/pool/pool_002
NSO NSO CABC123D/pool/pool_003
ZABC123D ZABC123D CABC123D/pool/pool_004
ZABC123D ZABC123D CABC123D/pool/pool_005
NSO NSO CABC123D/pool/pool_006
ZABC123D ZABC123D CABC123D/pool/pool_007
NSO NSO CABC123D/pool/pool_008
NSO NSO CABC123D/pool/pool_009
BABC123D BABC123D CABC123D_restricted/CABC123D1201E1
BABC123D BABC123D CABC123D_restricted/CABC123D2107
BABC123D BABC123D CABC123D_restricted/CABC123D2110
BABC123D BABC123D CABC123D_restricted/CABC123D2113
BABC123D BABC123D CABC123D_restricted/CABC123D2201
BABC123D BABC123D CABC123D_restricted/CABC123D2301E1
BABC123D BABC123D CABC123D_restricted/CABC123D2302E1
BABC123D BABC123D CABC123D_restricted/CABC123D2306E1
BABC123D BABC123D CABC123D_restricted/CABC123D2309E1
BABC123D BABC123D CABC123D_restricted/pool/pool_001
BABC123D BABC123D CABC123D_restricted/pool/pool_002
BABC123D BABC123D CABC123D_restricted/pool/pool_003
BABC123D BABC123D CABC123D_restricted/pool/pool_004
BABC123D BABC123D CABC123D_restricted/pool/pool_005
BABC123D BABC123D CABC123D_restricted/pool/pool_006
BABC123D BABC123D CABC123D_restricted/pool/pool_007
BABC123D BABC123D CABC123D_restricted/pool/pool_008
BABC123D BABC123D CABC123D_restricted/pool/pool_009

.
Any studies only on Rel. 1, or only on Rel. 2 will be listed here (missing permissions are marked by --?--)
PP/PS : Please compare the lines above and verify that each study or pool on GPSII release 2 is protected by
the same group as on the GPSII release 1 server.

.
All Special groups allocated as expected. I authorise VOB level access to be applied to enable authorised users to access the
|
|
|
|
|
|
|_____________________________________________________________ (signature)
_____________________ (date)
If some special groups are not assigned correctly contact us immediately.
Thank you for your prompt response to this email.
If there is more than one PP/PS then only one needs to review and sign this confirmation.
Your DI Migration team

Dave Garbutt and Norbert Spengler
```