

Solving the Source Control Issue

David Shannon, Amadeus Software, Oxford, UK

ABSTRACT

The administration of SAS code frequently depends on the file management and good practice of the individual developer. When many people share the same SAS programs it is likely a conflict or unintentional loss of code will occur. Source Control systems are designed to manage access, versions and the history of files. This paper presents a review of source control systems that are already available, or can be used, within the SAS environment.

The discussion begins by understanding what is meant by the terms Source Control and Version Control. Methods of working with source control are discussed. Finally the existing capabilities of Data Integration Studio and Drug Development solutions are presented.

This paper on "Solving the Source Control Issue" is appropriate for those who program SAS, maintain SAS code or are responsible for the long term security and availability of SAS programs. An understanding of the SAS programming language is not required.

INTRODUCTION

Source Control is a subject area quite familiar to software developers where the need to support many developers concurrently is a prerequisite. For the majority of SAS programmers this process is either separate from the programming environment or not performed at all.

The need to maintain professional source control is increasingly recognised by organisations that fall under regulatory jurisdiction. Typical examples are the financial industry and the development and production of medicines and devices in the life sciences industries. Furthermore, organisations following ISO 9000 standards will require source control.

Source control in SAS programming environments is usually managed through separate tools or functions from the SAS Display Manager or Enterprise Guide. This paper will discuss those options and explore how source control can be integrated within Enterprise Guide, or is integrated within existing enterprise level solutions from SAS Institute.

WHAT IS SOURCE CONTROL?

It is many things to many people.

Various terms are used such as *Version Control*, *Change Control*, *Source Control*, *Source Code Management* and *Revision Control* amongst others. This inconsistency in terminology leads to some confusion about what function is provided. It would not be unreasonable to comment that software vendors tailor these terms to suit their own products functionality.

The many answers and variety of definitions suggests a lack of cohesive education and definition on the subject. Ask yourself where you learnt source control (or any of the terms referred to above) and it is quite likely the answer will be in the workplace. Ask how you learnt it, and the answer will be most frequently be through a skills sharing process with colleagues.

To add structure and clarity to these terms, the following cumulative use of the terms Revision Control, Version Control, Source Control and Document Management is proposed:

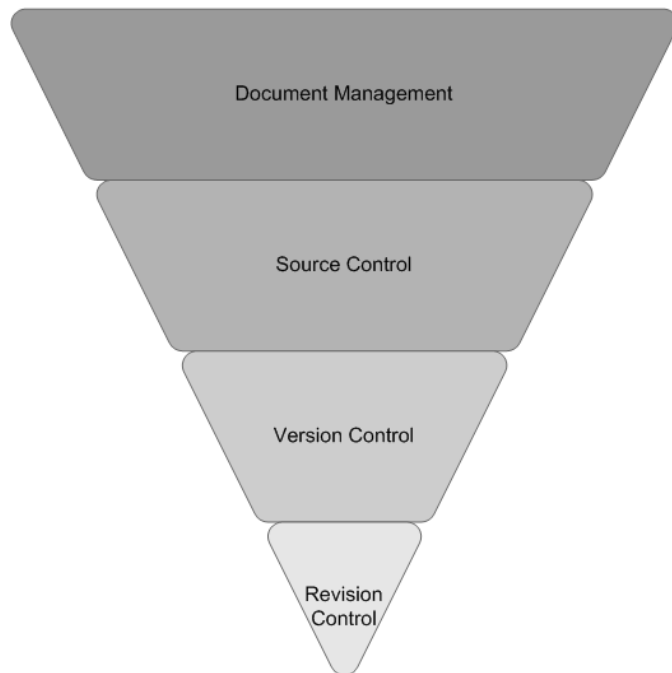
PhUSE 2009

Document Management: Goes beyond the definition of source control to include the ability to control file types beyond simply code files. In itself this is not different functionality to Source Control; however document management systems should offer workflow management functionality, printing etc.

Source Control: In addition to version control, refers to the central access, security and backups to the entire repository of code.

Version Control: In addition to revision control, this refers to the collection of programs used to create a specific version of a project.

Revision Control: Refers to the modifications to individual programs within a project.



The Cumulative Use of Source Control Terminology

Cumulatively *source control* should provide at least the following functions:

- Centralised repository: A central storage location, i.e. a repository, where project(s) files are stored. The historical versions of those files are stored here too. Both network and source control security should be used to manage access to the repository;
- Security: The integrity of the source control repository is paramount. Whatever forms the repository - e.g. a database, physical files and folders etc - it should be routinely backed up by your network team. Access to this location should be restricted so far as possible. Professional software packages support varying read, write and delete privileges to the contents of the repository;
- Team Synchronisation: This functionality allows all team members to synchronise their local working copy of the programs with the collective edits in the central repository;
- Exclusive Editing: Sometimes referred to as *playpens*, *sandboxing* or project repositories¹ which allows a programmer to make changes to SAS code without the risk of another team member editing the code at the same time. It also allows local code changes to be discarded should the programmer decide the code changes are unsuccessful. This process of exclusive editing is often referred to as *checking out* and the entering of code changes into the repository is known as *checking in*.
- Reverting to previous file versions: Source control should support the ability to rollback code to a previous version. Additionally, it should be easy to identify which version of a file is associated with a version of the entire project, or indeed easily be able to identify which versions of programs were used to create any given version of a project.
- Comparison between file versions: A mechanism to visualise the differences between versions of code should be integrated. This function is commonly utilised by programmers and there are many commercially available tools that provide such functionality to a greater or lesser degree of sophistication.

Additionally; the most basic of vendor supplied source control systems will support a technique called *branching*. In short this allows multiple programmers to concurrently edit the same file and merge the edits together. I have rarely observed this in practice with SAS code.

At this point it is appropriate to say that *configuration management* is not the same thing as source control. Configuration management is wider ranging. If anything source control may be a function which supports configuration management. Also avoid the term *change control* to prevent confusion with the same term used in project management to manage change of project scope and requirements.

PhUSE 2009

WHAT IS THE ISSUE ARE WE TRYING TO SOLVE?

An issue is *an important subject of debate*². The debate within organisations is to determine what level of control is required, and how it is to be achieved. There are many commercially available and open source solutions available with varying functionality, scalability and cost of ownership. Selecting the tool, if indeed there is one, which meets your current and medium term needs requires assessment.

The following are technical requirements that demonstrate the benefits of source control:

- The ability to restore any version of a program previously checked into your source control database. Relying upon network backups alone is not sufficient. Network backups are snapshots at timed intervals, typically daily or every few hours. Source control stores the version of code at the point your programmers check it into the database.
- Reproducing an output created with SAS from the code at its state perhaps several months ago, or even years ago requires the identification of exact code versions. This requirement also brings into question the need to source control not only code, but also input data, logs and outputs. Here the discussion can be seen to crossover with *document management systems*. Such systems are designed to encompass a wide range of disciplines and contents. Their tight integration with SAS programming environments is complex and are observed to provide weaker support for specialist functions provided by source control software.
- The ability to demonstrate that code quality assured on a previous project, in a test environment or simply at previous point in time can be provided with file comparison utilities. Most source control systems, to a greater or lesser extent, have such functionality within.

The rationale for using a form of source control is therefore clear. To manipulate a quote from Alexander Pope: To err is human, to correct takes time.

Source control should not be complicated, time consuming or introduce obstruction to the development of SAS programs. Ideally it should be a process integrated with the writing of SAS code, i.e. fit for purpose.

In many organisations a standardised process is used to control the storage of code in folders or zip files; whilst manual and open to mistakes, it has been considered sufficient if staff are appropriately trained.

WORKING WITH SOURCE CONTROL AND SAS CODE

High end SAS solutions, such as SAS Drug Development, support source control or document management within the solution. However, the tool of choice for SAS programmers is typically Enterprise Guide and classically the Display Manager, neither of which supports source control as defined in this paper. This section addresses methods of integrating source control functionality with SAS programming environments.

GROW YOUR OWN?

Both organisations and smaller teams are frequently observed to develop their own solution to the problem. Typically this takes the form of storing code, data and other files in date named folders or zip files. This provides a solution that directly fits the problem and is sufficiently agile to be modified with shifting requirements.

Zip files, or folders are typically named similarly to the following:

```
\path\Project Name yyyyymmdd v1.0
```

This is a method of freezing code, data and other objects at a given point in time or version.

The strengths of such approaches are speed and agility. Any user with appropriate disk access can perform the job.

Clearly, the weaknesses with this process are security and integrity. It is company guidelines, user training and attention to detail that govern the integrity of what is source controlled. The use of zip files (otherwise known as compressed folders) can offer some security with the password protection of zip file contents, when an appropriate zip file utility is used.

PhUSE 2009

SELECTING SOURCE CONTROL SOFTWARE

Commercial solutions, presented in order of the most frequently used according to a survey of software developers in 2008³:

- Microsoft's Visual SourceSafe in its original form is now over ten years old, but is still the most frequently used tool in the marketplace. Its shortcomings relative to newer tools are often seized upon by many in the anti-Microsoft camp, but do not be misled; it is quite suitable for small teams of locally based programmers. There is also a recently developed tool from Microsoft, designed for large and distributed teams of developers, Microsoft Team Foundation Version Control. Supporting both a command line interface and object embedding, it provides options for integrating with SAS. The cost of ownership depends on the version of SourceSafe. The original is relatively inexpensive, and requires no server or specialist knowledge.
- Subversion: Open source system built to better the functionality of CVS (see below). Supports (indeed seems to promote) the use of command line interfaces, therefore easy to loosely integrate with SAS.
- StarTeam: Comprehensive client – server solution from Borland. Designed for large distributed teams. Whilst appearing to comprehensively support some development environments, customising the solution to integrate with SAS would appear to be more cumbersome relative to the other solutions.
- CVS: A free solution openly available and used under the GNU General Public License. Whilst there are many freely available client interfaces, CVS is itself a command line driven interface, therefore easily to loosely integrate within SAS. Whilst widely respected in many programming communities, CVS lacks the enterprise level support and features that larger organisations and teams would need and expect.
- Perforce: Another comprehensive client – server solution from Perforce Software. Supporting many rich features this product also has an impressive list of software with which it integrates. There are also many API's which may be downloaded to support further third-party integration from C/C++ through Java and Windows programming languages. Perforce could be tightly integrated into any SAS tools that support extension through API's, such as Enterprise Guide, etc. The cost of ownership increases with this option.
- Vault: Describes itself as a *compelling replacement for Microsoft's Visual SourceSafe* from SourceGear. Pitched directly against Subversion, but with additional features, it is freely available to single users and relatively inexpensive to procure although clearly aimed at smaller user groups than StarTeam, Perforce and ClearCase. Supports a command line interface, therefore loosely integrated within SAS tools.

Notable other option:

- IBM Rational ClearCase: Describes itself as the complete *Change Management Solution*, and is generally considered a Rolls-Royce level product, with all the expected functionality and features such as electronic signatures. Aimed at larger teams and groups and designed to be hosted on a Windows, Solaris or Linux server. It will almost certainly be the most expensive to own from those reviewed in this paper. Pricing is openly available from IBM's website.

INTEGRATING WITH THE SAS DISPLAY MANAGER

It is quite feasible that any of the source control solutions supporting a command line interface could be incorporated within the Display Manager through one of two routes:

- Issuing operating system commands: Performing tasks such as getting latest versions of course code from the repository, then submitting the code through %INCLUDE statements or other means is quite possible. The advantages of such methods are relatively minimal. Security is easily bypassed with this option; hence its advantages are reduced to automatically synchronising local copies of SAS code with those from the central repository.
- A bespoke solution with SAS/AF[®]: Building a tool that integrates within the display manager via SAS/AF could add a level of sophistication. The ability to synchronise SAS code with a central repository using a level of security, advances the sophistication of the previous method. This is achievable with SAS/AF and its language, SCL. The command line functionality required is now hidden from the users view, but integration is still loose, meaning SAS must issue commands and wait for a set of anticipated responses from the source control solution.

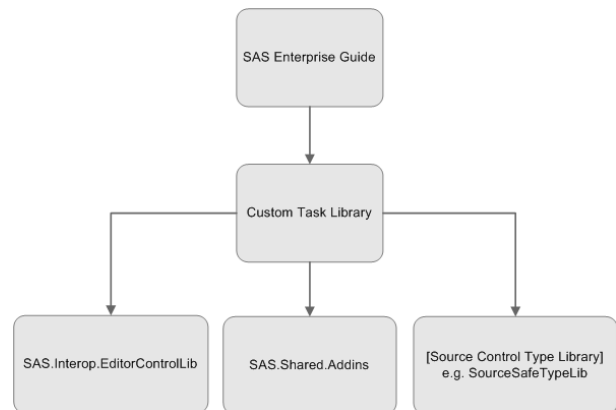
INTEGRATING WITH SAS ENTERPRISE GUIDE

Enterprise Guide provides an application programming interface (API) for use by programmers of the same languages in which Enterprise Guide itself was written. Embedding source control solution functionality is therefore achievable with knowledge of languages such as C# or Visual Basic.NET. This gives tighter integration than working with command line interfaces. When working with object models, programmers have far greater control when performing functions in external systems and handling unexpected errors.

A new task can be developed for Enterprise Guide, similarly to the existing Program Task found in Enterprise Guide 4.2. This can connect directly to a source control repository, placing the desired SAS program directly into the code window. Programs can then either be run within the task window or as part of the Enterprise Guide project.

Successfully integrating source control into a task requires understanding of using the Enhanced Editor ActiveX control, the Enterprise Guide Add-In object model and the object model provided by your source control solution. These libraries (DLL's) supply all the functionality needed to build a new task for use in Enterprise Guide, which itself is yet another library.

This hierarchy of libraries is displayed (right). Demonstrations are available of such an add-in at work within Enterprise Guide 4.2 during presentations of this paper.



EXISTING SAS SOLUTIONS WITH INTEGRATED SOURCE CONTROL

Although this paper is aimed at those who generate or write SAS code, it would not be balanced without drawing the reader's attention to the SAS Solutions that support Source Control natively.

SAS/AF

The classic Display Manager has native source control for SAS/AF developers called Source Control Manager (SCM). Available since SAS 6 and still available in SAS 9.2 under the Solutions -> Development and Programming menus in the SAS Display Manager.

SCM is designed for small groups of SAS/AF developers who wish to source control their AF frames and SCL code, rather than those who program in the 4GL language.

SAS DATA INTEGRATION SERVER

Data Integration (DI) Server supports a level of source control. Whilst features such as rollback and file versioning are omitted, security and concurrent access are well presented.

The figure below is of the History window for a job (ultimately a SAS program) in DI Studio called Import_Efficacy. Each time the programmer checks in a new version of the code, DI Studio prompts for a title and description of the edit. It is also clear when the job was originally added to the repository.

History for: Import_Efficacy					
History: 3 items					
Ver...	Title	Description	User	Date/Time	Action
3	QA Edits		David.Shannon	02-Jun-2009 21:44:40	Checked in
2	Initial Version		David.Shannon	02-Jun-2009 21:11:06	Checked in
1	Import_Efficacy			02-Jun-2009 21:11:06	Created

A useful audit of amendments can be displayed from this information; however comprehensive source control is not truly integrated within DI Studio at the time of writing (DI Studio 4.21).

PhUSE 2009

SAS DRUG DEVELOPMENT

SAS Drug Development is a comprehensive server based solution which integrates a Document Management System for SAS files. Typically hosted by SAS Institute this solution offers the best support for integrated source control from SAS Institute.

CONCLUSION

In conclusion, have we solved the Source Control Issue for SAS Programmers? There are hundreds of thousands of active SAS programmers globally and their needs differ between organisations and even between groups within organisations. The following conclusions will lead you to your own answer:

Firstly, we have understood that the meaning of Source Control is different to different user groups, and that terminology varies. A cumulative use of these terms is proposed to add clarity for intent of actions and use when referring to this topic:

Revision Control → Version Control → Source Control → Document Management

The issue of source control is discussed. Primarily the issue is driven by ISO 9000 standards and regulatory requirements. A need to reproduce outputs generated by SAS at any time in the past is born from these standards. Depending on your environment, source control is either a requirement or a “nice to have” option.

This paper considered existing methods of source control and the education available on the subject. Many small to medium sized teams and organisations use internally developed techniques such as copying code to alternative folders, or zipping SAS files with date stamps for future reference.

A brief review of both commercial and open source solutions is presented, with varying costs of ownership. It is clear there are varying levels of functionality, scalability and cost of ownership.

Pseudo methods are presented for integrating source control systems within the most commonly used tools of SAS Programmers, i.e. Display Manager Sessions and Enterprise Guide. The author has successfully integrated Microsoft Visual SourceSafe within Enterprise Guide, proving it is possible to source control SAS program files within this environment.

Finally, this paper considers those solutions targeted at specific needs; i.e. Data Integration Studio and Drug Development. The latter has the best source control offering from SAS Institute, albeit a relatively exclusive solution.

REFERENCES

1. SAS Data Integration Studio terminology.
2. The Oxford Dictionary, Oxford University Press, 1992.
3. VSOF Technology Blogs, Top Ten Version Control Systems, 28th August, 2008

CONTACT INFORMATION

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

David Shannon
Amadeus Software Limited
Orchard Farm, Witney Lane, Leafield, Oxfordshire. OX29 9PG.
Work Phone: 01993 878287
Fax: 01993 878042
Email: david.shannon@amadeus.co.uk
Web: www.amadeus.co.uk

Brand and product names are trademarks of their respective companies.