

Paper 145-2007

Projects, Programs and Links – Oh My!

Jerry A. High, Blue Cross and Blue Shield of Minnesota, Eagan, MN
Barbara J. Miller, Blue Cross and Blue Shield of Minnesota, Eagan, MN

ABSTRACT

As new desktop tools are introduced in the work environment it is common to assume that users inherently know how to use them because they are built on currently used technologies. Sometimes basic concepts are not clearly understood. Sometimes these concepts need to be restated in a way that relates them to their work.

The past year, Blue Cross Blue Shield of Minnesota deployed SAS Enterprise Guide (EG) v3. It was done with little user training. For many of the users, submitting programs was done via batch processing or an X-windows environment. It was easy to see where the programs needed to perform these tasks could be stored and then retrieved for further use or passing code onto others. At that point, most code was written as a single block that was very portable between users.

When moving to SAS EG v4, we decided that we should spend a little time with the SAS user community to explain how projects and programs relate and to also explain the concept of links and when/why they would be used.

This poster will visually explain the difference between saving your code within a SAS EG Project versus saving the code externally as a link and some reasons why you would choose one over another.

INTRODUCTION

Most desktop tools that people use today are packed full of features to help users get tasks done. Whether it's Microsoft's Excel spreadsheet or SAS' Enterprise Guide there are features that may not be commonly used, but can help a user do their work more effectively. For example, some may not be aware that one can highlight an area and drag the lower right corner and Excel will automatically fill the target cells based on what is in the highlighted area.

Traditional SAS programmers, who have picked up Enterprise Guide as their interface into a SAS server, might not have explored the differences of saving their code within the project versus saving them as links to the project. The proper use of these two approaches can make a difference in how you create programs and share them among your colleagues. We will explore the use of programs and links and give examples where they can help analysts be more effective programmers.

DEFINITIONS

A Project created in Enterprise Guide (EG) contains programs, data results, output and other components. You may recall with EG that programs can be either complete programs or a specific code block within a sequence of code blocks that may or may not be run together. In this paper we will use the term "program" even though it can be simply a code block. Programs can be saved within the project or saved external to the project. Programs saved external to the Project are accessed via Links within the Project. See Figure 1 below.

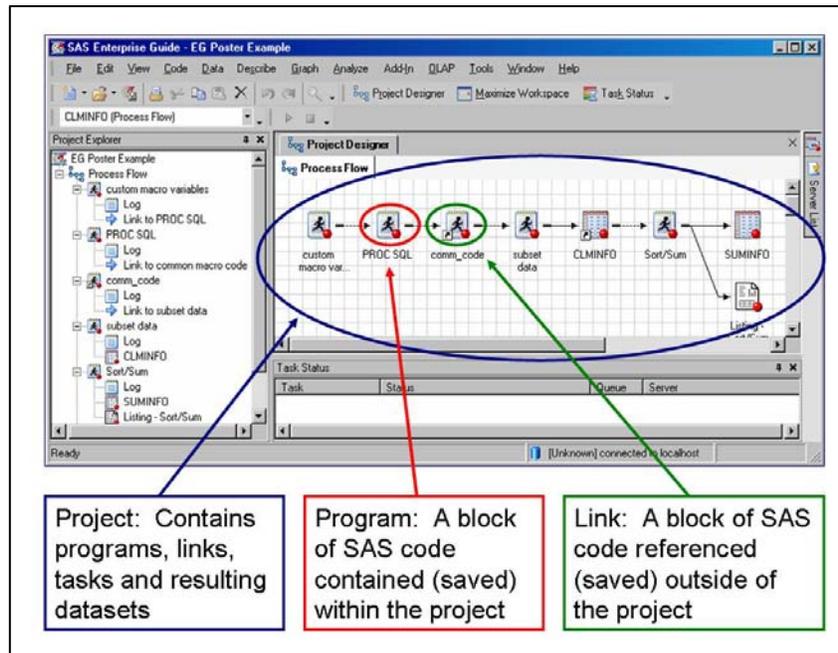


Figure 1

SAVING PROGRAMS AND LINKS

Once a Project is created, the artifacts within the Project can be saved. Saving the Project saves the programs (or code blocks) created within the Project as part of the Project. If you would like to save the programs external to the Project, you should first highlight the icon containing the program to be saved and select save (name of program) under the File menu list. See Figure 2 below.

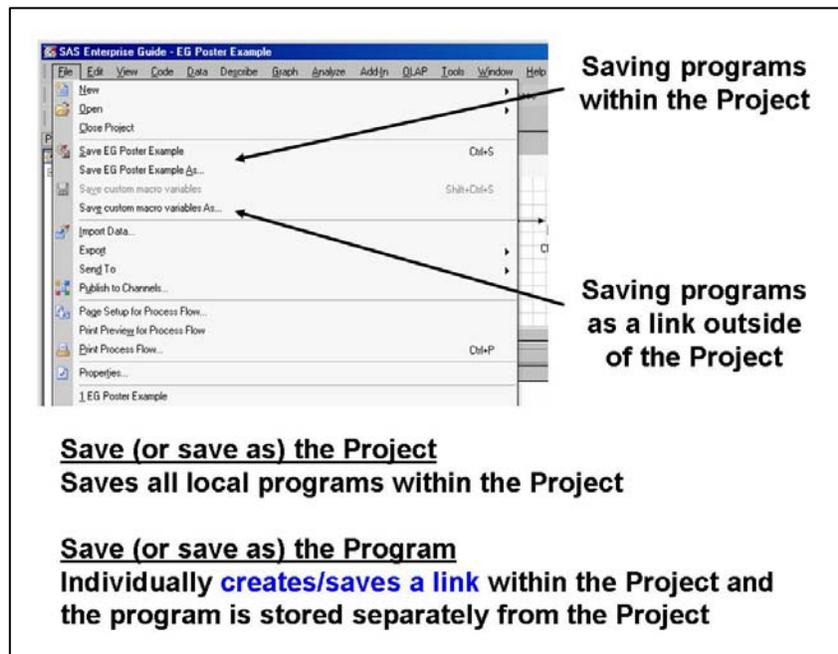


Figure 2

DIFFERENTIATING PROGRAMS FROM LINKS

There are a couple ways in which you can tell if a program is part of the Project or simply a Link to an external program.

The first way is to look at the icons within the Project. If you look carefully at an icon, you may see a small boxed arrow in the lower left-hand side of the icon. If that appears, it represents a Link to an external program. Absence of the boxed arrow indicates that the program is contained (saved) within the Project. This approach is similar to the approach you can see on your Microsoft Windows desktop. The term used here is a Shortcut. A Shortcut on the desktop (indicated by the boxed arrow in the lower left-hand corner of the icon) works the same way as a Link within the Project. See Figure 3.

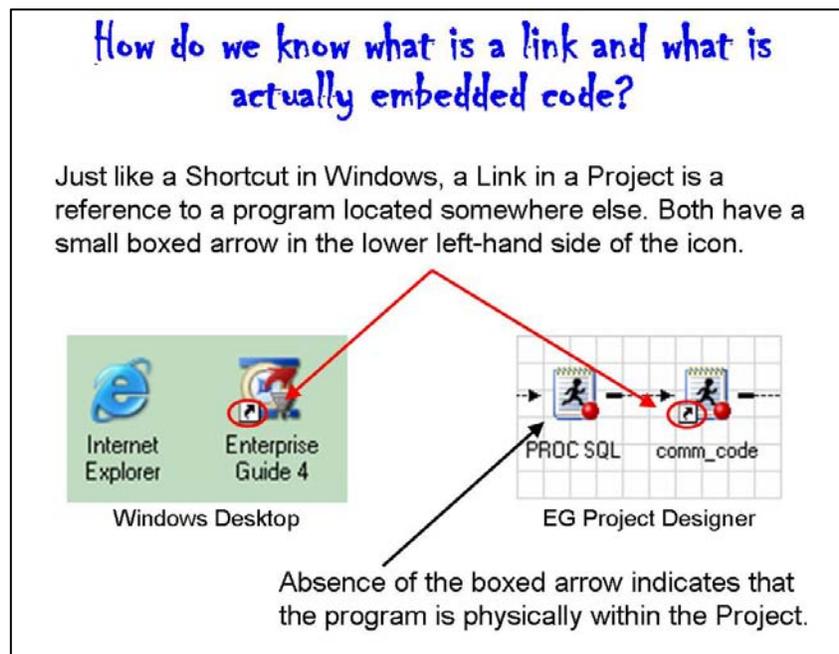


Figure 3

A second way to tell if a program is part of the Project or a Link to an external program is to review the properties of the program. To do this, highlight the icon of the program and select Properties under the File menu list. Alternatively, again with the icon highlighted, right click with the mouse and select Properties from the displayed menu list. If it is a program within the Project the description in the Properties box will state "(Embedded in Project)". If it is a Link to an external program to the Project the description will show the directory path and the program name. See Figure 4.

You should also note that within the Properties box that you have the option to "Save As" which will save the program within a Project external to the Project. For Links you have the option to "Embed" the external program within the project and discontinue the reference to the external program. You also have the option to "Save As" a different Link.

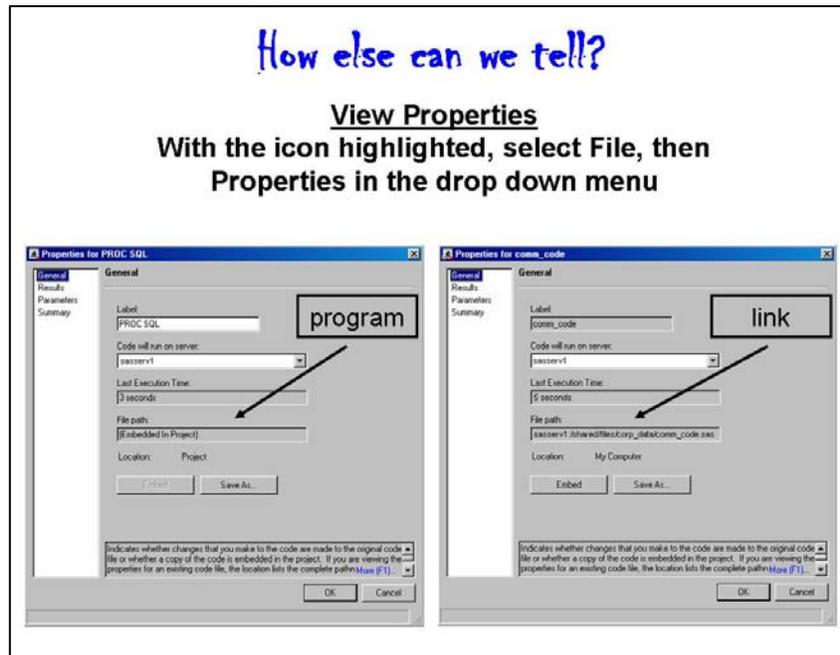


Figure 4

UNDERSTANDING HOW PROGRAMS AND LINKS ARE STORED

When you save your project all programs are stored as part of the Project. No external files other than the Project itself are created. If the Project contains Links, no updates are made to the external program that is referenced by the Link. To update the program referenced by the Link, you must choose to save the Link (hence the program). When you create a Link, the program is saved on its own in the directory specified using the name of the Link with the .sas extension. Since the Link only contains a reference to the external program, the size of the saved Project is much smaller.

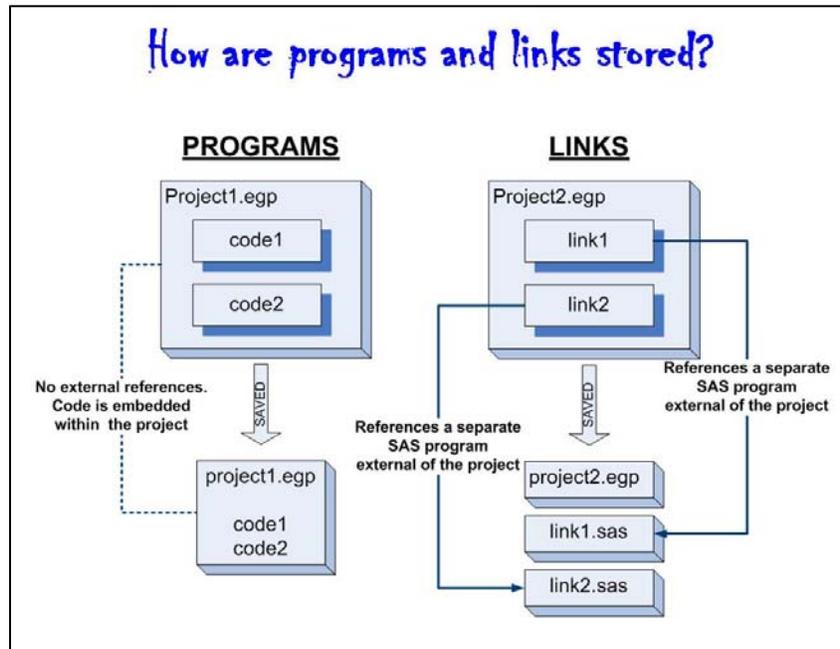


Figure 5

PORTABILITY OF PROJECTS

In the past when you wanted to share a SAS program, you merely copied (or moved) it to another directory. The other analyst could then run it without any issue. With Projects you need to consider the programs and Links within the Projects before you simply move them. As an example, consider a Project with one Linked program within it. If that Project is located in a private directory where other analysts don't have security rights, they cannot access or run that Project (see Figure 6).

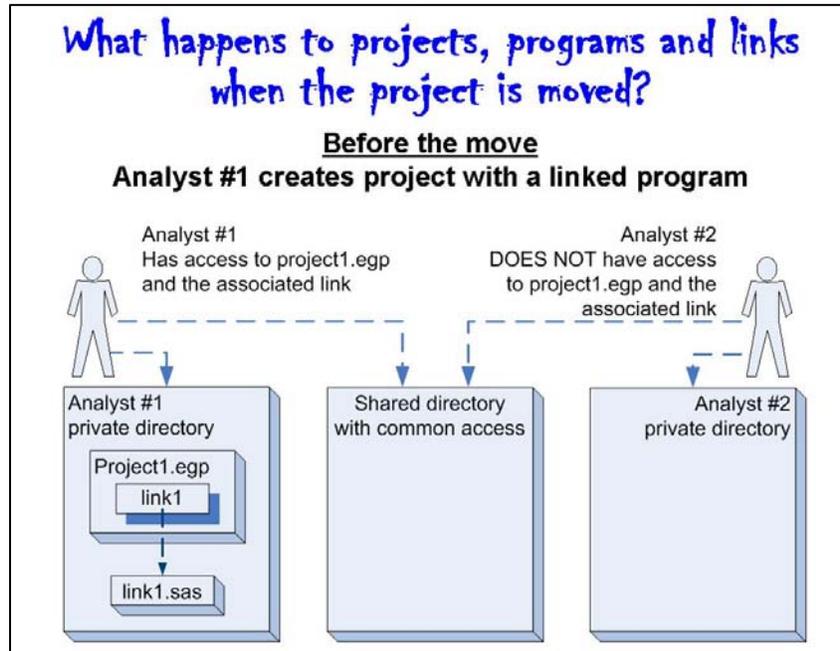


Figure 6

If you move only that Project to another shared directory where other analysts do have security rights, they will now have access to the Project. But what is important to understand is that even though other analysts have access to the Project, it will fail to execute properly since the program that is referenced by the Link is still in a restricted directory (see Figure 7).

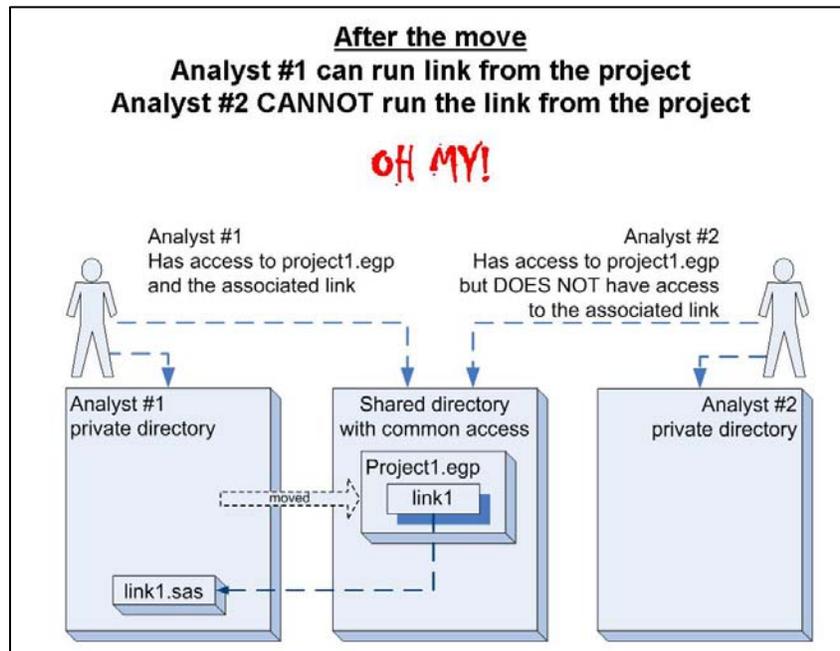


Figure 7

The program itself also needs to be in a directory to which they have security rights. Again, this is not the final solution. As illustrated in Figure 8, even with the SAS program in the same directory, the Project still refers to the program in the original location under the private directory. This means that when other analysts try to run the Project, the Linked program will fail due to lack of security rights to the program.

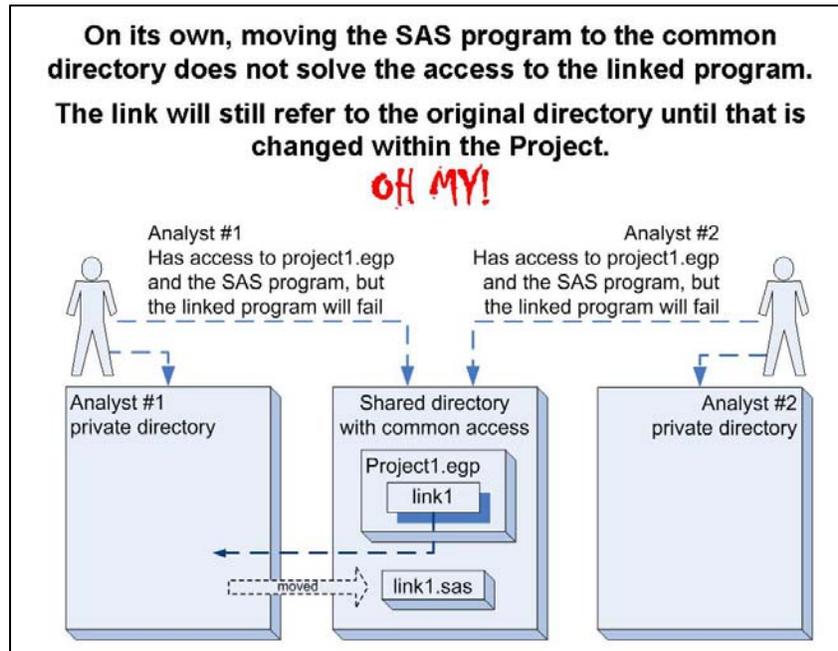


Figure 8

The final step needed is to modify the Properties of the Link in the project to point to the new directory. Once that is done, other analysts can access both the Project and the Linked program within the project. A summary is described in Figure 9.

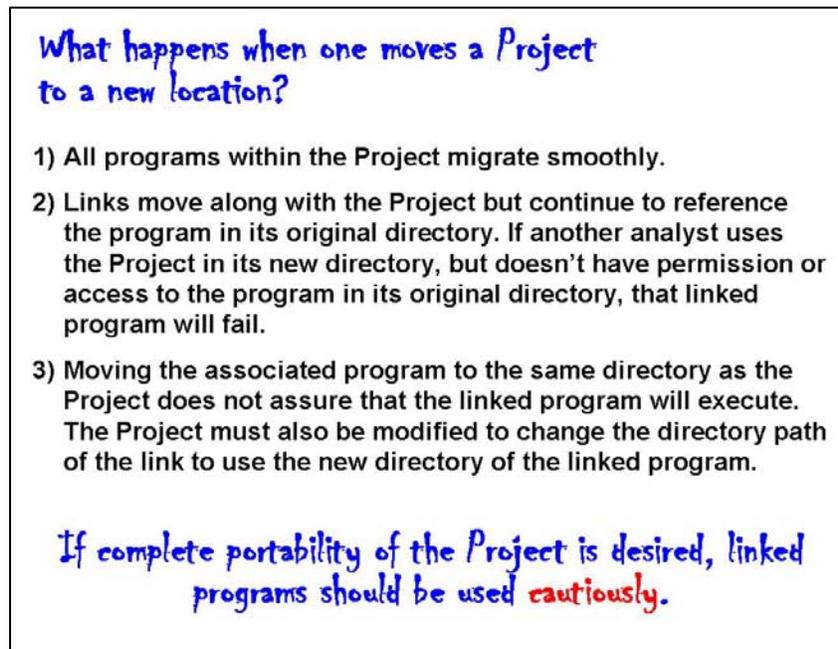


Figure 9

USING LINKS FOR MORE EFFECTIVE PROGRAMMING

With the concerns/issues of moving Projects that contain Links, you might wonder why anyone would want to use them. Links are best used with common programs (or code blocks) that need to be changed or updated periodically. If you have programming logic that is useful among many analysts, you can save it in a shared directory where it can be accessed by many Projects. These programs could be blocks of codes such as macro routines or format routines. By keeping the common program in one location updates only need to be performed in one location, not in multiple Projects. This is illustrated in Figures 10 and 11.

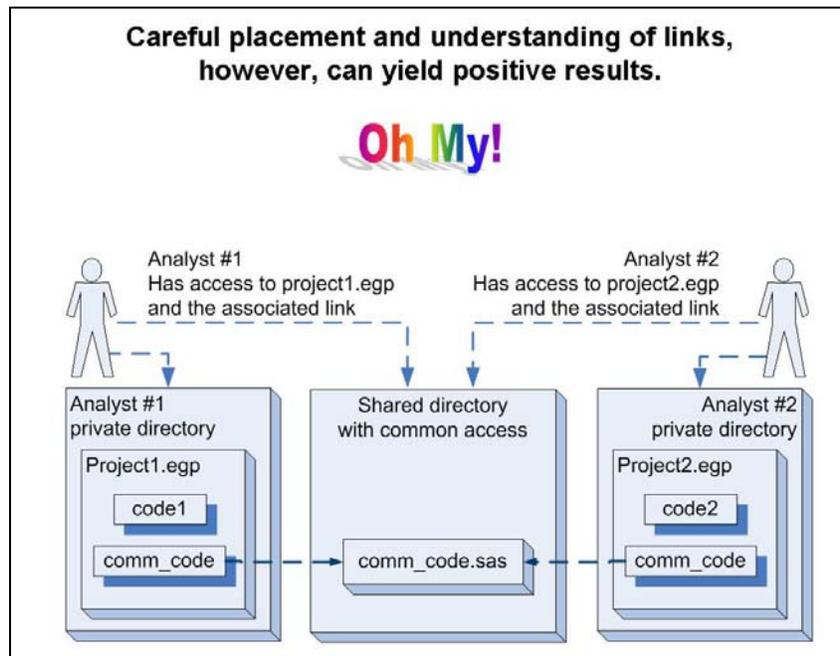


Figure 10

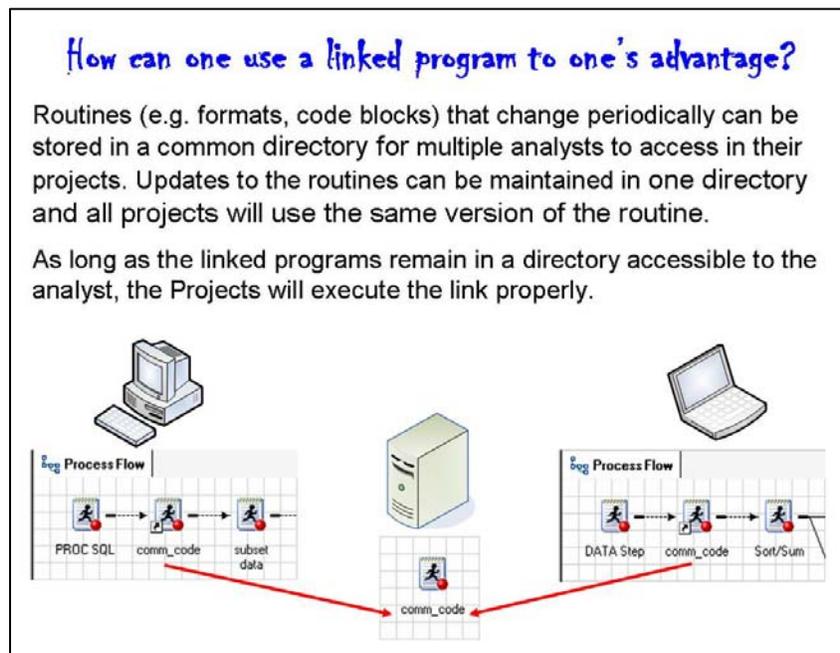


Figure 11

ACKNOWLEDGMENTS

We would like to thank Paul Christenson, Kevin Cornell and Mike Wesley from our analytic tools group for their assistance in the preparation and review of this poster.

RECOMMENDED READINGS

Slaughter, S. J., Delwiche, L. D. (2006). *The Little SAS® Book for Enterprise Guide® 4.1*. Cary, NC: SAS Institute Inc.

Getting Started with SAS® Enterprise Guide®. SAS® Online Tutorial. Retrieved February 27, 2007, from:
http://www.sas.com/apps/elearning/elearning_details.jsp?pubcode=59271

CONTACT INFORMATION

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

Jerry A. High
Blue Cross and Blue Shield of Minnesota
3535 Blue Cross Road
Eagan, MN 55122
Work Phone: 651-662-8548
E-mail: jerry_a_high@bluecrossmn.com

Barbara J. Miller
Blue Cross and Blue Shield of Minnesota
3535 Blue Cross Road
Eagan, MN 55122
Work Phone: 651-662-1145
E-mail: barb_miller@bluecrossmn.com

SAS and all other SAS Institute Inc. product or service names are registered trademarks or trademarks of SAS Institute Inc. in the USA and other countries. ® indicates USA registration.

Other brand and product names are trademarks of their respective companies.