

Realize synchronized updating of SAS code by hard link

Xiaoying Wang, BeiGene, Inc.

ABSTRACT

As we all know, there are many analysis tasks in a study. For example, SMC, DMC, DSUR, CSR etc. The SDTM, ADaM and TFL SAS programs could be shared between task folders with different raw data and different cutoff dates. It is repeated work if they are updated separately. It is error-prone and inefficient if we manually copy SAS programs from one task folder to another. Now, we can use hard link to synchronize SAS programs between task folders. Compared with manual copy, it will be more efficient and eliminate inconsistencies between task folders. We can also unlink a task folder from others on SAS program synchronization, if needed.

INTRODUCTION

In our daily programming work, do you often encounter the following situations?

1. Do you find yourself manually copying SAS programs from one task folder to another when starting a new task?
2. Do you sometimes forget to update some SAS programs in a task folder to make it consistent with others?
3. Do you ever struggle which task folder the latest version SAS programs are in?

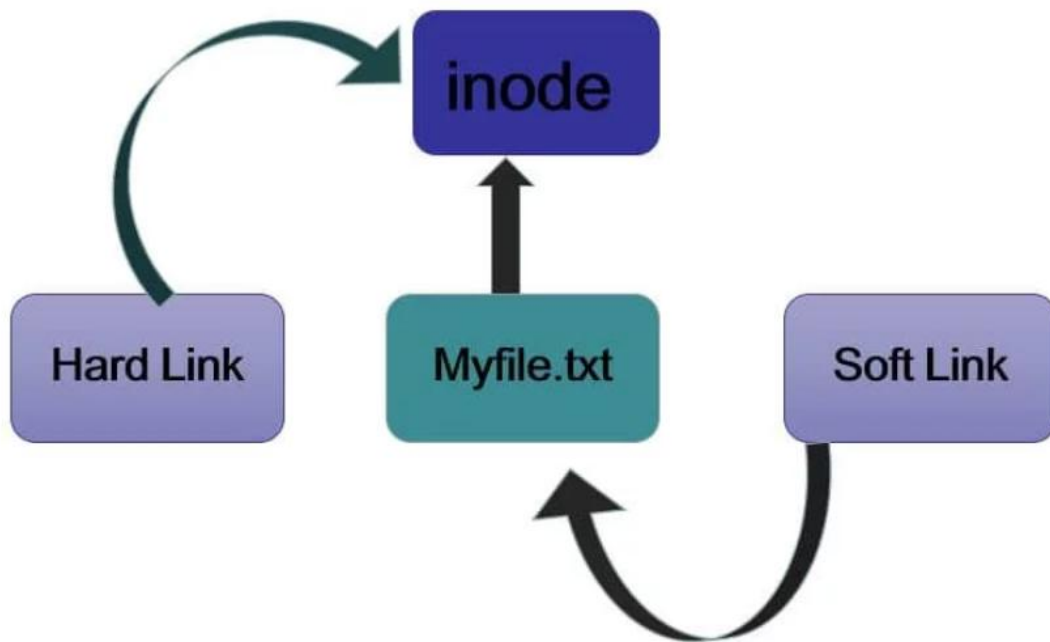
These issues can make our work inefficient and error prone. To address the issues, we are introducing Unix links to synchronize SAS programs between task folders. With Unix links, updating a SAS program in one task folder means updating all linked SAS programs in all task folders, without manual copy but 100% consistency.

WHAT IS A LINK IN UNIX

- A link in UNIX is a pointer to a file.
- Creating links is a kind of shortcut to access a file.

There are two types of links in Unix:

1. Hard Link
2. Soft Link or Symbolic Links



An inode, short for "index node", is a data structure in a file system that stores metadata about a file or directory. It contains information such as the file's size, permissions, timestamps, and pointers to the actual data blocks on the storage device where the file's contents are stored.

HARD LINK

- Both the original file and hard linked file have the same inode numbers, therefore they refer to the same physical file location.
- If the original file is deleted, the link is not deleted, hard links are more flexible.
- Are more efficient (in terms of access time) as compared to soft links.
- This can not be used to link a file on another file system, or to link directories.

Syntax in SAS:

```
X "ln [original filename] [link name]"
```

SOFT LINK

- The original file and the link have different inode numbers. Each soft linked file points to the original file.
- When the original file is deleted, the link remains as an invalid link and will not behave correctly. It is called as hanging link.
- Soft Link contains the path for original file, if original file is removed it impacts the link functionality.
- This can be used to link file on a different file system.
- Can be used to link directories.

Syntax in SAS:

```
X "ln -s [original filename] [link name]"
```

THE ADVANTAGE OF HARD LINK

Either hard link or soft link could be used according to different requirements. We use hard link to synchronize SAS programs, considering that linked files will not be deleted if original files are deleted accidentally. If some linked SAS programs are deleted accidentally, they can easily be reconnected to the hard link. Symbolic link is different. Deleting the original file means deleting all linked files.

INTRODUCTION FOR THE MACRO %PGMLINK

THE FUNCTION OF MACRO M_PGMLINK

1. Hard link SAS programs between task folders. This means that any updates in a SAS program will reflect in all linked folders, allowing for seamless collaboration across multiple task folders.
2. Archive the original programs in the task folders.
3. Unlink the task folders when a task folder does not need to be synchronized with others.

MACRO

```
%m_pgmlink(cenfold=, /*the central folder that keeps the latest pgm  
linkfold=, /*the task folder to hard link pgm in central folder  
lock=N);
```

Parameter	Value	Description
cenfold	*path*	The central folder that keeps the latest sdtm and adam pgm.
linkfold	*path*	The task folder to hard link sas pgm in central folder. After link, all pgm in this task folder and central folder are synchronized. They will be updated at the same time.
lock	Y;N	Default to N. N represents we will create the hard link between central folder and task folder. Y represents an unlink, the programs will be copied from the central folder. After this action, the program will no longer be updated in accordance with the central folder.

PROGRAMMING

```
%m_pgmlink(sdtmfold=/usrfiles/bgcrh/build/bgb_16673/bgb_16673_102/sdtm_rt,  
linkfold=/usrfiles/bgcrh/build/bgb_16673/bgb_16673_102/sdtm_test,
```

```
lock=N);
```

PROCESS OF THE MACRO

Step1: move all the original pgms in linked folder to archive folder

```
X "mv [linked folder/original filename]"
```

Step2: hard link the pgms in central folder to linked folder

```
X "ln [central folder/filename] [link folder/filename]"
```

Step3: unlink the pgms in central folder to linked folder

****remove the pgms in linked files**;**

```
X "rm [link folder/filename]"
```

****copy the pgms from central folder**;**

```
X "cp -p [central folder/filename] [link folder/filename]"
```

NOTE

1. Hard link can't be used to link directories. To link all the programs in the folder, we use a do loop. However, it is important to note that the files in subfolders will not be linked.
2. It is suggested that there should not be any irrelevant files in task folders. It is better to delete them before creating a hard link.
3. It is advisable to lock the task folder once the task is completed. In case you need to update synchronously again, you can also relink the task folder.

REFERENCES

[Unix Links \(Symbolic links\) /ln command in Unix - Tutorials Hut](#)

[Unix file basics : Inode, Soft Vs Hard link, Device files, Named pipes – The Geek Diary](#)

ACKNOWLEDGMENTS

Thanks to Chunpeng Zhao and management teams for their advice and programming colleagues for their help on this paper/presentation.

CONTACT INFORMATION

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

Xiaoying Wang
BeiGene, Inc.
xiaoying.wang@beigene.com