SAS® GLOBALFORUM 2015

The Journey Is Yours

Filling your SAS® Efficiency Toolbox:

Creating a Stored Process to Interact with Your Shared SAS® Server Using the X and SYSTASK Commands

SAS GLOBALFORUM

MEDICA®

You belong.

Filling your SAS® Efficiency Toolbox: Creating a Stored Process to Interact with Your Shared SAS® Server Using the X and SYSTASK Commands

Chad Stegeman and Cody Murray, Medica Health Plans

Introduction

SAS® Enterprise Guide® is a great interface for businesses running SAS® in a shared server environment. However, interacting with the shared server outside of SAS can require costly third-party software and knowledge of specific server programming languages. Using stored procedures we can:

- Copy or Move Files or Folders
- Change File or Folder Permissions
- Send specific commands to the server

Objectives

•Mastering X and Systask commands
•Leveraging the Server Log to print results
•Creating a Stored Procedure to make the code accessible to all users

X and Systask Commands

Printing Results with the Server Log

Utilizing the Stored Procedure

Conclusion

•Use SAS® functions to talk to the server/system.
•Utilize the log to output results to the results tab.
•Combine these with SAS® Enterprise Guide® Prompts to replicate the process.

References

-"X Command: Windows." *SAS(R) 9.2 Companion for Windows, Second Edition*. Copyright © 2015 SAS Institute Inc., n.d. Web. 16 Mar. 2015.

-"SYSTASK Statement: Windows." *SAS(R) 9.2 Companion for Windows, Second Edition*. Copyright © 2015 SAS Institute Inc., n.d. Web. 16 Mar. 2015.

Contact Information

-Chad Stegeman, chad.stegeman@gmail.com

-Cody Murray, murra582@umn.edu

X and Systask Commands

With the X and SYSTASK COMMAND functions users can pass the client-server commands directly through SAS®. These commands change depending on the SAS® environment used.

X Function

- The X COMMAND works with either PC or UNIX SAS.
- Using the X COMMAND with Windows will require some change to the code covered on this poster, but it is possible to make the code work. The commands will change from UNIX code to DOS command code. The command is passed to the operating environment and executed. *Only one command can be entered per X COMMAND in the PC environment.*
- The UNIX version of the X COMMAND processes commands differently than most functions. The X COMMAND prompts SAS® to start a shell to execute the commands that you specified. Each command can be processed differently depending on whether you entered one command or more than one command.

Systask Command

- The SYSTASK command works very similar to the X COMMAND with it having two different versions depending on what kind of SAS® being used.
- For the PC and UNIX versions of SAS® the SYSTASK runs commands as asynchronous tasks, this means that tasks are executed independently of all other tasks that are currently running. Asynchronous tasks run in the background, so you can perform additional tasks while the asynchronous task is still running.
- If there is an error on PC SAS® the command will cause the program to end. Additionally, there will output written to the SAS® log with Windows SYSTASK.
- In the UNIX environment, SYSTASK will output to the SAS® log and as expected converting PC SAS® programs with the SYSTASK command will most likely result in errors.

X Function Syntax

Windows Syntax:

X <'*command* '>;

no argument

open a DOS command window.

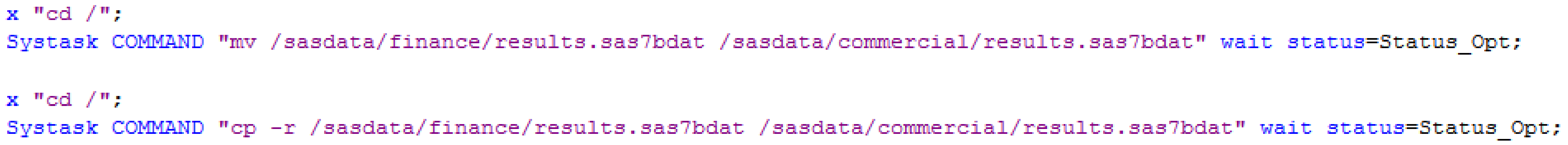
command

specifies the command that you want to execute.

UNIX Syntax:

X *UNIX-command*

X '*cmd1;cmd2....<;cmd-n>*



```
x "cd /";
Systask COMMAND "mv /sasdata/finance/results.sas7bdat /sasdata/commercial/results.sas7bdat" wait status=Status_Opt;

x "cd /";
Systask COMMAND "cp -r /sasdata/finance/results.sas7bdat /sasdata/commercial/results.sas7bdat" wait status=Status_Opt;
```

Systask Command Syntax

Windows Syntax:

SYSTASK COMMAND "*operating system command*"
    <WAIT | NOWAIT>
    <TASKNAME=*taskname*>
    <MNAME= *name-variable*>
    <STATUS= *status-variable*>
    <SHELL<="*shell-command*">>;
SYSTASK LIST <_ALL_ | *taskname*> <STATE> <STATVAR>;
SYSTASK KILL *taskname <taskname...>*;

UNIX Syntax:

SYSTASK COMMAND "*operating-environment-command*"
    <WAIT | NOWAIT>
    <TASKNAME=*taskname*>
    <MNAME= *name-variable*>
    <STATUS= *status-variable*>
    <SHELL<="*shell-command*">>
   <CLEANUP>;
SYSTASK LIST <_ALL_ | *taskname*> <STATE> <STATVAR>;
SYSTASK KILL *taskname <taskname...>*;

MEDICA®
You belong.

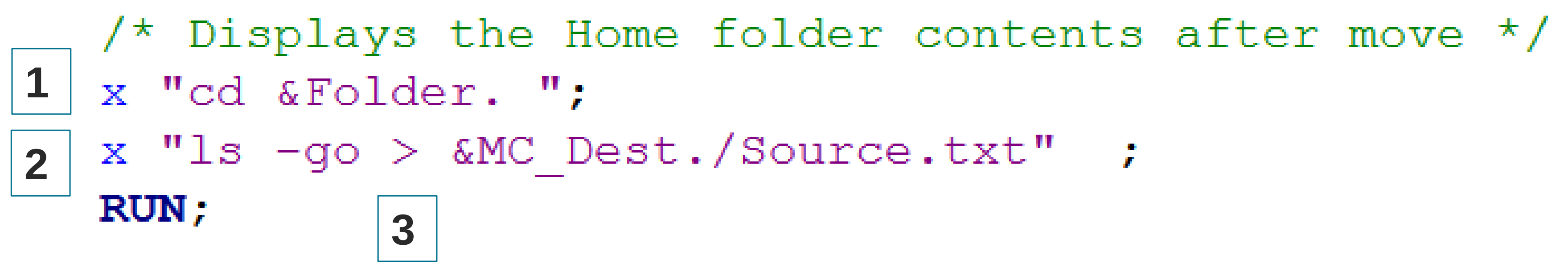
Filling your SAS® Efficiency Toolbox: Creating a Stored Process to Interact with Your Shared SAS® Server Using the X and SYSTASK Commands

Chad Stegeman and Cody Murray, Medica Health Plans

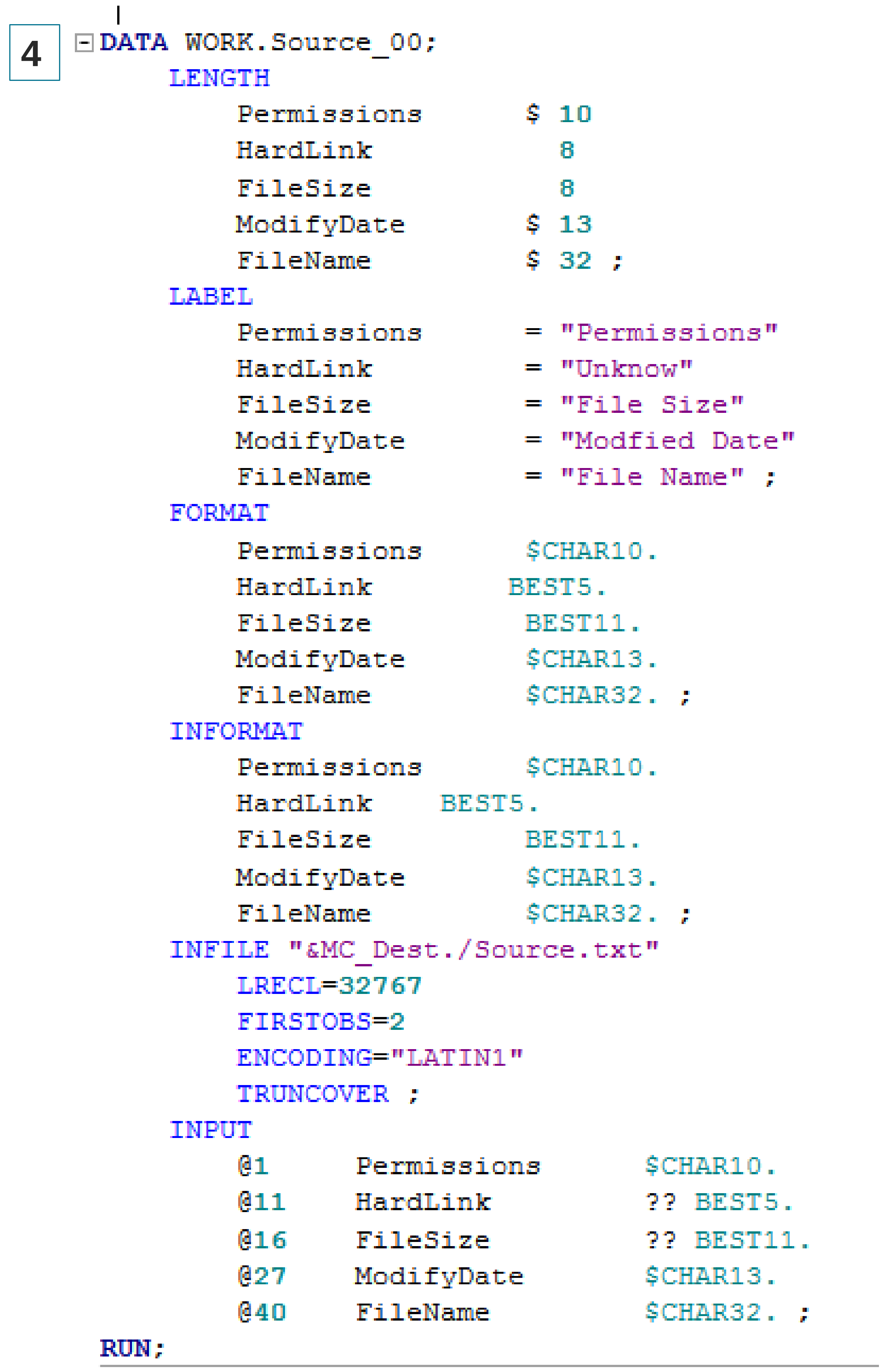
Printing Results with the Server Log

Syntax and coding will be slightly different depending on the platform you are running SAS®. *The example below uses SAS® in an AIX UNIX server environment:*

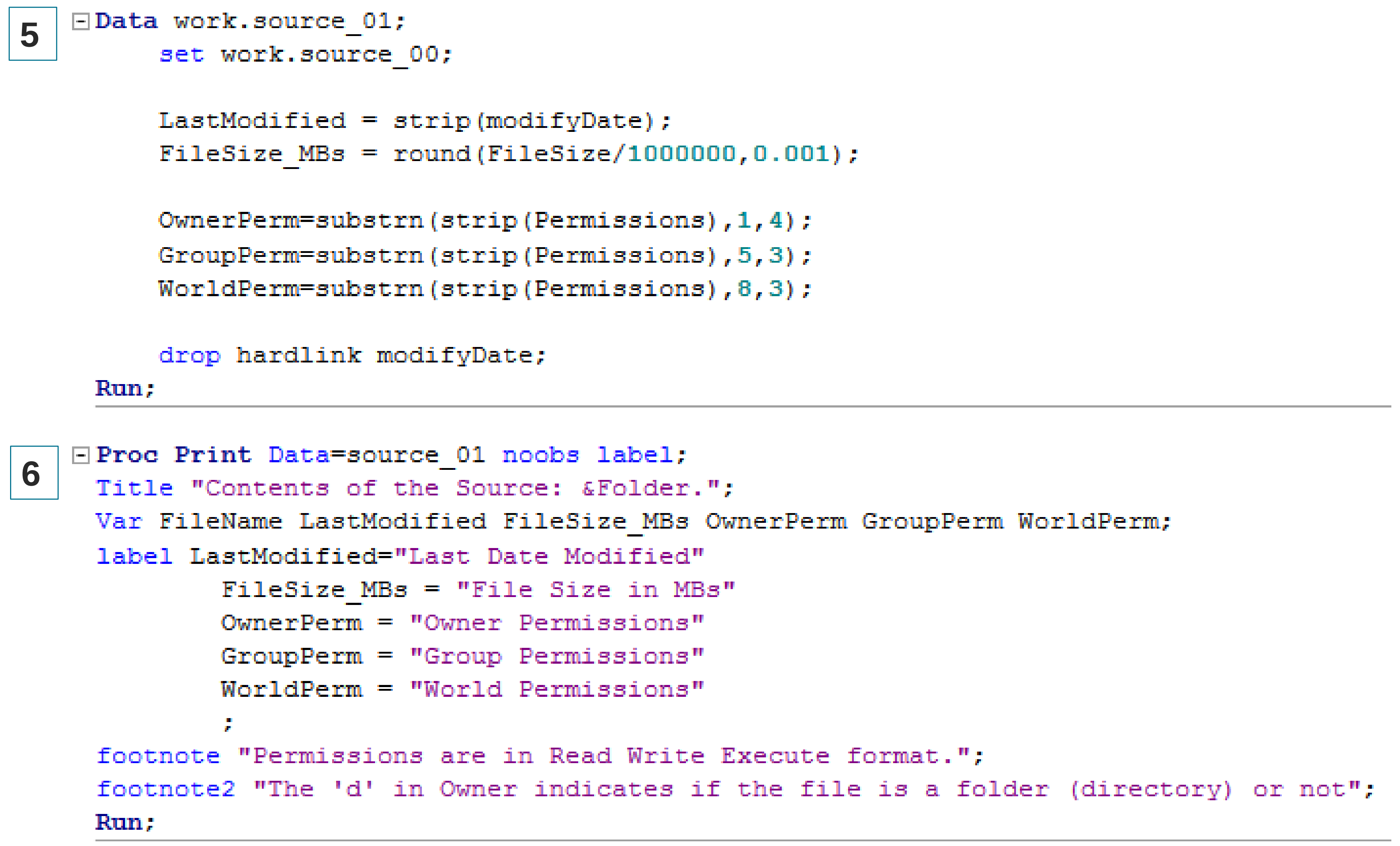
1. Go to the desired directory
2. Use the “ls” command to display the contents of the directory.
3. The “>” symbol writes the output to the destination text file
4. Import the text file into a SAS® dataset. You may need to edit the import process
5. Clean up dataset and separate permissions for end user
6. Use “Proc Print” to display the server log in the results.



```
/* Displays the Home folder contents after move */
x "cd &Folder. ";
x "ls -go > &MC_Dest./Source.txt"   ;
RUN;
```



```
DATA WORK.Source_00;
    LENGTH
        Permissions      $ 10
        HardLink           8
        FileSize           8
        ModifyDate       $ 13
        FileName         $ 32 ;
    LABEL
        Permissions      = "Permissions"
        HardLink         = "Unknow"
        FileSize         = "File Size"
        ModifyDate       = "Modfied Date"
        FileName         = "File Name" ;
    FORMAT
        Permissions      $CHAR10.
        HardLink        BEST5.
        FileSize         BEST11.
        ModifyDate       $CHAR13.
        FileName         $CHAR32. ;
    INFORMAT
        Permissions      $CHAR10.
        HardLink    BEST5.
        FileSize         BEST11.
        ModifyDate       $CHAR13.
        FileName         $CHAR32. ;
    INFILE "&MC_Dest./Source.txt"
        LRECL=32767
        FIRSTOBS=2
        ENCODING="LATIN1"
        TRUNCOVER ;
    INPUT
        @1      Permissions      $CHAR10.
        @11     HardLink         ?? BEST5.
        @16     FileSize         ?? BEST11.
        @27     ModifyDate       $CHAR13.
        @40     FileName         $CHAR32. ;
RUN;
```



```
Data work.source_01;
    set work.source_00;

    LastModified = strip(modifyDate);
    FileSize_MBs = round(FileSize/1000000,0.001);

    OwnerPerm=substrn(strip(Permissions),1,4);
    GroupPerm=substrn(strip(Permissions),5,3);
    WorldPerm=substrn(strip(Permissions),8,3);

    drop hardlink modifyDate;
Run;
```

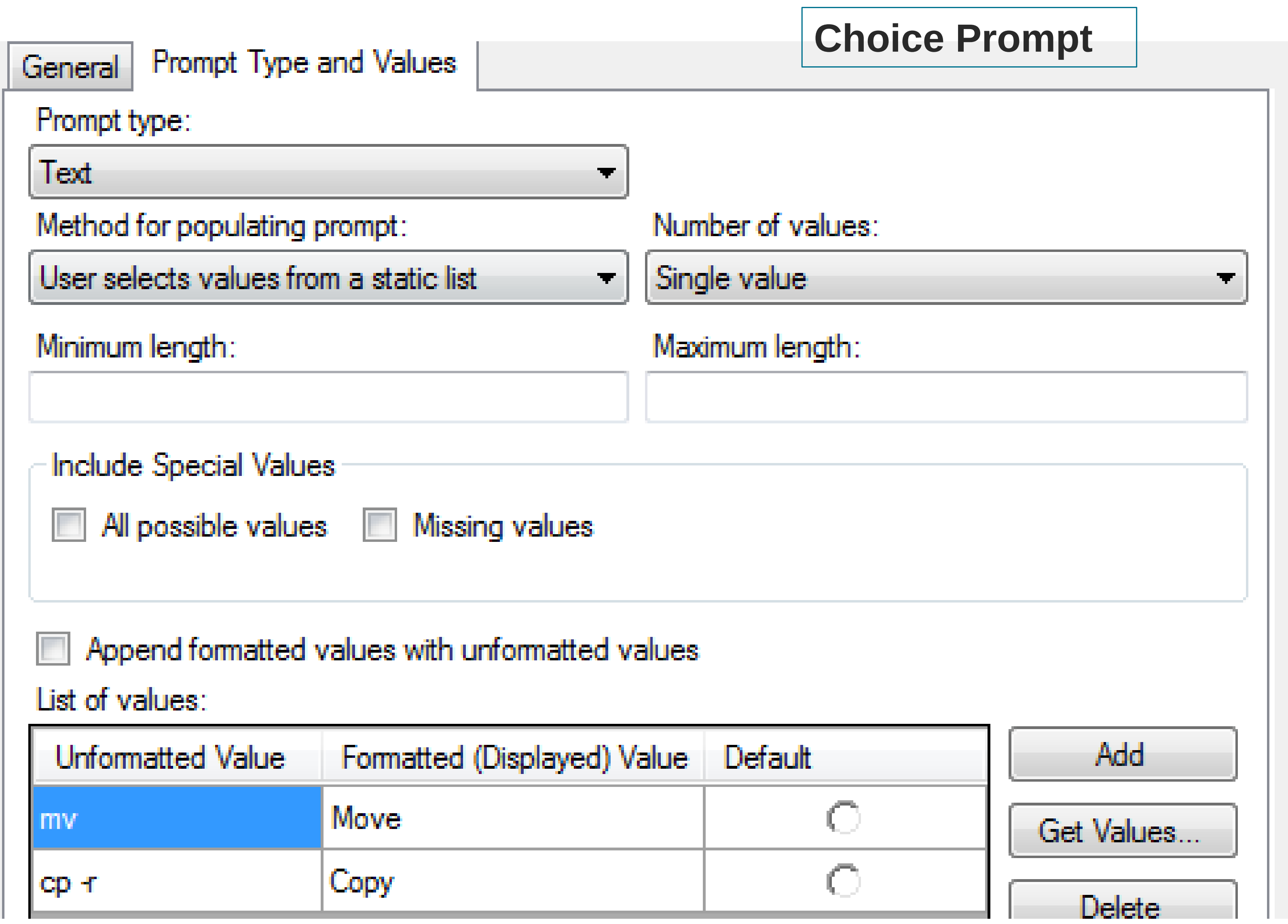
```
Proc Print Data=source_01 noobs label;
Title "Contents of the Source: &Folder.";
Var FileName LastModified FileSize_MBs OwnerPerm GroupPerm WorldPerm;
label LastModified="Last Date Modified"
        FileSize_MBs = "File Size in MBs"
        OwnerPerm = "Owner Permissions"
        GroupPerm = "Group Permissions"
        WorldPerm = "World Permissions"
        ;
footnote "Permissions are in Read Write Execute format.";
footnote2 "The 'd' in Owner indicates if the file is a folder (directory) or not";
Run;
```

Utilizing the Stored Procedure

Combining SAS® functions, prompts and log output. *The example below uses SAS® in an AIX UNIX server environment:*

1. Use Choice Prompt, file prompt and folder prompt
2. Use system macro to convert file prompt choice to a folder view
3. Run your command
4. Print your Server Log (*slide 5*)
5. Clean up server logs

1


Choice Prompt

General | Prompt Type and Values

Prompt type: Text

Method for populating prompt: User selects values from a static list

Number of values: Single value

Minimum length:

Maximum length:

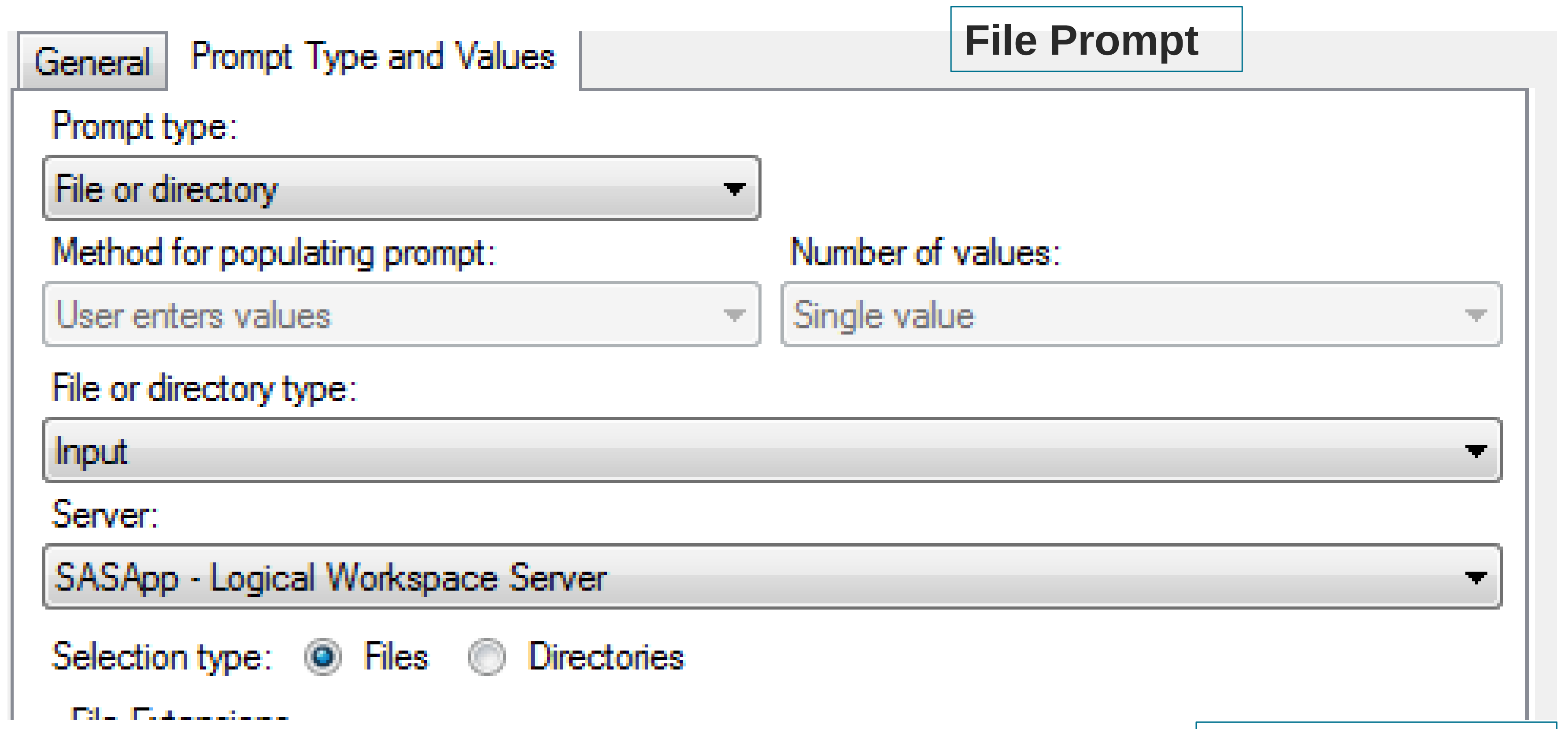
Include Special Values

□ All possible values □ Missing values

□ Append formatted values with unformatted values

List of values:

Unformatted Value	Formatted (Displayed) Value	Default
mv	Move	○
cp -r	Copy	○


File Prompt

General | Prompt Type and Values

Prompt type: File or directory

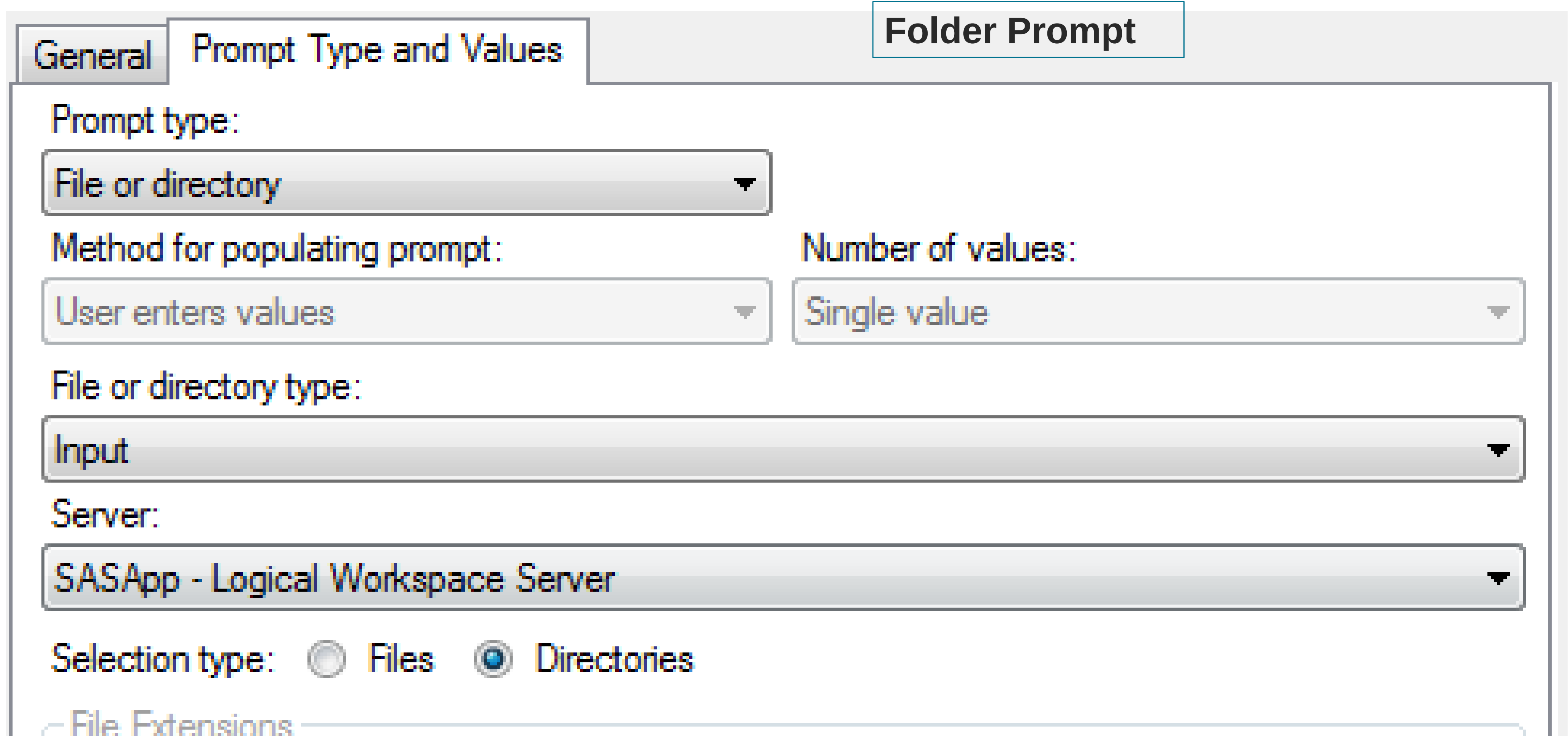
Method for populating prompt: User enters values

Number of values: Single value

File or directory type: Input

Server: SASApp - Logical Workspace Server

Selection type: ● Files ○ Directories


Folder Prompt

General | Prompt Type and Values

Prompt type: File or directory

Method for populating prompt: User enters values

Number of values: Single value

File or directory type: Input

Server: SASApp - Logical Workspace Server

Selection type: ○ Files ● Directories

File Extensions

2

```
/*trims folder name out of address*/
%let sl1 = %trim(%sysfunc(findc("&MC_File.",/,B)));
%let sl2 = %sysevalf(&sl1 - 2);

%let Folder = %trim(%substr(&MC_File.,1,&sl2)) ;
%put &Folder;
```

3

```
/*uses prompts to move/copy folder*/
x "cd /";
Systask COMMAND "&MC_Choice. &MC_File. &MC_Dest. " wait status=Status_Opt;

/* Displays the Home folder contents after move */
x "cd &Folder. ";
x "ls -go > &MC_Dest./Source.txt"  ;
RUN;

/* Displays the destination folder contents after move */
x "cd &MC_Dest. ";
x "ls -go > &MC_Dest./Target.txt";
RUN;
```

5

```
/*Deletes log files that are used for Results*/
filename dlt1 ("&MC_Dest./Source.txt");
filename dlt2 ("&MC_Dest./Target.txt");

data _null_;
        rc = fdelete('dlt1');
        rc = fdelete('dlt2');
run;

filename dlt1 clear;
filename dlt2 clear;
```

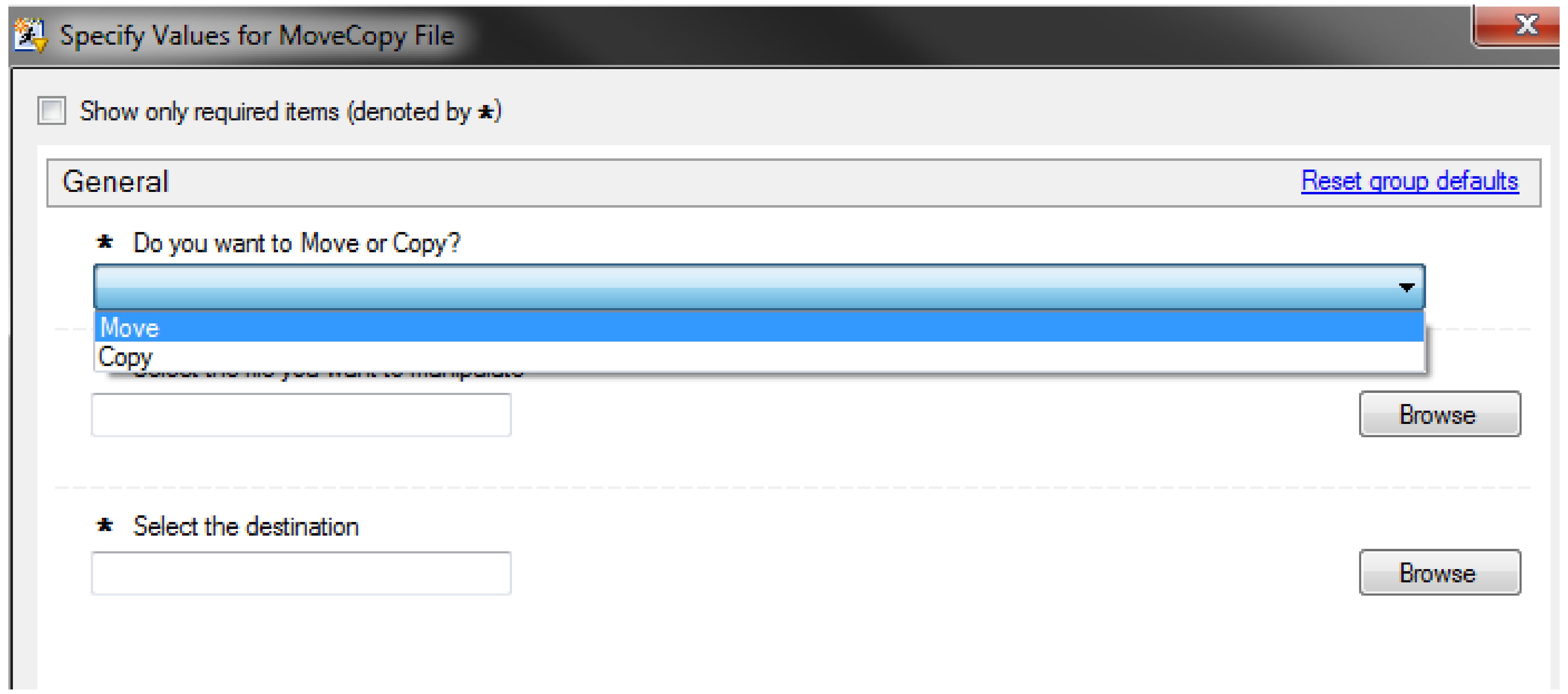
Example of User Interface

Combining SAS® functions, prompts and log output. *The example below uses SAS® in an AIX UNIX server environment:*

1. This is an example of what the end user would see in SAS®. *Combination of the Prompts from slide 6*
2. Final results tab of the stored procedure shown in step 1

Prompt Based Interface

1



Specify Values for MoveCopy File

□ Show only required items (denoted by *)

General — Reset group defaults

* Do you want to Move or Copy?

Move
Copy

Select the file you want to manipulate

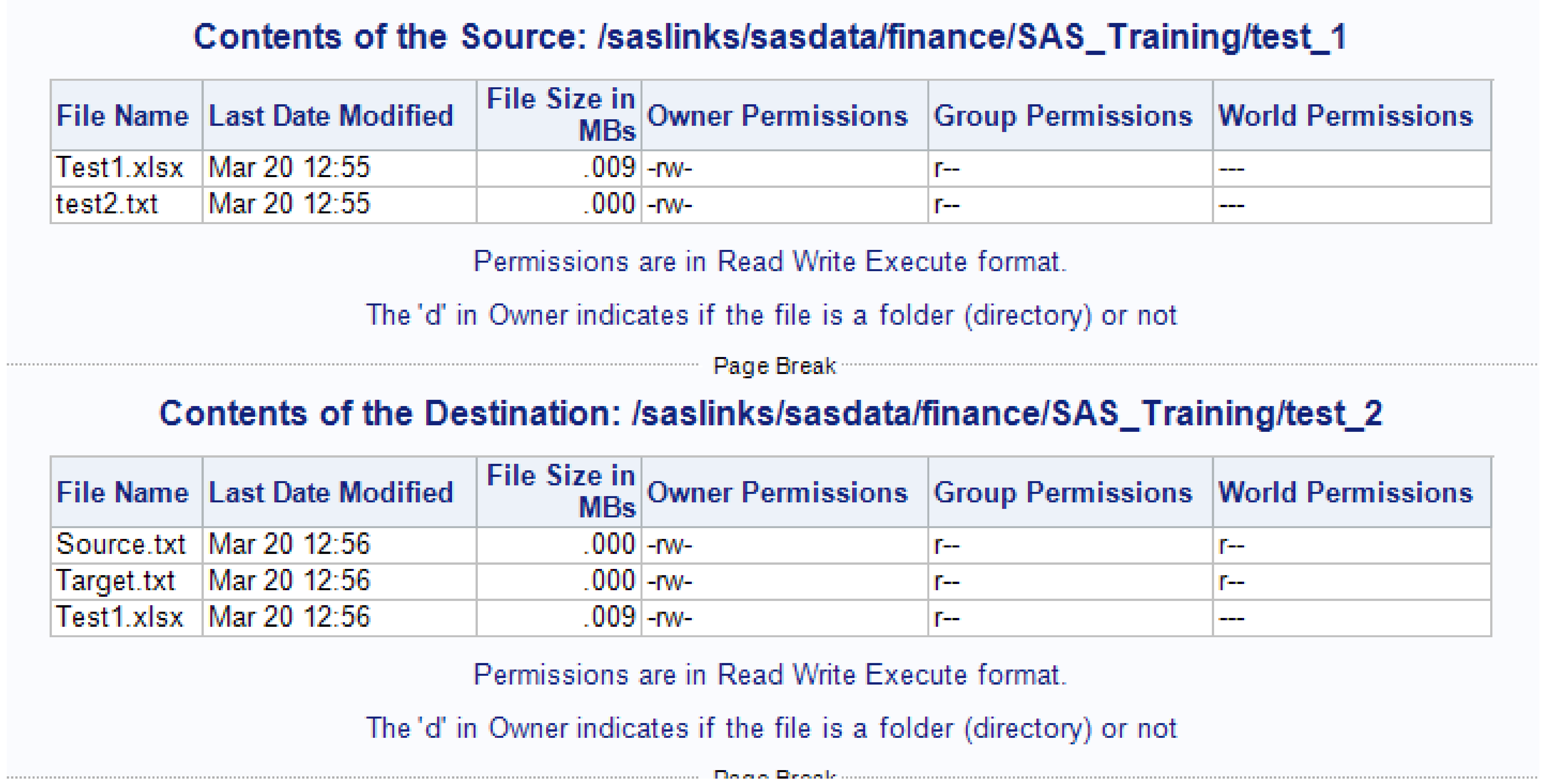
Browse

* Select the destination

Browse

Stored Procedure Output

2



Contents of the Source: /saslinks/sasdata/finance/SAS_Training/test_1

File Name	Last Date Modified	File Size in MBs	Owner Permissions	Group Permissions	World Permissions
Test1.xlsx	Mar 20 12:55	.009	-rw-	r--	---
test2.txt	Mar 20 12:55	.000	-rw-	r--	---

Permissions are in Read Write Execute format.

The 'd' in Owner indicates if the file is a folder (directory) or not

Page Break

Contents of the Destination: /saslinks/sasdata/finance/SAS_Training/test_2

File Name	Last Date Modified	File Size in MBs	Owner Permissions	Group Permissions	World Permissions
Source.txt	Mar 20 12:56	.000	-rw-	r--	r--
Target.txt	Mar 20 12:56	.000	-rw-	r--	r--
Test1.xlsx	Mar 20 12:56	.009	-rw-	r--	---

Permissions are in Read Write Execute format.

The 'd' in Owner indicates if the file is a folder (directory) or not

Page Break

SAS® GLOBAL FORUM

April 26-29
Dallas, TX