Automated Installation of SAS® on Multiple Operating Systems
Peter Slintak, CompuStyle, Ann Arbor, Michigan, USA

Abstract
This paper outlines the procedures to optimize the installation of the SAS System in a corporate environment. Once in place, the installation procedure takes just a click of the mouse button, installs in a fraction of the time of a normal installation, takes up less room on the hard disk, takes less network bandwidth to load, and has the exact configuration based on your corporate optimizations. The redirected installation process is 95% faster than a floppy disk installation and 85% faster than a CD-ROM installation. The size of SAS was reduced by 72% on the file server through the use of compression. This technique can be used in any operating environment. Response files can be invoked for Windows®, OS/2®, DOS®, or Macintosh® operating systems. The paper will go into details of optimizing a configuration, creating a customized session of SAS, writing a response file (with an example written in REXX®), compressing SAS on a file server, and different ways of invoking an installation.

The automated installation of SAS on multiple operating systems from a network is chosen when you want to run SAS from a client computer. The benefits include standardized configuration across the network, faster installation, less network traffic, and simplified installation by clicking on an icon or logging onto the server. The only drawback is writing a customized script. I include an example of one written in REXX because it's similar to pseudo code.

Requirements
This type of installation requires a LAN file server with 50 megabytes of free disk space, a client workstation that's connected to the LAN with enough resources to load SAS, and a file compression utility.

Customizing SAS
The first step is to install SAS using the SASLOAD program. Add any modules that you want to include to the SAS/CORE® installation. Be sure you're familiar with all of the modules you are loading, otherwise you might be installing device drivers, datasets, or modules that will never be used in your organization. Modify the CONFIG.SAS and AUTOEXEC.SAS files for your specific environment. Start SAS and modify your workspace environment. This includes the forms, keys, menus, options, fonts, colors, and preferences. Double check that the installation is robust by testing it thoroughly. Save your settings and exit SAS. Now you have created a template installation of SAS that can be used across the network.

Compressing SAS
The second step is to make a copy of your SAS files on the network. By using compression to save to the network, you decrease the storage size of SAS, the time it takes to copy, and you reduce network traffic. An example of using PKZIP™ is:

\texttt{pkzip -ex \-rp x:\apps\sas\sas.zip c:\sas\*.} *

where \texttt{-ex} uses maximum compression, \texttt{-rp} recurses subdirectories, \texttt{x:...} is the destination of the compressed file \texttt{sas.zip} on the server, and \texttt{c:\...} is the source subdirectory of SAS on the client machine.

Script File
The third step is to create a script file. The script file depends on your operating system environment. The script decompresses the network copy of SAS and copies it to a local drive. It then creates the SAS icons, the working directories, and the profiles. An example of an OS/2 REXX script follows:

\begin{verbatim}
\********************************************************************\
\/* SAS.CMD */
\/* REXXX program to create SAS */
\/* sessions and decompress SAS */
\/* sessions from network */
\/* */
\/* Purpose: Automate the installation */
\/* of SAS, creation of icons, and */
\/* working directories. */
\/* */
\/* Usage: SAS<number of sessions> */
\********************************************************************\
call rxfuncadd
'sysloadfuns', 'executil', 'sysloadfuns'
call sysloadfuns
\end{verbatim}

\begin{verbatim}
/* Pass argument numsessions from numsessions*/
parse arg numsessions
/* Check to see if number was passed */
\end{verbatim}
if numsessions="" then
  say "USAGE: SAS <Number of Sessions>"
else do
  say numsessions
  /* sasdrv can be hard-coded to a specific*/
  /* drive, program assumes a previous release*/
  /* of SAS is installed on the client. */
  sasdrv=findsas()
  say "SAS is on>" sasdrv
  'copy x:\apps\sas\*.sas |sasdrv||'\sas'
  sasname="SESSION"
  do count=1 to numsessions
    /* parse name and sasuser with number*/
    /* appended */
    outname=sasname||count
    sasuser="sasuser"||count
    call sysmkdir sasdrv||'\sas'||sasuser
    say outname
  end
  /* create the icon on the desktop */
  CALL SysCreateObject
    wpProgram","outname,"<WP_DESKTOP>","
    'EXENAME='||sasdrv||'\SAS\SAS.EXE;"||
    'PARAMETERS="%*" -CONFIG
    ||sasdrv||'\SAS\CONFIG.SAS'"||
    'AUTOEXEC
    ||sasdrv||'\SAS\AUTOEXEC.SAS;'||
    STARUPDIR=''||sasdrv||'\SAS'||sasuser||'
  end
end
/* uncompress SAS to the local drive, -d keeps*/
/* directory structure. */
x:\apps\sas\pkuxzip -d x:\apps\sas\sas.zip
  ||sasdrv||'\sas\*'
findsas: procedure
/* This procedure checks for SAS */
/* installed on other disk partitions*/
/* and returns the variable sasdrv. */
arg sasdrv
  CALL SysFileTree
    'C:\SAS\SAS.EXE','SASONC','FO'
  IF SASONC.0>0 then sasdrv='C:'
  else do
    /* Check other disk partitions. */
    CALL SysFileTree
      'D:\SAS\SAS.EXE','SASOND','FO'
    if SASOND.0>0 then sasdrv='D:'
  else say "SAS wasn't found"
  end
  return sasdrv

Executing the Script
The final step is to execute the script. Some possible ways to load the script are by network login automation, e-mail distribution, or command line script submission.

Network Login Automation
A script can be added to a person's network login, and have it run after they login. The installation is transparent to the person logging in, but it uses the system resources while the script is running. Possible modifications to the script include allowing installation at another time, only allowing installation during non-peak hours, or remote administration installations.

E-mail Distribution
A script can be sent by email by the systems administrator. This gives the person at the client machine more flexibility on when to upgrade their workstation.

Command Line
The script can be executed from a command line, if given the path of the script on the network. Notice that you can specify the number of SAS sessions this way. This example makes six SAS sessions:
  x:\apps\sas\sas.exe 6
Multiple sessions of SAS depend on the multi-tasking ability of the operating system. For example, it would not make sense to create multiple SAS sessions in DOS.

SETINIT.SAS Automation
Scripts can also be useful to update the SETINIT.SAS file before it expires. By using scripts in combination with network login automation, it eliminates the manual process of submitting a new SETINIT.SAS.

Conclusion
The redirected installation of SAS from a network can save a network administrator hours of mundane configuration, optimization, and standardization. With careful planning, any upgrades of SAS can be performed in a fraction of the time a conventional installation takes.
SAS and SAS/CORE are registered trademarks or trademarks of SAS Institute Inc. in the USA and other countries. ® indicates USA registration.

OS/2 and REXX are registered trademarks or trademarks of International Business Machines, Inc. in the USA and other countries. ® indicates USA registration.

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

CompuStyle
Peter Slintak
3420 Deward
Ann Arbor, MI 48105
313.665.5368