



Using Script Files

<i>Introduction</i>	201
<i>When to Use a Script</i>	201
<i>Basic Script Functions</i>	202
<i>Using a Script to Start and Stop SAS/CONNECT</i>	202
<i>General Script Statement Rules</i>	203
<i>Syntax</i>	203
<i>Writing Simple Scripts for Signing On and Signing Off</i>	204
<i>Script for EHLLAPI Connections</i>	204
<i>Script for TCP/IP Connections</i>	207
<i>Debugging a Script</i>	211

Introduction

A *script* is a SAS program stored in a file on the local host. However, the programming statements in a script are not the usual SAS programming statements. Instead, scripts use a specialized set of SAS statements called *script statements*. Scripts are executed to initiate or terminate SAS/CONNECT sessions. Scripts that initiate the connection are executed by submitting the SIGNON statement, and scripts that terminate the connection are executed by submitting the SIGNOFF statement. In most cases, the same script is used to sign on and sign off.

When to Use a Script

How do you know if you need to write or modify a script? Some SAS/CONNECT access methods, such as NetBIOS, DECnet, and in most cases APPC, do not use a script to start and stop a connection. The need for a script file when using the TCP/IP access method depends on whether you are connecting to a spawner that runs on a remote host and how that spawner was invoked. The TELNET and EHLLAPI access methods always require the use of script files.

See *Communications Access Methods for SAS/CONNECT and SAS/SHARE Software* for information about the various access methods and to determine if the access method that you use requires a script. The access methods documentation provides examples of script files. Your site may also have sample scripts available from your system administrator.

If the available sample scripts do not meet your purposes, you can write your own script. If you do need to write or modify a script, review the examples in this chapter, and see Chapter 28, “Script Statements,” on page 247 for descriptions of the script statements that are used in the examples.

Basic Script Functions

A script can be a simple, short program or a long, complex program depending on what you want the script to do. All scripts must do at least three things:

- 1 invoke the SAS System on the remote host (by using the SAS command).
- 2 set the appropriate communications options for the remote SAS session in the SAS command. On the remote host, the script sets the COMAMID= and DMR system options.
- 3 determine when the remote SAS session is ready for communications with the local SAS session. In most cases, the script waits for messages from the remote SAS session.

Most scripts also perform a number of other optional functions. For example, they may

- issue the remote host logon command and prompt the user for a userid and a password.
- issue informative messages to the user about whether script execution is proceeding successfully.
- combine both SIGNON and SIGNOFF functions.
- conditionally execute labeled portions of the script so that one script can accommodate multiple types of connections (for example, TCP/IP connections to both a spawner and a TELNET daemon).
- issue remote host commands, such as commands that set session features or define remote host files.
- define any response that is expected from the remote host.
- conditionally execute script subroutines to handle successful operations and error conditions.

The sample scripts from SAS Institute perform all of these functions. By studying their contents, you can learn how to use script statements to do these things.

Note: Scripts that log on to the remote host include information that is specific to the computing installation. They may need minor modifications to work with your logon sequence. △

Using a Script to Start and Stop SAS/CONNECT

This section shows you how to start and stop SAS/CONNECT sessions by using the sample scripts that are supplied by SAS Institute. You will find these scripts in the following directories where the SAS System is installed: *

Windows	!sasroot\CONNECT\SASLINK
OS/2	!sasroot\CONNECT\SASLINK
OS/390	prefix.CTMISC
CMS	SASCONNE MACLIB

* The term !sasroot is not actually part of the pathname; it represents the name of the directory where the SAS System is installed at your site.

OpenVMS Alpha	SAS\$ROOT:[TOOLS]
UNIX	!sasroot/misc/connect

All sample scripts start and stop SAS/CONNECT. An automatic logon script prompts you for userid and password input to log on to a remote host. You must log on to the remote system before running a manual logon script.

Most access methods use automatic logon scripts. Only EHELLAPI gives you the choice of manually logging on to the remote host. For easy association of a script and its intended remote system, the naming conventions for most of the sample scripts are based on the access method and the name of the remote operating system. For example, TCPTSO.SCR is the script for a TCP/IP access method and a TSO remote host, and the TELCMS.SCR is the script for a TELNET access method and a CMS remote host.

If you are an experienced user and you already know how to start and stop the connection, you can skip the step-by-step instructions and see “Setting the Autoexec File for SAS/CONNECT” on page 198 and “Starting Multiple SAS/CONNECT Sessions” on page 199.

General Script Statement Rules

To write a script, you need to read about the specific information for each statement in the script. This section contains general rules that apply to some or all script statements. Each statement description in Chapter 28, “Script Statements,” on page 247 indicates the applicable access method. In addition, Table 28.1 on page 247 shows the access method for which each script statement is effective. A script statement that is not valid for your access method is ignored.

Syntax

- All script statements must end with a semicolon.
- Script statements have a free format, which means that there are no spacing or indentation requirements. A statement can be split across several lines, or one line can contain one or more statements. Statement keywords can be in uppercase, lowercase, or mixed case.
- Text strings that are enclosed in quotation marks are case sensitive. For example, if your script defines a text string in a WAITFOR statement, be sure that the uppercase and lowercase letters in the text string exactly match the text string from the remote host.
- Any script statement can include a label specification. The label must be a valid SAS name, with a maximum of eight characters. The first character must be a letter or underscore. A label must be followed immediately by a colon (:) and must be defined only one time in the script.
- Some script statements specify a time in seconds. The form of the time specification is

n SECONDS;

where *n* can be any number; this number may include decimal fractions. For example, all of the following time specifications are valid:

0 SECONDS;

0.25 SECONDS;
 1 SECOND;
 3.14 SECONDS;
 SECOND is an alias for SECONDS.

- If a script statement specifies a quoted string, such as a remote host command, you can use either single or double quotes. To embed quotes in script statements, follow the same rules that you use for embedded quotes in SAS statements.

Writing Simple Scripts for Signing On and Signing Off

This section illustrates how to write two simple scripts. The first script signs on and signs off over an EHLLAPI 3270 connection to an OS/390 remote host. This script assumes that the user has already logged on to OS/390 and simply wants to establish a connection by using SAS/CONNECT. The second script is for signing on and signing off over a TCP/IP connection. This script uses UNIX as the remote host.

When you are writing scripts or modifying existing scripts, the WAITFOR and TYPE statements require special attention to detail. To ensure that the script recognizes the expected prompt for each stage of signing on, you must be careful to specify the exact sequence of prompts and responses for the remote host.

The simplest method for determining these is to go to the remote host and manually go through the process that you want to capture in the WAITFOR and TYPE statements. For each display on the remote host, choose a word from that display for the WAITFOR statement. Whatever information you type to respond to a display should be specified in the TYPE statement. Be sure to note all carriage returns or other special keys.

If OS/390 is the remote host and you need to use a TYPE statement that has more than 80 characters in a sign-on script, divide the TYPE statement into two or more TYPE statements. To divide the TYPE statement, insert a hyphen (-) at the division point. The remote OS/390 host interprets the hyphen as the continuation of the TYPE statement from the previous line. For example, to divide the following TYPE statement:

```
type
"sas options ('dmr comamid=pclink')"
enter;
```

change it to:

```
type "sas options ('dmr comamid=-" enter;
type "pclink')" enter;
```

Note: Do *not* add any spaces around the hyphen. Δ

Script for EHLLAPI Connections

Before executing this script, you should have already logged on to the remote system; this script signs on to the remote OS/390 host.

```
/* trace on; */
/* echo on; */
/*****/
/* Copyright (C) 1990 by */
/* SAS Institute Inc., Cary NC */
/* */
/* name: tso.scr */
```

```

/*                                                     */
/* purpose: SAS/CONNECT SIGNON/SIGNOFF                */
/*          script for connecting to an               */
/*          OS/390 (with TSO) host by using           */
/*          the EHLLAPI access method                */
/*                                                     */
/* notes:   The communication parameters may         */
/*          need to be changed for your site        */
/*                                                     */
/* assumes: 1. This script assumes that the         */
/*          remote session is already               */
/*          logged on.                               */
/*                                                     */
/*          2. The command to execute SAS in        */
/*          your remote OS/390 (with TSO)          */
/*          environment is "sas". If                 */
/*          this is incorrect for your              */
/*          site, change the contents of            */
/*          the line that contains:                  */
/*          type "sas ...                            */
/*                                                     */
/* support:  SAS Institute staff                     */
/*****
1 log "NOTE: Script file 'tso.scr'
           entered.";

2 if signoff then goto signoff;

/*****
/* EHLLAPI SIGNON                                     */
/*****

3 waitfor 'READY', 0 seconds: noinit;
log 'NOTE: Starting remote SAS now.';

/*****
/* NOTERMINAL suppresses prompts from               */
/* remote SAS session. NO$SYNTAXCHECK               */
/* prevents remote side from going                  */
/* into syntax checking mode when a                 */
/* syntax error is encountered.                       */
/*****
type "sas options('dmr,comamid=pclink,
noterminal,no$syntaxcheck')" enter;

4 continue:
waitfor 'IN PROGRESS', 20 seconds: waitsas;

5 onok:
log 'NOTE: SAS/CONNECT conversation
           established.';

stop;

```

```

/*****
/*  EHLAPI SIGNOFF                                */
*****/

6
signoff:
  log 'NOTE: SAS/CONNECT conversation
      terminated.';
  log 'NOTE: Remote session left logged on.';
  stop;

/*****
/*  SUBROUTINES                                */
*****/

7
waitsas:
  log 'NOTE: Waiting for startup screen...';
  type EREOF enter;
  goto continue;

/*****
/*  ERROR HANDLING                                */
*****/

8
noinit:
  snapshot;
  log 'ERROR: Did not get remote prompt.
      Remote session not active.';
  log 'NOTE: You must log on to the remote
      session before signing on
      using this script file.';
  abort;

9
nosas:
  snapshot;
  log 'ERROR: Did not get SAS software
      startup messages.';
  abort;

```

- 1 LOG statements write messages to the local host's SAS log providing information about the progress of the SIGNON.
- 2 The IF statement tests for a limited set of conditions and then directs processing to the appropriate statement label. For example, if the SIGNOFF command or statement invokes this script, processing jumps to the SIGNOFF label. See step 6 in this list.
- 3 These statements write the message to the local host's log. The WAITFOR statement looks for the remote host READY prompt and processing goes to step 8 if no prompt is received. When the READY prompt is received, the TYPE statement invokes the SAS System on the remote host. The DMR system option is necessary to invoke a special processing mode for SAS/CONNECT. The COMAMID= system option specifies the access method that is used to establish the connection. The DEVICE= option specifies the GRLINK graphics device driver in case you'll be doing remote graphics processing.

- ④ The WAITFOR statement looks for the message **IN PROGRESS**, which is displayed when a SAS session starts on the remote host. If the message is not found, processing goes to the WAITSAS label in step 7.
- ⑤ This LOG statement prints a message to the log to indicate that you have successfully established a connection.
- ⑥ These statements are executed when step 2 directs processing here. The two notes are written to the local log, and the connection is terminated.
- ⑦ Processing continues from step 4. The note is written to the local log, and processing returns to step 4.
- ⑧ The SNAPSHOT statement captures messages displayed on the remote host and writes them to the local host's log. The log messages can help you determine what has occurred on the remote host that interrupted the sign on. You can save or print the local log to have a record of these messages. The error and note messages are written to the local log, and the script aborts.
- ⑨ The SNAPSHOT statement captures messages displayed on the remote host and writes them to the local host's log. The log messages can help you determine what has occurred on the remote host that interrupted the sign on. You can save or print the local log to have a record of these messages. The error message is written to the local log, and the script aborts.

Script for TCP/IP Connections

```

/* trace on; */
/* echo on; */
/*****
/* Copyright (C) 1990 */
/* by SAS Institute Inc., Cary NC */
/* */
/* name: tcpunix.scr */
/* */
/* purpose: SAS/CONNECT SIGNON/SIGNOFF */
/* script for connecting to any */
/* UNIX host via the TCP/IP access*/
/* method */
/* */
/* notes: 1. This script may need */
/* modifications that account */
/* for the local flavor of */
/* your UNIX environment. The */
/* logon procedure should */
/* mimic the events that you */
/* go through when */
/* "telnet"-ing to the same */
/* UNIX host. */
/* */
/* 2. You must have specified */
/* OPTIONS COMAMID=TCP in the */
/* local SAS session before */
/* using the SIGNON command. */
/* */
/* assumes: 1. The command to execute SAS */

```

```

/*          in your remote (UNIX)          */
/*          environment is "sas". If        */
/*          this is incorrect for your     */
/*          site, change the contents      */
/*          of the line that contains      */
/*          type 'sas ...'                 */
/*                                          */
/* support: SAS Institute staff            */
/*****/

❶ log "NOTE: Script file
      'tcpunix.scr' entered.";

      if not tcp then goto notcp;
❷ if signoff then goto signoff;

/*****/
/*  TCP/IP SIGNON                          */
/*****/

❸ waitfor 'login:', 120 seconds: noinit;

/*****/
/*  UNIX LOGON                              */
/*  LF is required to turn the line        */
/*  around after the login name has       */
/*  been typed. (CR will not do)         */
/*****/

❹ input 'Userid?';
type LF;
❺ waitfor 'Password', 30 seconds : nolog;
input nodisplay 'Password?';
type LF;

unx_log:
/*****/
/* Common prompt characters are $,>,% }    */
/*****/
❻ waitfor '$', '>', '%', '}',
      'Login incorrect'      : nouser,
      'Enter terminal type'  : unx_term,
      30 seconds             : timeout;

log 'NOTE: Logged onto UNIX...
      Starting remote SAS now.';

/*****/
/* Invoke SAS on the remote host.        */
/*****/

❼ type 'sas -dmr -comamid tcp -device
      grlink -noterminal -no\$$syntaxcheck' LF;
❽ waitfor 'SESSION ESTABLISHED',
      90 seconds : nosas;

```



```

9 log 'NOTE: SAS/CONNECT
    conversation established.';
  stop;

  /*****
  /* TCP/IP SIGNOFF */
  *****/
signoff:
10 waitfor '$', '>', '%', '}',
    30 seconds;

  type 'logout' LF;
  log 'NOTE: SAS/CONNECT conversation
    terminated.';
  stop;

  /*****
  /* SUBROUTINES */
  *****/
unix_term:

  /*****
  /* Some UNIX systems want the */
  /* terminal-type. Indicate a basic */
  /* tele-type. */
  *****/
  type 'tty' LF;
  goto unix_log;

  /*****
  /* ERROR ROUTINES */
  *****/
11
timeout:
  log 'ERROR: Timeout waiting for remote
    session response.';
  abort;

nouser:
  log 'ERROR: Unrecognized userid or
    password.';
  abort;

notcp:
  log 'ERROR: Incorrect communications
    access method.';
  log 'NOTE: You must set "OPTIONS
    COMAMID=TCP;" before using
    this script file.';
  abort;

noinit:
  log 'ERROR: Did not understand remote

```

```

        session banner.';

nolog:
  log 'ERROR: Did not receive userid or
      password prompt.';
  abort;

nosas:
  log 'ERROR: Did not get SAS software
      startup messages.';
  abort;

```

- ❶ The LOG statement sends the message that is enclosed in quotation marks to the log file or the log window of the local SAS session. Although it is not necessary to include LOG statements in your script file, the LOG statements keep the user informed about the progress of the connection.
- ❷ The IF/THEN statement detects whether the script was called by the SIGNON command or statement or the SIGNOFF command or statement. When you are signing off, the IF/THEN statement directs script processing to the statement labeled SIGNOFF. See step 10.
- ❸ The WAITFOR statement waits for the remote host's logon prompt and specifies that if that prompt is not received within 120 seconds, the script processing should branch to the statement labeled NOINIT.
- ❹ The INPUT statement displays a window with the text **userid?** to allow the user to enter a remote host logon userid. The TYPE statement sends a line feed to the remote host to enter the userid to the remote host.
- ❺ The WAITFOR statement waits for the remote host's password prompt and branches to the NOLOG label if it is not received within 30 seconds. The INPUT statement that follows the WAITFOR statement displays a window for the user to enter a password. The NODISPLAY option is used so the password is not displayed on the screen as it is entered.
- ❻ The WAITFOR statement waits for one of several common UNIX prompts and branches to various error handles if a prompt is not seen. Verify that the WAITFOR statement in the script looks for the correct prompt for your site.
- ❼ This TYPE statement invokes the SAS System on the remote host. The DMR option is necessary to invoke a special processing mode for SAS/CONNECT. The COMAMID= option specifies the access method that is used to make the connection. The NOTERMINAL system option suppresses the prompts from the remote SAS session. The NOSSYNTAXCHECK option prevents the remote session from going into syntax checking mode when a syntax error is encountered.
- ❽ The phrase **SESSION ESTABLISHED** is displayed when a SAS session is started on the remote host by using the DMR and COMAMID=TCP options. The WAITFOR statement looks for the words **SESSION ESTABLISHED** to be issued by the remote host to know that the connection has been established. If the **SESSION ESTABLISHED** response is received within 90 seconds, processing continues with the next LOG statement. If the **SESSION ESTABLISHED** response does not occur within 90 seconds, the script assumes that the remote SAS session has not started and processing branches to the statement labeled NOSAS.
- ❾ When the connection has been successfully established, you must stop the rest of the script from processing. Without this STOP statement, processing of the remaining statements in the script continues.
- ❿ This section of code is executed when the script is invoked to terminate the connection. The first IF statement (see step 2) sends processing to this section of

the script when the script is invoked by a SIGNOFF command or statement. Note that this section waits for a remote host prompt before typing **LOGOUT** in order to log off the remote host. The script then issues a LOG statement to notify the user that the connection is terminated and stops the connection.

- ⑪ These statements are processed only if the prompts expected in the previous steps are not received. This section of the script issues messages to the local SAS log and then abnormally ends (from the ABORT statement) the processing of the script as well as the sign on.

Debugging a Script

When writing scripts, you can take advantage of programming techniques to simplify debugging a new or a modified script. These techniques are outlined in this section.

- For TCP/IP connections, the ECHO statement causes remote host messages to be displayed while a WAITFOR statement executes. This enables you to monitor activity on the remote host during the WAITFOR pause.
- For EHLLAPI connections, the SNAPSHOT statement sends all messages on the current 3270 display to the local LOG window or file so you can see if any error messages or conditions occurred in the remote session.
- The TRACE statement enables you to specify that some or all script statements be displayed as the script executes. This capability can help you isolate the source of a script problem. The TRACE statement can be used with any type of connection.

The correct bibliographic citation for this manual is as follows: SAS Institute Inc., *SAS/CONNECT User's Guide, Version 8*, Cary, NC: SAS Institute Inc., 1999. pp. 537.

SAS/CONNECT User's Guide, Version 8

Copyright © 1999 by SAS Institute Inc., Cary, NC, USA.

ISBN 1-58025-477-2

All rights reserved. Printed in the United States of America. No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, electronic, mechanical, photocopying, or otherwise, without the prior written permission of the publisher, SAS Institute Inc.

U.S. Government Restricted Rights Notice. Use, duplication, or disclosure of the software by the government is subject to restrictions as set forth in FAR 52.227-19 Commercial Computer Software-Restricted Rights (June 1987).

SAS Institute Inc., SAS Campus Drive, Cary, North Carolina 27513.

1st printing, September 1999

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.

IBM®, AIX®, DB2®, OS/2®, OS/390®, RS/6000®, System/370™, and System/390® are registered trademarks or trademarks of International Business Machines Corporation. ORACLE® is a registered trademark or trademark of Oracle Corporation. ® indicates USA registration.

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

The Institute is a private company devoted to the support and further development of its software and related services.