# Creating HCON Autolog Script Files

## Special Notices

Information in this document is correct to the best of our knowledge at the time of this writing. Please send feedback by fax to "AIXServ Information" at (512) 823-4009.

Please use this information with care. IBM will not be responsible for damages of any kind resulting from its use. The use of this information is the sole responsibility of the customer and depends on the customer's ability to evaluate and integrate this information into the customer's operational environment.

> **Note:** The information in this document has **not** been verified for AIX 4.1.

## About this Document

This document contains tips for creating an HCON autolog script file with AIX on a RISC System/6000. You may also wish to refer to the sample autolog script files in /usr/lib/hcon/SYSvm1 and /usr/lib/hcon/tso.

Several fixes are mentioned in this document. Fixes (PTFs) for AIX 3.2.5 and later can be downloaded via Internet with the FixDist service. For FixDist information and instructions, order fax number 1228 from 1-800-IBM-4FAX. If you don't have access to Internet or if your machine is at a pre-3.2.5 AIX level, you can request a specific fix number from 1-800-CALL-AIX. All other services from 1-800-CALL-AIX which are not Program Services require a software support contract.

## Notes

- If APAR IX43662 is installed on the system, the autolog trace function will create a file of additional debug information. This file will be either in the home directory or in /tmp. It will be named SYSxxx.trace where SYSxxx is the autolog file used.

## Creating Autolog Script Files

You can start creating an HCON autolog script file with either the genprof or hconutil command. In either case, the following screen is displayed:

```
 NODE ID : ________________
 LOGON  STARTS: __         LOGOFF  STARTS: __

EVENT      PROMPT               INPUT            KEY       NEXT EVENTS
 1   ________________ ________________________ ______  __ __ __ __ __ __
 2   ________________ ________________________ ______  __ __ __ __ __ __
 3   ________________ ________________________ ______  __ __ __ __ __ __
 4   ________________ ________________________ ______  __ __ __ __ __ __
 5   ________________ ________________________ ______  __ __ __ __ __ __
 .   ________________ ________________________ ______  __ __ __ __ __ __
 .   ________________ ________________________ ______  __ __ __ __ __ __
28   ________________ ________________________ ______  __ __ __ __ __ __
29   ________________ ________________________ ______  __ __ __ __ __ __
30   ________________ ________________________ ______  __ __ __ __ __ __
```

The fields on the above screen are:

**NODE ID** This is the name of the autolog script. Autolog scripts are typically named according to the system being accessed. The script file will be stored in the user's home directory in the format "$HOME/SYS<NODE_ID>" (for example "/u/mark/SYScics2a"). When autolog is used, HCON will first check the user's $HOME directory. If HCON does not find the autolog NODE ID (script name) there, it will check the /usr/lib/hcon directory. If it does not find it there, an error will occur.

**LOGON STARTS and LOGOFF STARTS** These fields indicate the line numbers at which the logon and logoff sequences begin. In most cases, LOGON STARTS is 1. LOGOFF STARTS is determined after completing the LOGON portion.

**PROMPT** This is the sequence that autolog will key on in the incoming data stream. It works in the same manner as the g32_search and HLLAPI HA_SEARCH_PS routines.

Valid values, in addition to actual screen data, are "?" (for any character), "*" (which is a wildcard), and ":" (used to separate fields of the autolog script). If the user wants to search specifically for any of these special characters, the character must be preceded by two backslashes. So, if the desired key sequence is "***", as for TSO data streams, the PROMPT must contain "***" (backslash backslash asterisk backslash backslash asterisk backslash backslash asterisk). Having **two** backslashes lets AUTOLOG know to take the next character as is.

When choosing a PROMPT, make sure that it is unique and that it is from the bottom of the screen but NOT from the OIA (operator information area). This is because, if you are using SNA (either SNA DFT or SNA SERVICES), the host will often send a screen in segments. It is always best to have the PROMPT in the LAST segment so that

no data collision occurs. This will prevent HCON from sending data at the same time the Host is sending another segment. Once AUTOLOG detects the PROMPT in the data stream, it goes on to process the INPUT and KEY fields.

**INPUT** This is the data that will be sent upon a successful find of the PROMPT.

**KEY** This is the key that will be sent after the INPUT upon successful find of the PROMPT. The format is the KEY followed in parentheses by the number of times the key is sent. Sending the Enter key twice, "ENT(2)", can cause problems due to the speed of the Host.

Valid KEYs are ENT (ENTER), CLE (CLEAR), PA1, PA2, PF1-PF9, TAB NEW (newline), CUP (cursor up), CDN (cursor down), CRT (cursor right), CLT (cursor left), REQ (sys request), ATN (attention), and RES (reset).

**NEXT EVENTS** These columns indicate which event should be executed after the current event is completed.

If the PROMPT was successfully found for the current event, then the first NEXT EVENTS column is used.

If the PROMPT was not successfully found for the current event, then the other columns are used, starting with the second column for the first failure, the third column for the second failure, etc.

The following numbers in the NEXT EVENTS columns have special meaning:

**0** Indicates successful completion of the LOGON or LOGOFF. This number may be in any column.

**-1** Indicates that the event failed and execution should return to the previous event. This number may be in any of the failure columns.

**-2** Indicates that the event failed and that the autolog program should be exited, with the current event number returned as an error. This may be in any of the failure columns.

## Example Autolog Script File

The following is an example autolog script file, first shown as a raw file and then shown as seen through "genprof".

```
::::1:9
AUSTIN VM1:vm1:ENT(1):2:1:14:-2
CP READ &id:ENT(1):4:2:2:2:2:3
CP READ &id:ENT(1):4:3:3:3:3:-1
new password:&pw:ENT(1):5:4:4:4:4:-2
HOLDING::CLE(1):6:5:5:5:5:-2
VM READ::ENT(1):7:6:6:6:6:15
Ready;::CLE(1):0:7:7:7:7:8
Ready;::CLE(1):0:8:8:8:8
?::CLE(1):10
RUNNING:logoff:ENT(1):11
AUSTIN VM1:::0:11:11:11:11:12
AUSTIN VM1:::0:12:12:12:12:13
AUSTIN VM1:::0:13:13:13:13
?::ENT(1):1
RECONNECTED:i cms:ENT(1):16:-2
VM/XA::ENT(1):7:16:16:-2
```

Figure 1. Example Autolog Script as Raw File

```
 NODE ID : vm1_____________
 LOGON  STARTS: 1_        LOGOFF  STARTS: 9_

EVENT      PROMPT                INPUT            KEY      NEXT EVENTS
 1   AUSTIN VM1_________ vm1_____________________ ENT(1) 2_ 1_ 14 -2 __ __
 2   CP READ____________ l &id____________________ ENT(1) 4_ 2_ 2_ 2_ 2_ 3_
 3   CP READ____________ l &id____________________ ENT(1) 4_ 3_ 3_ 3_ 3_ -1
 4   new password_______ &pw_____________________ ENT(1) 5_ 4_ 4_ 4_ 4_ -2
 5   HOLDING____________ _____________________ CLE(1) 6_ 5_ 5_ 5_ 5_ -2
 6   VM READ____________ _____________________ ENT(1) 7_ 6_ 6_ 6_ 6_ 15
 7   Ready;_____________ _____________________ CLE(1) 0_ 7_ 7_ 7_ 7_ 8_
 8   Ready;_____________ _____________________ CLE(1) 0_ 8_ 8_ 8_ 8_ __
 9   ?__________________ _____________________ CLE(1) 10 __ __ __ __ __
10   RUNNING____________ logoff_______________ ENT(1) 11 __ __ __ __ __
11   AUSTIN VM1_________ _____________________ ______ 0_ 11 11 11 11 12
12   AUSTIN VM1_________ _____________________ ______ 0_ 12 12 12 12 13
13   AUSTIN VM1_________ _____________________ ______ 0_ 13 13 13 13 __
14   ?__________________ _____________________ ENT(1) 1_ __ __ __ __ __
15   RECONNECTED________ ipl cms______________ ENT(1) 16 -1 __ __ __ __
16   VM/XA______________ _____________________ ENT(1) 7_ 16 16 -1 __ __
```

Figure 2. Example Autolog Script as Seen Through genprof

As you may be able to tell from comparing the two figures above, the ":" in the raw file is a field separator and the first line contains the LOGON and LOGOFF START positions.

## Explanation of the Example

In the above example, the script is named "vm1". The LOGON starts on line one and the LOGOFF starts on line nine.

When the emulator first starts, it will search for a PROMPT of "AUSTIN VM1". If that is successfully found, "vm1" will be sent, the ENTER key will be sent, and NEXT EVENT will take us to EVENT 2.

If the PROMPT "AUSTIN VM1" had not been found, the NEXT EVENT would loop us back to EVENT 1 where the PROMPT "AUSTIN VM1" would be looked for again. The loop back to the event allows time for the HCON emulator to come up. Such a loop should also be done when SNA attachments are brought up because the host usually will not send the first screen of data until **all** connections have been established. This in **no way** slows down the AUTOLOG process because AUTOLOG checks the presentation space for a PROMPT each time data is received from the Host. If the PROMPT is found, the AUTOLOG program continues on and does NOT wait the specified AUTOLOG timeout value. The default timeout value is three seconds and can be changed within the HCON session profile ("Autolog step time") or in the logon ID string (uid) via the "time=##" variable.

If the PROMPT is NOT found after the loop, the autolog file will go to EVENT 14, PROMPT on any character, send no INPUT, send the Enter key once, and start all over at EVENT 1. Many times a real 3278/3279 will come up with a blank screen first and this sequence allows for that. Returning to EVENT 1, if the PROMPT "AUSTIN VM1" is not found this time, the NEXT EVENT -2 is invoked which ends the logon.

Lines two and three are the same in order to allow the host time to process the communication link startup. &id and &pw are variables for the Host login ID and the Password. They are placeholders in the AUTOLOG to conceal the identity. There are several ways to feed the user ID and password into the AUTOLOG script (see “Methods of Inputting Autolog Variables” on page 6).

### Notes

Notice that you can send a key without input (as in events 5-9), input without a key, or neither any input or key (as in events 11-13).

Because the input field only accepts 20 characters, input with no key is useful for finishing a very long command. For example, two events could be used for the long command. The first event would contain the PROMPT, the first 20 characters of the INPUT, no KEY, and a NEXT EVENT field for a successful PROMPT of the second event. The second event would contain no PROMPT, the rest of the INPUT, the ENT(1) KEY and then the NEXT EVENT field.

# Methods of Inputting Autolog Variables

## Method 1

The uid and pw values within the g32_open(x) functions can be used by the AUTOLOG application so that **no** prompting for the user occurs. Also, HLLAPI with APAR IX42127 allows the user to use AUTOLOG and the variables hllapi_uid and hllapi_pwd, which are very similar to the g32_open(x) parameters.

If you are using API, the uid will contain the Host logon ID, the AUTOLOG NODE ID and possibly the other AUTOLOG parameters, which are the trace and time variables. For example:

```
uid="robert,vm1,trace,time=10"
```

The Host logon ID is robert, the AUTOLOG node ID (script name) to be used is vm1, trace is on, and the AUTOLOG timeout value is 10 seconds (instead of the default of 3). The trace and time parameters are optional and do not have to be specified.

The trace facility allows for the AUTOLOG sequence to be displayed to the console. The match value indicates if the PROMPT was successfully found or not (0 means not found and 1 means found). The time value will change the AUTOLOG timeout value for finding a PROMPT. Though the timeout value in the example is 10, the performance does not degrade because AUTOLOG will go to the NEXT STEP (event) once the PROMPT is found and will not wait for the timer to end.

The pw value will contain the Host password, pw="p1word" where p1word is the user's password.

## Method 2

You can also define within the session profile the Host logon ID, the Autolog node ID, Autolog trace, and the Autolog step time. Each of these corresponds to each of the values described in “Method 1”:

```
Host logon id                 robert
Autolog node id               vm1
Autolog trace                 yes
Autolog step time (seconds)   10
```

Either the user could be prompted for the password or, if HLLAPI or API is used, the password could be defined as in “Method 1.”

If the user defines the uid and has different values in the HCON Session profile, the values in the uid parameter will be used.

## Method 3

The user could be prompted for the values. This will happen if both the uid and pw parameters are NULL.

After the g32_open(x) command, the fxfer command, or the HLLAPI routines SET_SESSION_PARAMETER of AUTOLOGON and a CONNECT, the user would be prompted for the uid string.

For example:

```
Enter logon id string:
robert,vm1,trace,time=10  (This would be entered by the user)

Enter password:
   (The user would type in the password, but it is not
   echoed on the screen)
```

Remember that the trace and time parameters are optional and do not have to be specified.

## Method 4

The fxfer command allows the user to bypass the "Enter logon id string" by using the -x option. The format for our example would be:

```
-x robert,vm1[,trace,10].
```

APAR IX36987 allows any values not specified by the -x flag to default to those in the HCON session profile.

APAR IX43441 adds a new flag, "-p password", for implicit file transfers. An implicit file transfer is one for which the HCON session has not been started.

You can also feed the parameters into file transfer from a shell script. This is explained in InfoExplorer, "How to Create HCON File Transfer Shell Procedures".

## Reader's Comments

Please fax this form to (512) 823-4009, attention "AIXServ Information". You may also e-mail comments to: elizabet@austin.ibm.com. These comments should include the same customer information requested below.

Use this form to tell us what you think about this document. If you have found errors in it, or if you want to express your opinion about it (such as organization, subject matter, appearance) or make suggestions for improvement, this is the form to use.

If you need technical assistance, contact your local branch office, point of sale, or 1-800-CALL-AIX (for information about support offerings). These services may be billable. Faxes on a variety of subjects may be ordered free of charge from 1-800-IBM-4FAX. Outside the U.S. call 415-855-4329 using a fax machine phone.

When you send comments to IBM, you grant IBM a nonexclusive right to use or distribute your comments in any way it believes appropriate without incurring any obligation to you.

**Note:** If you have a problem report or item number, supplying that number may help us determine why a procedure did or did not work in your specific situation.

Problem Report or Item #: Branch Office or Customer #:

Be sure to print your name and fax number below if you would like a reply:

Name: Fax Number:

END OF DOCUMENT (hcon.autolog.sna, 4FAX# 1432)