OpenVMS Technical Journal V8

What's in your AlphaServer console?

Kostas G. Gavrielidis (GSE-MSE) Master Technologist HP Services

Overview

An HP Services engineer at a customer production environment must find out how a given AlphaServer has booted up and identify some of the console parameter settings.

This article explores how to get to the Alpha console and how to get the settings for the console environment variables from the operating system via implemented console callbacks.

In HP customer production environments, are we very rarely allowed to shutdown or reboot a production system, let alone try to connect a device to the console of the system. Our customers will allow only a secure connection to their production systems via the SSH[1] protocol. This will get us only to the operating system level. So how do we get the environmental variables and their settings from the boot loader[2] without taking the system down?

What is the SRM?

All HP Alpha systems running the OpenVMS, Tru64 UNIX, and the Linux operating systems come with the System Reference Manual (SRM) console subsystem in their firmware, perhaps along with other consoles such as the ARC/AlphaBIOS console. The SRM has control of the system hardware and provides the following features (please see the Alpha Architecture Reference Manual for more details):

- Initializes, tests, and prepares the hardware platform for system software

[1] `ssh` (secure shell) is a program for logging into a remote system and for executing commands on the remote system. It is intended to replace `rlogin` and `rsh`, and provide secure encrypted communications between two un-trusted hosts over an insecure network.

[2] A boot loader or boot manager is a program that loads the image of the operating system to be run on a particular hardware platform. Other boot loaders include the GNU's Grand Unified Boot Loader (GRUB), the Linux Loader (LiLo), the Windows NT loader (NTLDR), the aboot loader, the Alpha miniloader (MILO), Initial System Loader (ISL) for PA-RISC systems, Extensible Firmware Interface (EFI) for Itanium systems.

- Loads in memory and starts the execution of the system software
- Controls and monitors the state changes of all processors
- Provides services to system software that simplify system software control of, and access to platform hardware
- Provides a monitor and control of the system

The Privileged Architecture Library (PALcode) provides a mechanism to implement a range of functions, provided by the SRM, consistently without microcode. These functions range from the binary encoding of the instruction and data to the execution mechanisms and synchronization primitives.

Some of the more useful SRM commands (at the ">>>" prompt), include: `show device`, `show config | more`, `e pc, e sp, e ps, e r26, crash, halt`, `init`, `boot`, `memtest`, `memexer_mp`, `show fru`, `sys_exer`, `test`, and `wwidmgr`.

Here is the SRM command to show all devices.

```
>>> show device
Resetting I/O buses...
ewa0: link up : Negotiated  100BaseTX: full duplex
ewb0: link up : Negotiated  100BaseTX: full duplex
dka0.0.0.14.0              DKA0              COMPAQ BD036745A4  B010
dqb0.0.1.13.0              DQB0             Compaq   CRD-8402B  1.03
dva0.0.0.0.0               DVA0
ewa0.0.0.9.0               EWA0              00-10-64-30-D3-E0
ewb0.0.0.11.0              EWB0              00-10-64-30-D3-DF
pga0.0.0.16.0              PGA0        WWN 1000-0000-c92a-7195
pgb0.0.0.17.0              PGB0        WWN 1000-0000-c92a-71f5
pka0.7.0.14.0              PKA0                  SCSI Bus ID 7
>>>
```

This is the SRM command to show hardware configuration.

```
P00>>> show config | more
                      Compaq Computer Corporation
                    Compaq AlphaServer ES45 Model 2

Firmware
SRM Console:    V6.1-3
PALcode:        OpenVMS PALcode V1.93-37, Tru64 UNIX PALcode V1.88-28
Serial ROM:     V2.18-F
RMC ROM:        V1.0
RMC Flash ROM:  V1.9

Processors
CPU 0           Alpha EV68CB pass 2.4 1000 MHz  8MB Bcache
CPU 1           Alpha EV68CB pass 2.4 1000 MHz  8MB Bcache
CPU 2           Alpha EV68CB pass 2.4 1000 MHz  8MB Bcache
CPU 3           Alpha EV68CB pass 2.4 1000 MHz  8MB Bcache

Core Logic
Cchip           Rev 17
Dchip           Rev 17
PPchip 0        Rev 17
PPchip 1        Rev 17
TIG             Rev 2.6

--More-- (SPACE - next page, ENTER - next line, Q - quit)
```

Here is the commnad for examining the pc, sp, ps and r26 registers.

```
P00>>> e pc
PC psr:                0 (    PC) FFFFFFFF0048C1F0
P00>>> e sp
gpr:                1E (   R30) FFFFFE06863BF3D0
P00>>> e ps
ipr:                17 (    PS) 0000000000001F00
P00>>> e r26
gpr:                1A (   R26) FFFFFFFF004A4660
```

What are the console environment variables?

An environment variable is a name and value association that is maintained by the console program. There are two types of environmental variables: volatile and nonvolatile. The volatile variables are initialized to their default by a system reset, and the nonvolatiles remain as set across system power cycles. Alpha systems have a variety of variables with values set up within the SRM system console. These environment variables control the particular behavior of the console program and the system hardware, the particular console interface presented to the operating system, various default values for the operating system bootstrap, and related control mechanisms. In other words, ''the environment variables provide an easily extensible mechanism for managing complex console state.''

Most users will never see or have a need to use the console system on the Alpha platforms. Technologists who provide proactive consulting and support to the customer production environments, however, must know the settings of certain important console variables such as: `auto_action`, `boot_dev`, `booted_dev`, `boot_file`, `booted_file`, `boot_osflags`, `booted_osflags`, `console`, `os_type`, `pal`, and `sys_serial_num`. The specific number and names of these variables vary from platform to platform and by firmware version.

How to get to the Alpha console

The console prompt varies from ">>>" to "`Pnn>>>`"

where `nn` is the `00` | `01` | `02` | … | `nn` CPU processor.

Several methods are available to get to the console:

- After an orderly shutting down of the system

  On Tru64 UNIX:

  ```
  # /usr/sbin/shutdown –h now "system is going down…"
  ...
  # /usr/sbin/halt
  >>>
  ```

  On OpenVMS:

  ```
  $ @SYS$SYSTEM:SHUTDOWN.COM
  ...
  >>>
  ```

- Pressing the HALT button on the system
- From a direct serial connection to the Alpha console, issue the <Ctrl>P command to get to the console prompt ">>>" (or "`P00>>>`"), as shown in the example below.

```
Welcome to OpenVMS (TM) Alpha Operating System, Version V7.3-2

Username:

<Ctrl>P

halted CPU 0
CPU 1 is not halted
CPU 2 is not halted
CPU 3 is not halted

halt code = 1
operator initiated halt
PC = ffffffff88670438
P00>>>
```

To control the behavior of the system hardware, SRM provides a variety of environment variables as a means of managing the system states. For a complete list of all the console environment variables and their values issue the command "`show *`" from the console prompt "`P00>>>`".

```
P00>>>show *
auto_action             RESTART
boot_dev                dgb2001.1003.0.10.0 dgb2001.1004.0.10.0
dga2001.1001.0.8.0 dga2001.1002.0.8.0
boot_file
boot_osflags            3,0
boot_reset              OFF
bootbios
bootdef_dev             dgb2001.1003.0.10.0 dgb2001.1004.0.10.0
dga2001.1001.0.8.0 dga2001.1002.0.8.0
booted_dev              dgb2001.1004.0.10.0
booted_file
booted_osflags          3,0
char_set                0
com1_baud               9600
com1_flow               SOFTWARE
com1_mode               THROUGH
com1_modem              OFF
com2_baud               9600
com2_flow               SOFTWARE
com2_modem              OFF
console                 serial
controlp                ON
...
pal                     OpenVMS PALcode V1.98-43, Tru64 UNIX PALcode V1.92-33
...
sys_serial_num          4228KSTZA000
tt_allow_login          1
tty_dev                 0
version                 V6.9-2 Nov 18 2004 09:57:20
wwid0                   2001 1 WWID:01000010:6005-08b4-0001-278f-0000-7000-
052f-0000
wwid1
wwid2
wwid3
P00>>>
```

One of the tools that allows console access among other things is ConsoleWorks[3] by TECSys Development Inc. (TDi). Figure 1 shows HTTP access to ConsoleWorks. The port used by ConsoleWorks is 5176, and normally the path to it will be similar to the URL: http://hostname.domain.com:5176/.

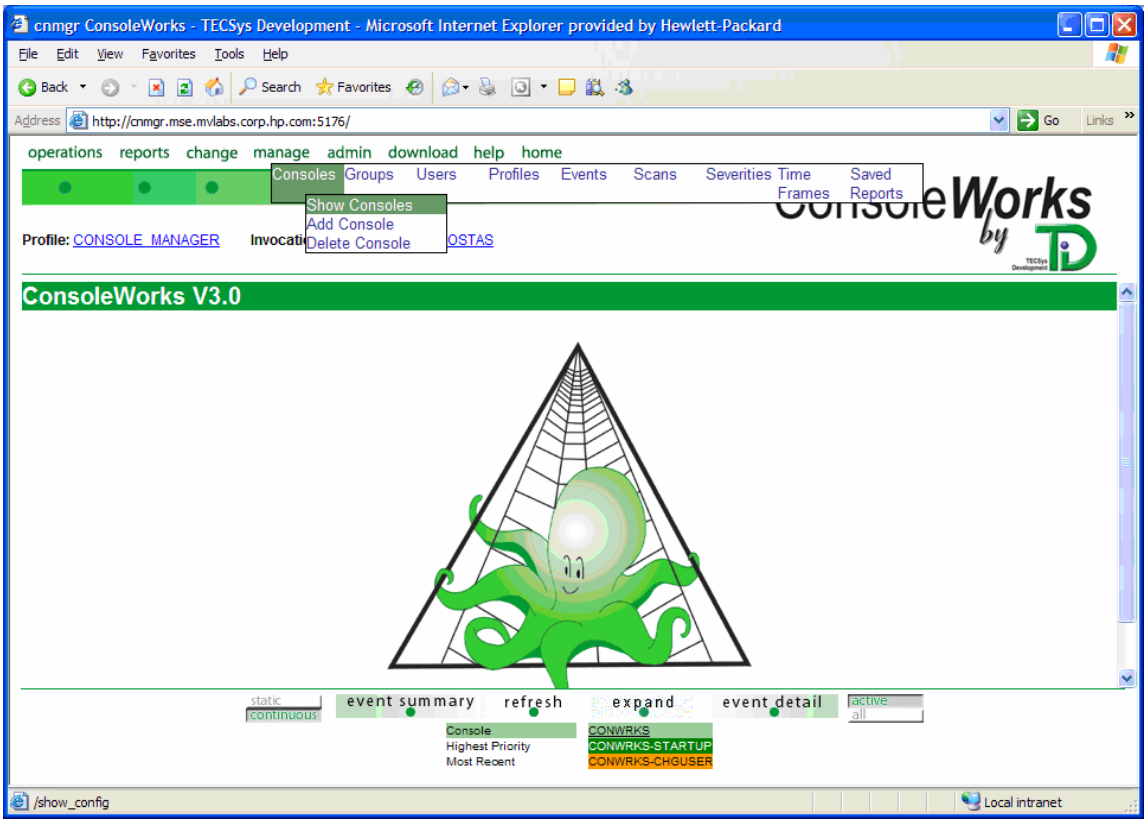


Figure 1 - ConsoleWorks Web interface

Once connected, choose the top menu option *manage* → *Consoles* → *Show Consoles* to get a list of all configured and available consoles to connect to. The resulting screen will be similar to Figure 2 below.

[3] ConsoleWorks is developed by TECSys Development Inc. (TDi), and it is a Web-based enterprise event monitoring, event management and regulatory compliance software solution. It provides secure remote monitoring and management for enterprise networks, server, devices and applications. Additionally captures, audits and logs console data from these devices. For more details see the URL: http://www.tditx.com/consoleworks.html

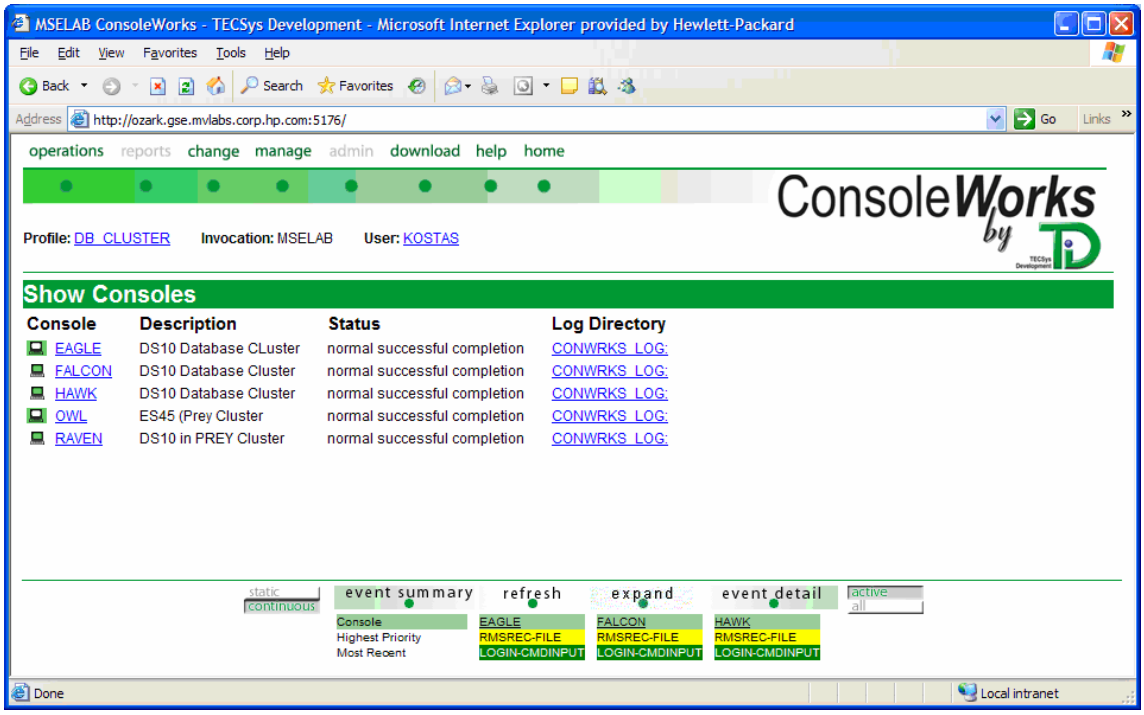


Figure 2 - Available consoles

HP Tru64 UNIX Alpha Console

In HP Tru64 UNIX, two system functions, `getsysinfo(2)` and `setsysinfo(2)`, are available for use by C programs to get and set the console environment variable values. The `getsysinfo(2)` function with the `GSI_PROM_ENV` operation can be used to obtain the current value of the PROM console environment variables. The following small C program shown below – `getcvar.c` – illustrates this capability.

```
msemrac1.mse.mvlabs.corp.hp.com> cat getcvar.c
   /*
    * Module: getcvar.c  Version: X0.0-000 January 31st, 1996 /KGG
    */
   #include <sys/types.h>
   #include <machine/hal_sysinfo.h>
   main(int argc, char **argv)
   {
           char env[132];
           int ret;
           if (argc == 1)  {
                   puts("Usage: getcvar string");
                   exit(1);
           }
           if ((ret = getsysinfo(GSI_PROM_ENV, env, 132, 0, argv[1])) == -1) {
                   perror("getsysinfo");
                   exit(1);
           }
           printf("%s: %s\n", argv[1], env);
           exit(0);
    }
```

For more information on both the `getsysinfo()` and `setsysinfo()` functions please refer to the man pages.

Makefile for the getcvar C program

```
msemrac1.mse.mvlabs.corp.hp.com> cat getcvar.Makefile
#++
# Module: getcvar.Makefile
# Version: X0.0-000 January 31st, 1996 /KGG
#--
OBJS = getcvar.o
getcvar: $(OBJS)
        cc $(OBJS) -lm -o getcvar
        ls -l
        getcvar boot_reset
        getcvar boot_osflags
        getcvar bootdef_dev
        getcvar boot_file
        getcvar auto_action
getcvar.o:
        cc -c getcvar.c
```

Compiling and running the getcvar C program

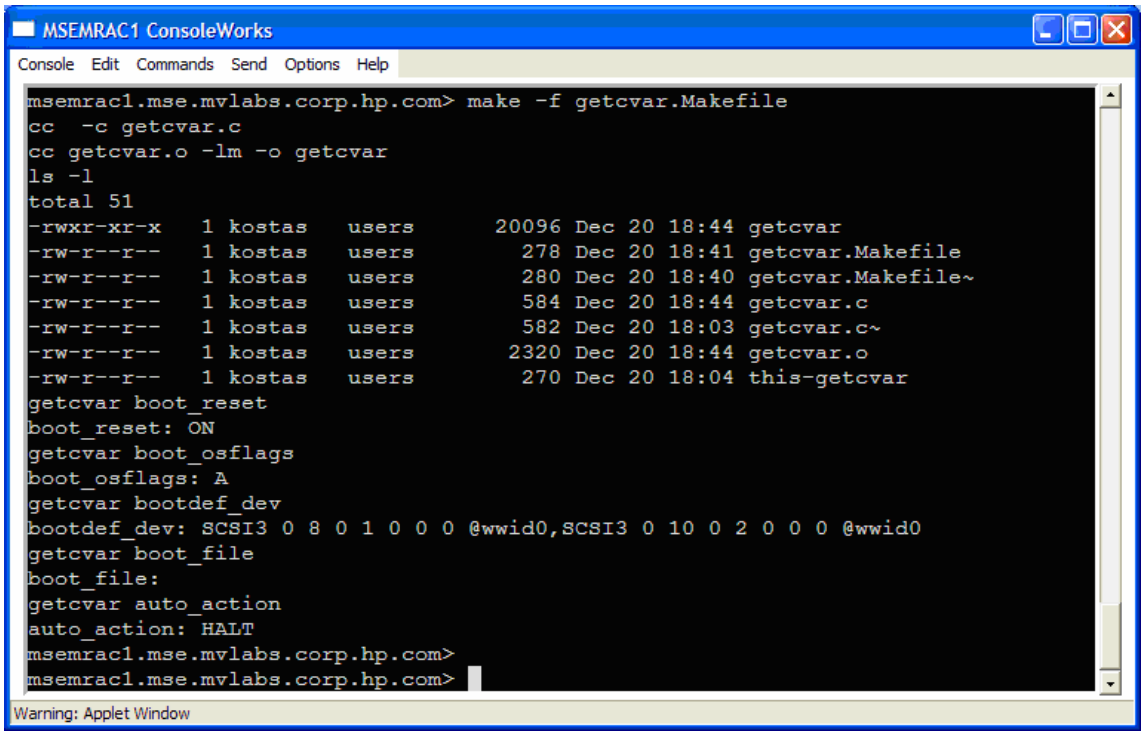


Figure 3 - Compiling and running the getcvar.c program

As with the `getcvar` program, one can write a `setcvar` program that would take a variable name and variable value pair and set it, as in this example:

```
# setcvar boot_reset OFF
```

And in general:

```
# setcvar <console_var_name> <console_var_value>
```

by making the following modifications to the `getcvar` program.

Include the <`machine/prom.h`> header file in addition to the others:

```
#include <machine/prom.h>
```

Define two string variables, `env_name` and `env_value`, and assign the environment variable name and its value to each one, respectively, from the command line:

```
char *my_name;
char *env_name, *env_value;

my_name = argv[0];
env_name = argv[1];
env_value = argv[2];
```

Call the system function `setsysinfo()` with the correct parameters:

```
if ((ret = setsysinfo(SSI_PROM_ENV, env_value, strlen(env_value)+1,
                      env_name, PROM_CONVERT_TYPE)) == -1) {
   fprintf(stderr,
         "%s: setsysinfo() failed setting %s to '%s': %s\n",
         my_name,
         env_name, env_value,
         strerror(errno));
   return(-1);
}
```

For assistance with writing your version of the `setcvar` program contact the author for a working version of the `setcvar` program.

/sbin/consvar Tru64 UNIX program

Another method of getting and setting the values of the console variables involves the `/sbin/consvar` program.

```
kamlia.zk3.dec.com> /sbin/consvar help
Firmware Console Environment Variable Manager
 consvar [-v] [-nc] -g -s -a -l -d [variable] [value]
 -v                    Verbose
 -g variable           Get the value of variable
 -s variable value     Set the value of variable
 -l                    List all variables with their values
 -a                    Save all variables to NV storage
 -d                    Print out exception database data
 -nc                   Do NOT perform conversion
```

Obtain a verbose (`-v` option) of all variables with their values (`-l` option):

```
kamlia.zk3.dec.com> /sbin/consvar -v -l
Firmware Rev: 6.8-13
system fam:34  cpu:8  smm:1814
auto_action = HALT
boot_dev = dsk6(4 paths)
bootdef_dev = dsk6(4 paths)
booted_dev = dsk6
boot_file =
booted_file =
boot_osflags = s
booted_osflags = s
boot_reset = ON
Failed to get dump_dev
enable_audit = ON
license = MU
char_set =
language = 0x36
tty_dev = 0
Failed to get scsiid
Failed to get scsifast
com1_baud = 9600
com1_modem = OFF
com1_flow = SOFTWARE
Failed to get com1_misc
com2_baud = 9600
com2_modem = OFF
com2_flow = SOFTWARE
Failed to get com2_misc
Failed to get password
secure = off
logfail = 0
srm2dev_id =
```

HP OpenVMS Alpha Console

The HP OpenVMS operating system provides two methods of getting the console environment variables: the lexical function `f$getenv()` and the `sys$getenv()` system service library call. Figure 4 illustrates how to use the lexical function `f$getenv()` from the operating system prompt to get the value of the `auto_action` console variable.

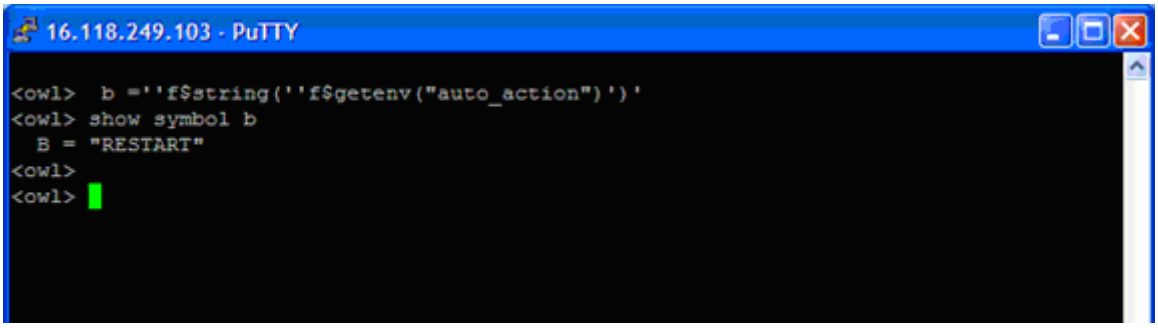
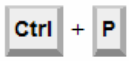


Figure 4 - Using the f$getenv() lexical function

The next example, illustrates how to obtain the value of the console variable `auto_action` from the console using a direct serial connection to the system console:

```
Welcome to OpenVMS (TM) Alpha Operating System, Version V7.3-2

Username:    <Ctrl>P

halted CPU 0
CPU 1 is not halted
CPU 2 is not halted
CPU 3 is not halted

halt code = 1
operator initiated halt
PC = ffffffff88670438
P00>>> show auto_action
auto_action             RESTART
P00>>> cont

continuing CPU 0
```

Note the **<Ctrl>P** is not the username but the two keyboard keys:



To enter the SRM console, hold down the Ctrl key and press the P key.

Figure 5 illustrates how to get the value of the Alpha console variable `auto_action` from the console using the ConsoleWorks tool.

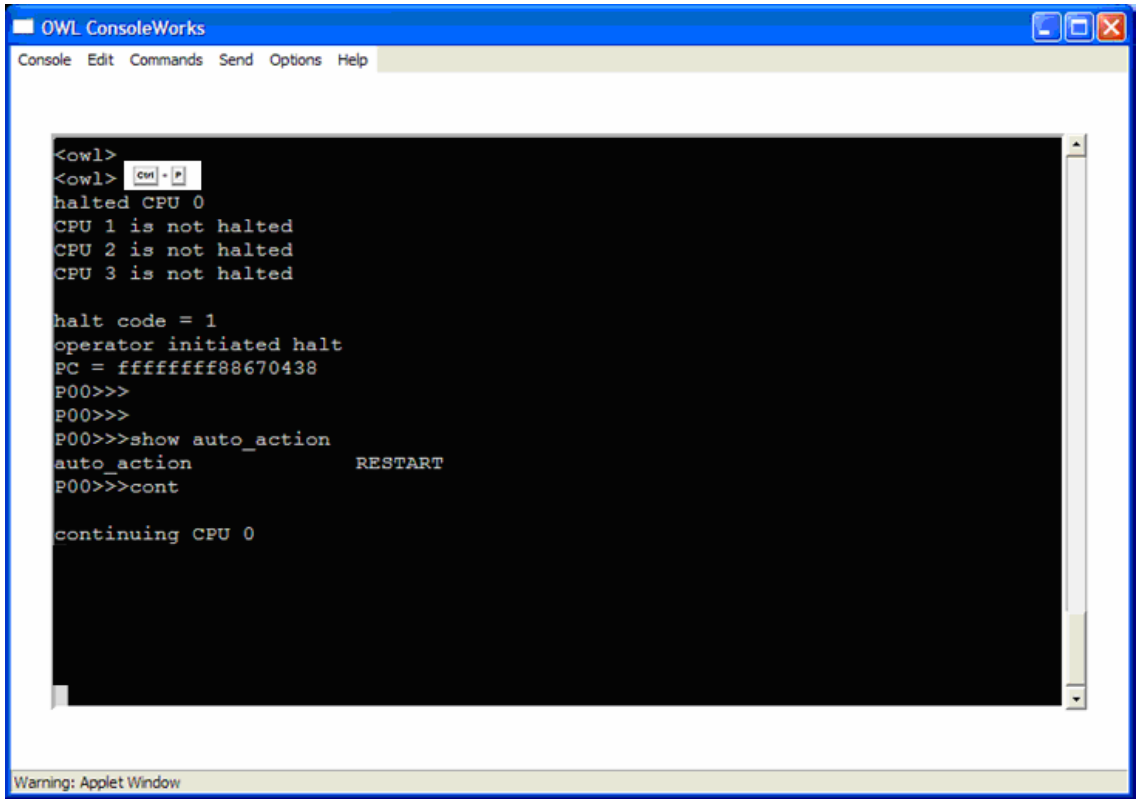


Figure 5 - ConsoleWorks access to Alpha console

Existing HP tools that get the console variables and their values

The two tools used by HP Services professionals to collect AlphaServer system configuration information are the sys_check for HP Tru64 UNIX and the VMS_Check[4] for HP OpenVMS platforms.

sys_check

The sys_check tool gets the console variables on HP Tru64 UNIX systems. The following window (fig. 6) represents a typical report of the information collected by sys_check.

Figure 6 - sys_check report on the console variables

[4] For more information on VMS_Check please reference the OpenVMS Technical Journal article http://h71000.www7.hp.com/openvms/journal/v7/vms_check_tool.html or contact the author.

VMS_Check

The VMS_Check tool gets the console variables on HP OpenVMS systems. The following window (fig. 7) represents a portion of the information provided by VMS_Check on the HTML generated code as it is viewed in Microsoft Internet Explorer.

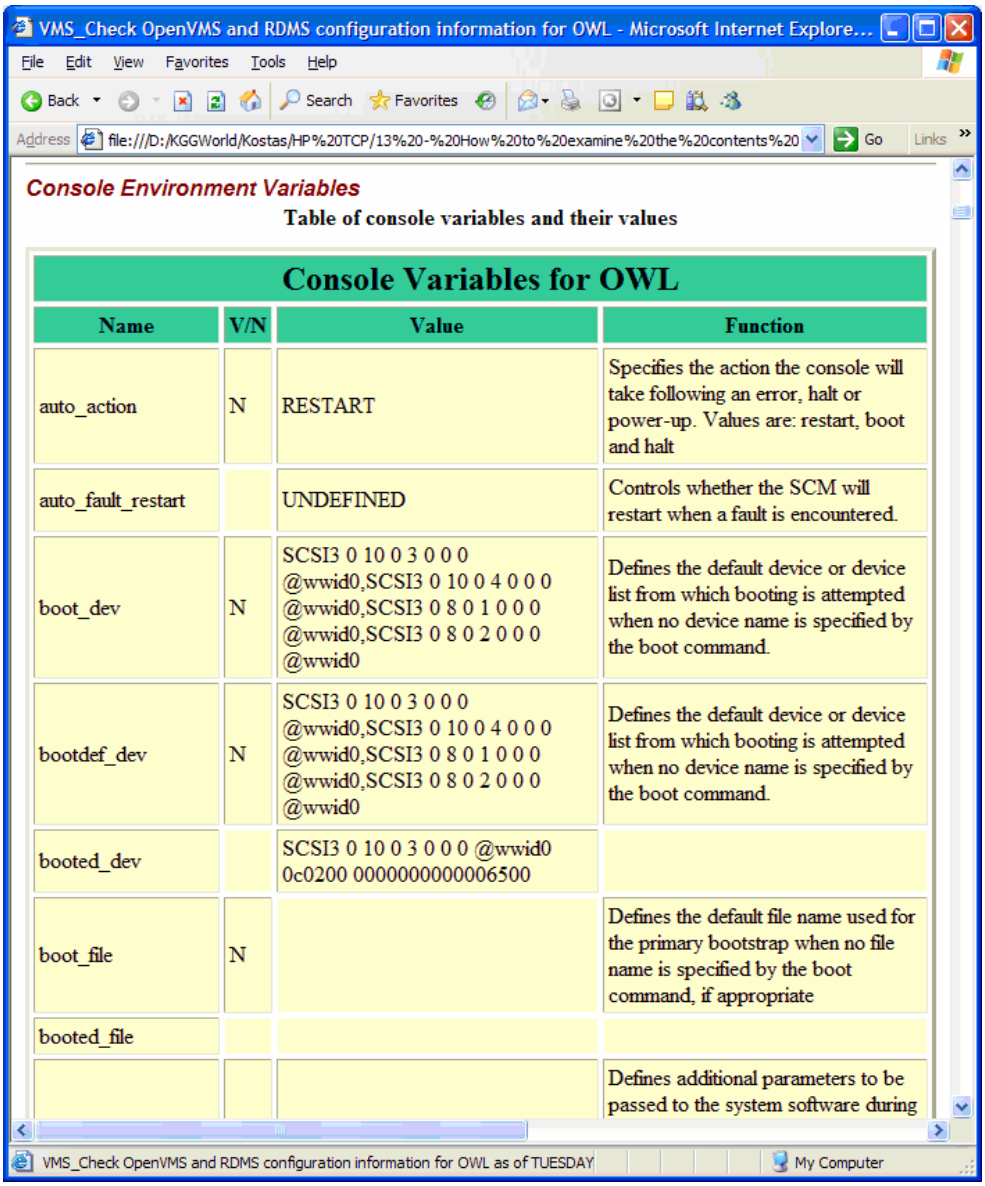


Figure 7 - VMS_Check report of the console variables

How to update the SRM console

There are several methods available to update the SRM firmware, depending of the Operating System. One method, that is common to all, involves creating a bootable firmware update CD-ROM from an ISO image available from the URL: ftp://ftp.digital.com/pub/DEC/Alpha/firmware/index.html, and then booting the system from the created bootable CD-ROM.

Summary

This article explored the Alpha SRM console and the different ways to access it and to set and retrieve the values of the console environment variables.

References

GRUB – ftp://alpha.gnu.org/pub/gnu/grub/

LILO – http://www.tldp.org/HOWTO/LILO.html

MILO - http://www.tldp.org/HOWTO/MILO-HOWTO/index.html

EFI - http://www.intel.com/technology/efi/

TECSys Development Inc (TDi) - http://www.tditx.com/

Sys_check - http://h30097.www3.hp.com/sys_check/

VMS_Check – To learn more about VMS_Check please see the OpenVMS Technical Journal article at the URL: http://h71000.www7.hp.com/openvms/journal/v7/vms_check_tool.html and to get a copy of the VMS_CHECK tool please download it from the URL: http://h71000.www7.hp.com/openvms/journal/v7/vms_check.zip . Please email the author with any questions or comments regarding VMS_Check.

hp AlphaServer ES47/Es80/GS1280 Server Management SRM Console Reference Guide V1.0 - http://h18002.www1.hp.com/alphaserver/download/srm_reference.pdf

Alpha Systems Firmware Updates - ftp://ftp.digital.com/pub/DEC/Alpha/firmware/index.html

Alpha Architecture Reference Manual – http://lab46.corning-cc.edu/haas/spring2005/asm/public/doc/Alpha_Architecture_Reference_Manual.pdf

Acknowledgements

The author wishes to acknowledge his GSE-MSE colleagues Laurence Oberman and Joe Bowker for providing access to the ConsoleWorks configurations on Tru64 Unix and OpenVMS systems in the MSE interoperability lab. In addition, I would like to thank Ken W. Campbell, Bayard W. Wenzel, and Ken Sullivan, also from GSE-MSE, for their review of this article and for their valuable feedback and contributions.

About the Author

Kostas G. Gavrielidis works in GSE-MSE which is part of HP Services Customer Support and has been at HP for more than 22 years. Currently, and for the last 12 years, he is involved with the MSE proactive consulting projects for our customer production Database Management systems, and he works on the analysis and performance improvements for SAP R/3, Oracle, Rdb, Ingres, SYBASE, SQL Server on UNIX, Linux, OpenVMS, and Windows platforms.

For more information

Please contact the author with any questions or comments regarding this article or the VMS_Check tool.

To get to the latest issue of the OpenVMS Technical Journal, go to: http://www.hp.com/go/openvms/journal.