**VMS\_Check – a Tool to collect OpenVMS Configuration data**

By Kostas G. Gavrielidis

Several layered products and utilities exist for collecting operating system configuration and performance data, along with layered product configuration and performance information for the HP OpenVMS operating system. OpenVMS currently runs on three different hardware architectures, namely VAX, Alpha and Itanium. This Knowledge Brief presents the VMS\_Check utility - this has been developed by the author to collect OpenVMS configuration information. VMS\_Check is written entirely in the DIGITAL Command Language (DCL).

DCL is similar to any of the UNIX shells, such as the Bourne shell (sh), the C shell (csh) and the Korn shell (ksh). DCL is a command language interpreter and as such it parses commands and passes control to other programs that make up the OpenVMS operating system. While programs developed on any one of the OpenVMS compilers such as, C/C++, Pascal, BLISS, FORTRAN, COBOL may not run unchanged or without re-linking across all the three architectures, DCL procedures still work without change.

## What is VMS\_Check?

VMS\_Check was created in the spirit of the sys\_check[[1]](#footnote-1) and cfg2html[[2]](#footnote-2) tools which run on UNIX systems. VMS\_Check is a DCL procedure that runs on all three OpenVMS architectures and is extendable in the sense that you can include in it any series of OpenVMS commands you would like to issue as if you were on the OpenVMS operating system command prompt ($). In the current version of VMS\_Check, the data collected from any system, standalone or in an OpenVMS Cluster, is presented in its original form with HTML wrappers. The main report is an HTML file that has the name of: VMS\_Check-<Nodename>-<DDMONYYYY-HHMM>.HTML as in this example: VMS\_Check-OWL-14MAR2005-1516.HTML. This main file is supported by numerous text and HTML files that contain the actual data that make up the complete system report.

## Why develop VMS\_Check?

VMS\_Check started out as a small procedure with the goal to collect database-related configuration information. Slowly it grew to a large DCL command procedure that now includes operating system and storage configuration information. The primary goal was to collect the data on a customer’s configuration.

All of the VMS\_Check report sections include information in tables or plain text which can easily be used elsewhere, such as in any of the Microsoft tools Word, Excel, etc.

An example of a table generated on Cluster systems, their version, node name, current date and time and system uptime is shown below.

|  |
| --- |
| BBCX Cluster Nodes Table |
| OpenVMS Version | Node Name | Current Date and Time | Uptime |
| OpenVMS V7.3-2 | BBC200 | 25-FEB-2005 11:40:47.70 | 27 10:33:44  |
| OpenVMS V7.3-2 | BBC202 | 25-FEB-2005 11:40:47.73 | 27 09:51:18  |
| OpenVMS V7.3-2 | BBC204 | 25-FEB-2005 11:40:47.78 | 9 12:45:43  |
| ... | ... | ... | ... |
| OpenVMS V7.3-2 | BBC309 | 25-FEB-2005 11:40:48.12 | 19 12:44:58  |
| OpenVMS V7.3-2 | BBC311 | 25-FEB-2005 11:40:48.15 | 2 12:05:09  |

Table 1: Cluster Nodes Table

## What type of information is collected by VMS\_Check?

VMS\_Check collects setup and configuration information for database and associated layered products such as Oracle, Rdb, ACMS, Ingres etc. on OpenVMS platforms.

 

Figure 1: VMS\_Check report in the MS IE and Mozilla Firefox browsers

## VMS\_Check Report Sections

As with most generated HTML reports, VMS\_Check creates a Table of Contents (TOC) at the beginning of the report. The following is an example of the TOC generated by VMS\_Check:

***Table of Contents***

* **OpenVMS Operating System**

Console: Variables| RAD Information| Partition Information
Procedures for System: Sylogin| Startup| Shutdown| ModParams
System identification information: GETSYI of this node| GETSYI of all VMScluster nodes|
Show commands for: System| CPU| Memory| Pool | Files| Reserved Memory| RMS| Users| Logicals| Symbolics|
Analyze: System| RAD| SpinLock|
Errolog and Crash Information: Errorlog| DECevent| Crash analysis
SYSGEN Parameters: SYSGEN| Startup| Special| All
System Access and Control: UAF Records| User Rights| Proxies| SYSTEM UAF record| DEFAULT UAF records| Table
Installed Images: Installed images
System tests: RADcheck
VMScluster Configuration Information: VMScluster| Noders Table| GETSYI table for all VMScluster nodes

* **Storage Subsystem Configurations**

Storage: Devices| IO Bus| IO Circuits| IO Devices| Mounted| Devices Table| Devices Charts| Devices Fragmented Files| Stripe| RAID| FDDI| HSC| HSJ

* **Network and Related Products Information**

Network| NCP | NCL | LATCP | LANCP | UCX | MultiNet

* **Database Information/Configurations**

SQL: SQL| SQL Images| UAF Records
Relational Database Operator: RDO
Oracle Rdb: Images| Logicals| Versions| Databases| Schemas| Statistics| UAF Records
Oracle RDBMS: Oracle| Schemas| Statistics| UAF Records
Sybase RDBMS: Sybase| UAF Records
Ingres RDBMS: Ingres| UAF Records

* **Transaction processing and other layered product information**

ACMS: ACMS| Images
TDMS: TDMS
DECforms: DECforms
PathWorks: PathWorks
DECWindows: DECwindows
DECthreads: Images
CMA: Images
Layered Products: Installed| Installation History| Licensed
HyperSort: Images

* **Performance Data**

Performance Solution Advisor (PSA)| Monitor Utility

* **Tables**

VMScluster nodes| GETSYI information for all VMScluster nodes| Devices| SYSUAF

* **Interactive Sessions**

0:Mornitor| 1:Rdb| 2:Rdb| 3:ACMS| 4:PSA| 5:SPL

**Goto:** Top|Contents|Bottom

## Navigation in VMS\_Check

From inside the main generated HTML report file and under each section of the report, you can jump to different sections of interest in the report.

The following menu appears:

**Goto:** Top | Contents | Bottom
**Section:** System |Storage |Network | Database |Layered Products | Performance | Interactive

The **Goto** references jump to the Top, Table of Contents and Bottom sections of the report. The **Section** references jump to the System, Storage, Network, Database, Layered Products, Performance and Interactive sections of the report.

## VMS\_Check generated internal references to other files

The main report HTML file generated by VMS\_Check has internal references to other text and HTML files generated by VMS\_Check. These file references and their content description is summarized in the Table below. This example was generated for the OpenVMS system named OWL.

|  |
| --- |
| Internal References generated for OWL |
| Reference | Description |
| Devices\_Mounted-OWL.txt | All devices mounted  |
| Devices\_DU-OWL.txt | All DU devices  |
| Devices\_HSJ-OWL.txt | All HSJ devices  |
| SDA-of-running-OWL.txt | Analyze System |
| SPL-of-running-OWL.txt | Spinlock Information |
| ... | ... |
| PSA-Brief-OWL.TXT | Performance Analysis (Brief) |
| PSA-Full-OWL.TXT | Performance Analysis (Full) |
| PSA-Perf-OWL.TXT | Performance Evaluation |
| MON-ALL-SUM-OWL.TXT | Monitor all classes (Summary) |
| MON-ALL-AVE-OWL.TXT | Monitor all classes (Average) |

Table 2: Internal References

## VMS\_Check generated external URL references

HP OpenVMS homepage: <http://h71000.www7.hp.com/>
HP OpenVMS Documentation: <http://h71000.www7.hp.com/doc/>
HP OpenVMS Patches: <http://h71000.www7.hp.com/serv_support.html>
HP OpenVMS Frequently Asked Questions (FAQ): <http://h71000.www7.hp.com/faq/openvms_faq.html>
HP OpenVMS Services: <http://h20219.www2.hp.com/services/cache/10647-0-0-225-121.aspx>
HP OpenVMS Systems Solutions and Applications: <http://h71000.www7.hp.com/solutions/index.html>
HP Support and Drivers: <http://www.hp.com/country/us/en/support.html>
HP Storage products: <http://www.hp.com/country/us/en/prodserv/storage.html>
HP Products and Services: <http://www.hp.com/country/us/en/prodserv.html>
HP Software products: <http://www.hp.com/country/us/en/prodserv/software.html>
VMS\_Check kit and documentation: Contact Kostas.Gavrielidis@hp.com

## Report Highlights

Some of the section highlights in the generated report are included here for your reference.

### Mounted Devices

Looking at the Storage section of the report and one very useful area, namely the mounted devices table, VMS\_Check generates a table of all devices mounted with information on each device, that is: the Device name, Volume Name, Device Type, Total blocks, Free blocks, %Free, Fragmentation Index and Fragmentation Report.

|  |
| --- |
| Devices on OWL  |
| Device | Volume | Type | Total blocks | Free blocks | %Free | Frag Index | Frag Report |
| \_DSA2: | ALP\_SITEB | DGX00 | 68251131 | 24772725 | 36 | 44.1 | frpt |
| \_DSA11: | ALPHA\_USER | DGX00 | 71112778 | 24899409 | 35 | 5.2 | frpt |
| \_DSA12: | DATABASE | DGX00 | 213291762 | 38709535 | 18 | 2.4 | frpt |
| \_DSA500: | OWL\_PAGE | DKX00 | 35565080 | 20564320 | 57 | 30.2 | frpt |

Table 3: Mounted Devices Table

Another useful part of this report section is the generated bar graph showing the %Free Disk Space and Fragmentation Index for all devices, along with a pie chart graph of the total Disk Capacity, which includes the total MB’s used and free disk space for devices.



Figure 2: Mounted Devices Charts

### Console Environment Variables

The Table 4 contains some of the information that can be collected from the console.

|  |
| --- |
| Console Variables for OWL  |
| Name | V/N | Value | Function |
| auto\_action | N | RESTART | Specifies the action the console will take following an error, halt or power-up. Values are: restart, boot and halt |
| auto\_fault\_restart |  | UNDEFINED | Controls whether the SCM will restart when a fault is encountered. |
| Boot\_dev | N | SCSI3 0 10 0 3 0 0 0 @wwid0,SCSI3 0 10 0 4 0 0 0 @wwid0,SCSI3 0 8 0 1 0 0 0 @wwid0,SCSI3 0 8 0 2 0 0 0 @wwid0 | Defines the default device or device list from which booting is attempted when no device name is specified by the boot command. |
| Bootdef\_dev | N | SCSI3 0 10 0 3 0 0 0 @wwid0,SCSI3 0 10 0 4 0 0 0 @wwid0,SCSI3 0 8 0 1 0 0 0 @wwid0,SCSI3 0 8 0 2 0 0 0 @wwid0 | Defines the default device or device list from which booting is attempted when no device name is specified by the boot command. |
| ... |  | ... | ... |

Table 4: Table of the console environment variables and their values

### GETSYI information

Looking at some of the data collected, we can take one example, as shown in the following Table. This shows GETSYI information for a single node:

|  |
| --- |
| GETSYI Information for EMPIRE  |
| Type | Value | Description |
| ACTIVECPU\_CNT | 2 | Count of the CPUs actively participating in the current boot of the symmetric multiprocessing (SMP) system.  |
| AVAILCPU\_CNT | 2 | Number of CPUs available in the current boot of the SMP system.  |
| ARCHFLAG | 245760 | Architecture flags for the system  |
| ARCH\_NAME | Alpha | Name of the CPU architecture on which the process is executing  |
| BOOTTIME | 24-SEP-2004 14:11:24.00 | The time when the node was booted.  |
| … |  |  |
| VP\_MASK | 0 | Longword mask, the bits of which, when set, indicate which processors in the system have vector coprocessors.  |
| VP\_NUMBER | 0 | Unsigned longword containing the number of vector processors in the system.  |

Table 5: GETSYI Information from a single node

### GETSYI information cluster wide

Table 6 shows cluster-wide GETSYI information.

|  |
| --- |
| GETSYI Information for all VMSCluster Nodes  |
| Item | EMPIRE Value | AIRTRN Value | …  | STORM Value |
| CLUSTER\_EVOTES | 11 | 11 |  | 11 |
| CLUSTER\_FSYSID | 0000000034AD | 0000000034AD |  | 0000000034AD |
| CLUSTER\_MEMBER | TRUE | TRUE |  | TRUE |
| CLUSTER\_NODES | 11 | 11 |  | 11 |
| DECNET\_FULLNAME | LOCAL:.EMPIRE:: | LOCAL:.AIRTRN:: |  | LOCAL:.STORM:: |
| HW\_MODEL | 1976 | 1962 |  | 2030 |
| HW\_NAME | AlphaServer ES45 Model 2 | AlphaServer DS10L 617 MHz |  | hp AlphaServer ES47 7/1000 |
| NODENAME | EMPIRE | AIRTRN |  | STORM |
| ... | ... | ... | ... | ... |
| NODE\_SYSTEMID | 0000000035A5 | 0000000034B0 |  | 00000000343A |
| NODE\_VOTES | 1 | 0 |  | 1 |

Table 6: GETSYI information cluster wide

## How to use VMS\_Check?

There are several ways to invoke the VMS\_Check tool. The most useful is the command procedure included with the software kit. This is named: VMS\_Check-Detail.COM.

VMS\_Check-Detail.COM invokes the VMS\_Check command procedure by setting the appropriate privileges and flags for VMS\_Check to collect HTML and DETAIL configuration information on the RDBMS’s and on the OpenVMS environment.

$ @VMS\_Check-Detail

This DCL command procedure actually performs these steps:

...

$! Determine output file name

$!

$ nodename = f$getsyi("nodename")

$ systime = f$edit(''f$time()', "TRIM")

$ date = f$element(0, " ", systime)

$ time = f$element(1, " ", systime)

$ day = f$element(0, "-", date)

$ mon = f$element(1, "-", date)

$ year = f$element(2, "-", date)

$ hour = f$element(0, ":", time)

$ min = f$element(1, ":", time)

$ if f$length(day) .eq. 1 then day = "0"+day

$ filename = "VMS\_Check-"+nodename+"-"+day+mon+year+"-"+hour+min+".HTML"

$! show symbol filename

$ SET PROC/PRIV=(ALL,NOBYPASS)

$ exec := SPAWN/NOWAIT/INPUT=NL:/OUTPUT='filename'

$! show symbol exec

$ EXEC @VMS\_Check.COM HTML DETAIL

Other methods of invocation of VMS\_Check include:

* On a single system environment

 $ SET PROC/PRIV=(ALL,NOBYPASS)

 $ EXEC:= SPAWN/NOWAIT/INPUT=NL: -

 /PROCESS=VMS\_Check -

 /OUTPUT=VMS\_Check-'F$getsyi("NODENAME")'.HTML

* To generate HTML output and to get a DETAIL description of the current environment, use:

 $ EXEC @VMS\_Check.COM HTML DETAIL

* To debug VMS\_Check use the following command:

 $ EXEC @VMS\_Check.COM DEBUG NODETAIL

The DEBUG option creates the file VMS\_Check.DEBUG in the current directory. This file has a record of all actions performed by VMS\_Check preceded by a time stamp at the start of each action.

* On A VMS Cluster Environment

To generate HTML output from all the nodes on the VMS Cluster, use:

$ MCR SYSMAN

SYSMAN> SET ENV/CLUSTER

SYSMAN> DO -

\_SYSMAN> SPAWN

\_SYSMAN> /INPUT=NL:/OUTPUT=VMS\_Check\_'F$getsyi("NODENAME")'.HTML -

\_SYSMAN> /PROCESS=VMS\_Check -

\_SYSMAN> @DSA110:[KOSTAS.DEV]VMS\_Check.COM HTML NODETAIL

SYSMAN> EXIT

## Successful completion of the VMS\_Check report

The completion state is given at the end of the generated report. Here is an example of a successfully generated report:

|  |
| --- |
|  \*---------------------------------------------------------------------\* \* \* \* VMS\_Check has successfully completed. \* \* \* \*---------------------------------------------------------------------\* |

*Generated by VMS\_Check.COM X0.1-177 on 16-MAY-2005 14:13:31.10*

## Transfer the generated files and review the VMS\_Check report

To transfer and review the report and associated generated files from the OpenVMS environment to other environments such as a Windows laptop, you need to ftp transfer, in ASCII mode, all the files generated by VMS\_Check. I use the following ftp commands to transfer the VMS\_Check files to my laptop for review:

**ftp> open owl.mse.lkg.hp.com**

Connected to owl.mse.lkg.hp.com.

220 owl.mse.lkg.hp.com FTP Server (Version 5.4) Ready.

User (owl.mse.lkg.hp.com:(none)): **kostas**

331 Username kostas requires a Password

Password:

230 User logged in.

**ftp> cd [kostas.dev.prey]**

250-CWD command successful.

250 New default directory is DSA11:[KOSTAS.DEV.PREY]

**ftp> hash**

Hash mark printing On ftp: (2048 bytes/hash mark) .

**ftp> prompt**

Interactive mode Off .

**ftp> mget \***

...

**ftp> quit**

## Summary

This Knowledge Brief describes the VMS\_Check tool for OpenVMS environments. It describes how to use it and what type of information is collected. Some of the report areas that might be of interest to System Managers and HP Customer Services professionals are highlighted.

1. The sys\_check tool provides configuration and analysis of information gathered on the system. It can be very useful when debugging or diagnosing system problems. Currently sys\_check gathers over 60 components and subsystems, and performs over 200 analysis operations. Sys\_check gathers this information into easy to browse and transportable files. These files are sent to support engineering when escalating IPMT cases. Sys\_check runs on all supported version of the Tru64 UNIX operating system. Sys\_check is included in the Tru64 UNIX operating system and the patch kits. [↑](#footnote-ref-1)
2. The cfg2html is a UNIX shell script that creates a HTML (and ASCII) system documentation for HP-UX 10+11, AIX, SCO-UX, SunOS and Linux systems. Plugins for SAP, Oracle, Informix, MC/SG, FibreChannel, TIP/ix, Mass Storage like XP48/128/256/512/1024/12000, EVA3000/EVA5000, Network Node Manager and OmniBack/DataProtector etc. are included. [↑](#footnote-ref-2)