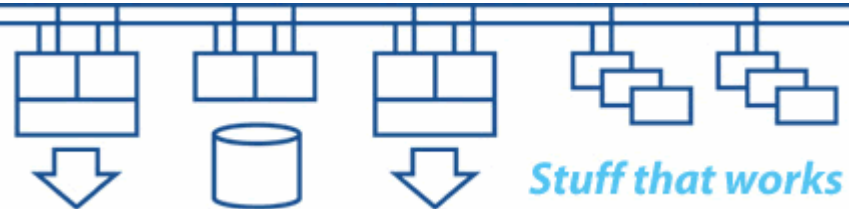


OpenVMS on current platforms (Integrity, Alpha and VAX)

Colin Butcher, XDelta Limited

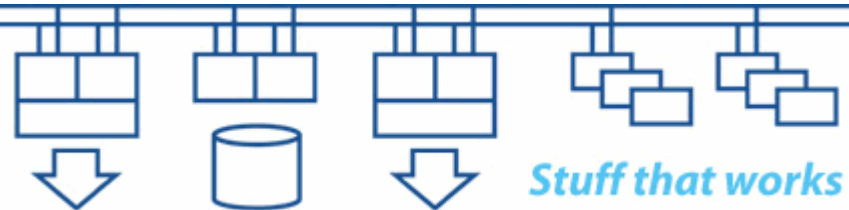
Martin Riley, HP



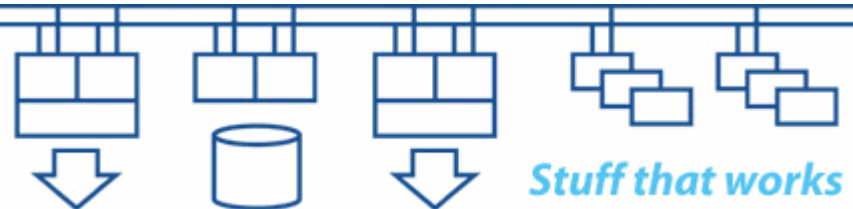
Gain an understanding of how OpenVMS is installed and configured on the different hardware platforms. The seminar will concentrate on the differences in the hardware platforms, the boot process and the installation process.

This seminar is unlikely to answer all of your questions, so please be prepared to contribute and share your knowledge.

- **Introduction to OpenVMS – some fundamentals**
- **Hardware platform differences**
- **Console subsystems**
- **The boot process**
- **OpenVMS installation on VAX, Alpha and Integrity**
- **Post installation actions**
- **Creating bootable optical media from OpenVMS**



- **Occam's Razor:**
“Pluralitas non est ponenda sine neccesitate”
“Entities should not be multiplied unnecessarily”
“Keep it as simple as possible”
- **Hanlon's Razor:**
“Never attribute to malice that which can be adequately explained by stupidity”
- **Colin's Razor:**
“Allow for failures, success is one of many possible outcomes”

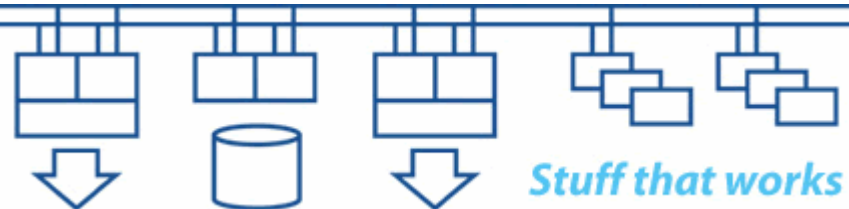


VAX and Alpha are used interchangeably to refer to system architectures, processors / microprocessors and platforms

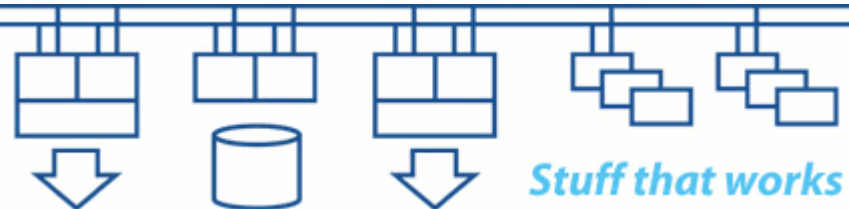
- **VAX VMS was used up to V5.5-2H4**
- **OpenVMS VAX and OpenVMS Alpha are used from V6.0 onwards**

Intel implementations are different:

- **IA-64 is the Intel architecture**
- **Itanium is the Intel microprocessor family**
- **Integrity is the HP platform**
- **OpenVMS I64 is the HP operating system name**



- **History**
- **Provides genuine backwards compatibility**
- **Pioneered cluster technology – and still the best! Uses a “shared everything” model – the DLM is the key.**
- **VAX, Alpha and Integrity hardware platforms**
- **Partitioning and Galaxy**
- **System and device interconnects**
- **Cluster common system disc file structure**



- **ODS2 v ODS5 disc structure**
- **Crucial file locations (SYSUAF etc.)**
- **Startup mechanisms (hardware autoconfigure, startup sequence, crucial files etc.)**
- **System management features (image backup / restore, consistent command line interface etc.)**
- **Security features (files, devices, queues, [ab]users etc. protection) and access control**

V1.x ... V3.x (included RSX11 compatibility mode)

V4.x (V4.7A) (clustering started here)

V5.x (V5.5-2H4)

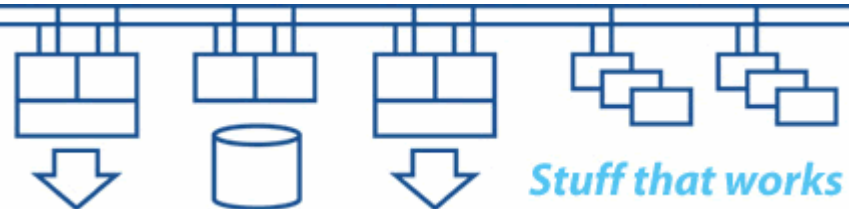
V6.x (V6.2-1H3)

V7.x (V7.3-2)

V8.x (EFT 8.2)

Current versions of OpenVMS:

V7.3 (VAX), V7.3-2 (Alpha), E8.2 (Alpha & Integrity)



Phase I .. III

Phase IV

Phase IV Plus

Phase IV Extensions

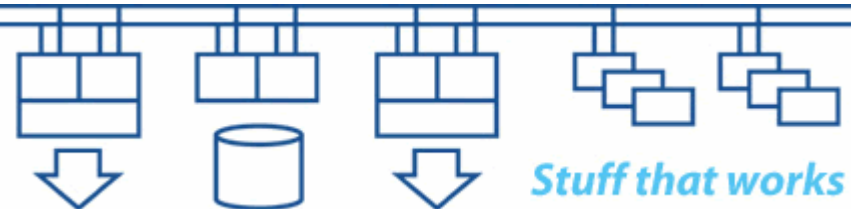
Phase V (DECnet/OSI, DECnet-Plus)

Current versions of DECnet-Plus:

V7.3-ECO04 (VAX V7.3)

V7.3-2 (Alpha V7.3-2)

M8.2 (Alpha & Integrity E8.2)



UCX V1.x – V4.x

Current version of UCX:

V4.2-ECO05 (VAX and Alpha)

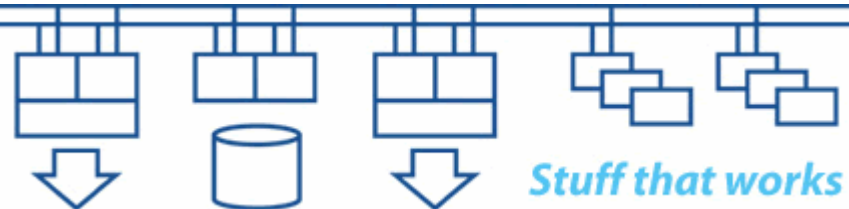
TCP/IP Services V5.x from OpenVMS V7.1

Current versions of TCP/IP Services:

V5.3-ECO04 (VAX V7.3 & Alpha V7.3-1)

V5.4-ECO02 (Alpha V7.3-2)

T5.5-8I (Alpha & Integrity E8.2)



V1.1

V1.2-3 ... V1.2-6 (CDE on Alpha only, V7.3-1 and earlier)

V1.3-1 (Alpha only, V7.3-1 and later)

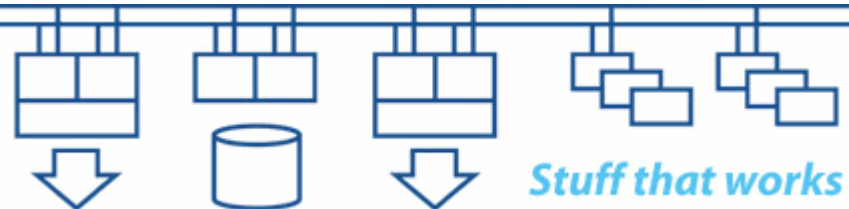
T1.4.1 (Integrity EFT 8.1)

Current versions of DECwindows/Motif:

V1.2-6 (VAX V7.3, no CDE)

V1.3-1 (Alpha V7.3-2)

K1.5 (Alpha & Integrity E8.2)

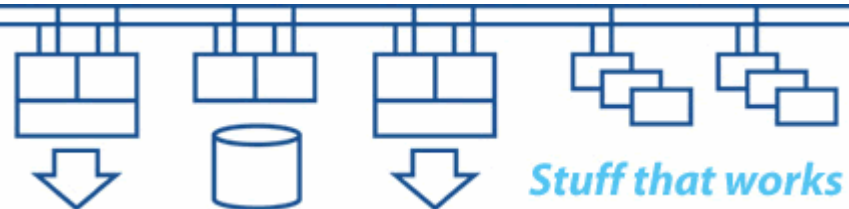


- **RX2600:**
 - **OpenVMS I64 E8.2, DECnet-Plus M8.2, TCP/IP T5.5-8I**
- **2x AlphaServer DS10L:**
 - **OpenVMS Alpha V7.3-2, DECnet-Plus V7.3-2, TCP/IP V5.4**
 - **OpenVMS Alpha E8.2, DECnet-Plus M8.2, TCP/IP T5.5-8I**
- **VAX 4000-100:**
 - **Nemonix UW SCSI and Fast ethernet (100Mbit/sec FDX)**
 - **OpenVMS VAX V7.3, DECnet-Plus V7.3, TCP/IP V5.3**

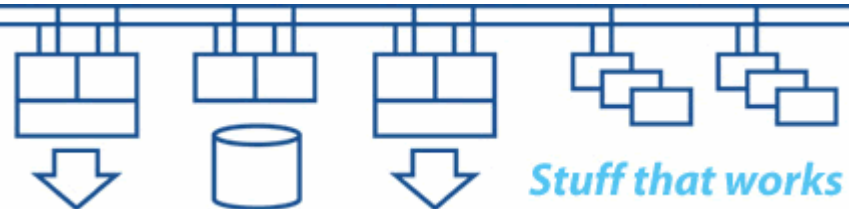
- **32bit CISC. Designed when memory was expensive, so instruction set tries to save space where possible (variable length instructions etc.)**
- **Includes specific instructions to assist when writing operating system software, such as providing guaranteed synchronisation of access to data structures.**
- **Early VAXes (VAX-11) included PDP11 instructions. Later VAXes use PDP11 software emulation.**
- **Support a wide range of IO devices and bus structures.**
- **“Bomb-proof” engineering, especially big VAXes.**

- **64bit RISC. Designed for performance.**
- **Best performance achieved with data aligned on 64bit boundaries (no need for bit masking in registers etc.).**
- **Needs synchronisation issues to be carefully considered and coded. What was a single “atomic instruction” on VAX can be multiple instructions on Alpha, so instruction flow through the CPU can be interrupted.**
- **“VAX like” console commands (SRM console).**
- **“PC like” BIOS (AlphaBIOS console).**
- **Predominantly PCI bus based.**

- **64bit EPIC. Designed when memory is plentiful and cheap.**
- **Itanium Processor Family architecture relies on compiler output to generate an efficient instruction and data flow through the CPU**
- **Needs synchronisation issues to be carefully considered and coded. What was a single “atomic instruction” on VAX can be multiple instructions on Integrity, so instruction flow through the CPU can be interrupted.**
- **Instructions are packaged in “bundles” of up to three instructions per bundle – which are then processed entirely in parallel.**



- **No “VAX like” or “Alpha like” console**
- **Has multiple consoles:**
 - **Management Processor (MP)**
 - **Baseboard Management Console (BMC)**
 - **Both attempt to be common across the entire hardware range, unlike VAX and Alpha which tended to be machine specific.**
- **Uses Extensible Firmware Interface (EFI) rather than BIOS**
- **PCI bus based (3.3volt only)**



- “>>>” prompt
- Serial and graphics consoles (VAXstations)
- Power-up self tests
- Device detection
- Device naming convention matches OpenVMS device naming convention
- Device configuration (SCSI ID, DSSI ID etc.)
- System configuration (language, auto boot, etc.)
- Data passed in CPU registers (R5 used for boot flags) to boot loader.

- “>>>” prompt
- Serial and graphics consoles
- Power-up self tests
- Device detection
- Device naming convention matches OpenVMS device naming convention
- Environment variables
 - Device configuration (SCSI ID, DSSI ID etc.)
 - System configuration (language, auto boot, etc.)
- Data passed to boot loader, eg: `boot –flags 0,0 <device>`

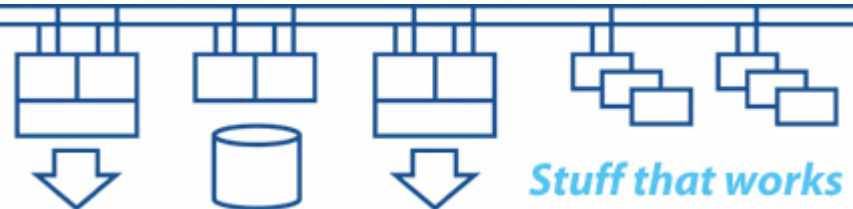
- **MP console for “Management Processor”**
 - **Runs with base box level power, even with system off**
 - **Local, remote (modem) and network connectivity**
 - **Console configuration (terminal type, etc.)**
 - **<ctrl-H>, <ctrl-B>**
 - **Network configuration (hostname, IP address, etc.)**
 - **Multiple console sessions (one writer, many readers)**
 - **Provides ability to copy files over network (firmware updates etc.)**

- **BMC console for “System Board”**
 - **Runs when main board powered up**
 - **Local connectivity (9 pin serial) and logical connectivity (MP console sessions)**
 - **Power-up self tests**
 - **Device detection**
 - **Console configuration (terminal type, etc.)**
 - **<ctrl-H>**
 - **No graphics console**

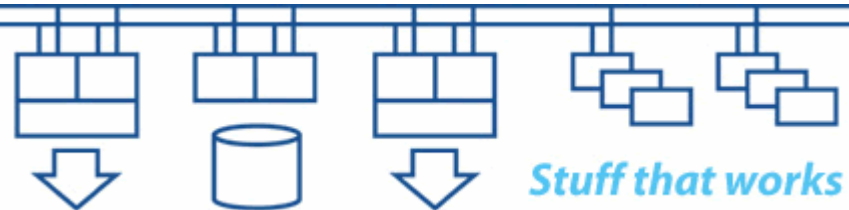
- **EFI (extensible firmware interface):**
 - Mini operating system
 - FAT formatted file system (FAT12, FAT16, FAT32), OpenVMS currently presents a FAT16 partition to EFI.
 - Boot menu and defaults
 - Environment variables (VMS_FLAGS etc.)
 - VMS_LOADER.EFI - finds and loads IPB.EXE
 - Data passed to boot loader, eg: `efi> vms_loader -fl 0,0`
- **IPB.EXE (boot loader) understands the OpenVMS file structure, EFI does not.**

- **Console firmware**
- **Loader finds VMB (method varies with platform), eg:**
 - On system disc via boot block
 - Console ROM or NVRAM
- **Boot flags passed in registers**
- **Reads executive into memory**
- **Reads system parameters (SYSBOOT> if flags set) and initialises fixed data structures**
- **Passes control to executive**

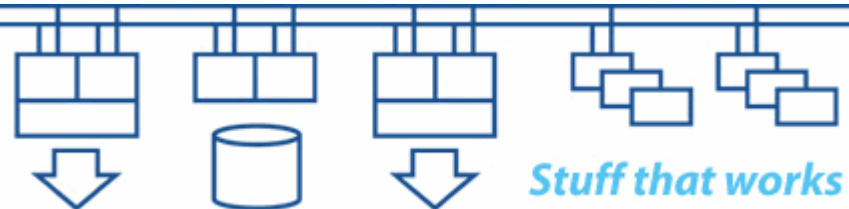
- **Console firmware**
- **Loader finds APB on system disc via boot block**
- **Boot flags passed in registers**
- **Reads executive into memory**
- **Reads system parameters (SYSBOOT> if flags set) and initialises fixed data structures**
- **Passes control to executive**



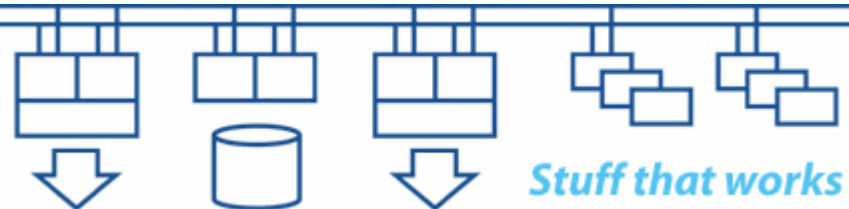
- **EFI boot loader from FAT partition (hidden as a container file in the system disc file structure)**
- **Boot flags passed through environment variables**
- **Reads executive into memory**
- **Reads system parameters (SYSBOOT> if flags set) and initialises fixed data structures**
- **Passes control to executive**
- **Structures created with SET BOOTBLOCK command**
- **2048 byte boot block for IDE/ATAPI optical media**
- **512 byte boot block for SCSI optical media**



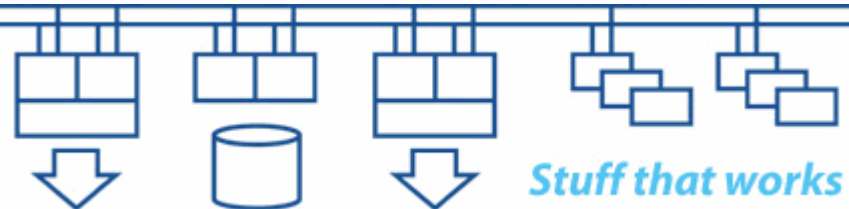
- **Console firmware**
- **Devices appear as a set of CSRs (Control and Status Registers) in physical memory - the IO space**
- **Devices have Interrupt Vectors which connect a device interrupt request to the device driver Interrupt Service Routine**
- **CSR addresses and contents indicate device type**
- **SYSGEN AUTO ALL will scan IO space to find devices and set up OpenVMS device drivers to communicate with the hardware**



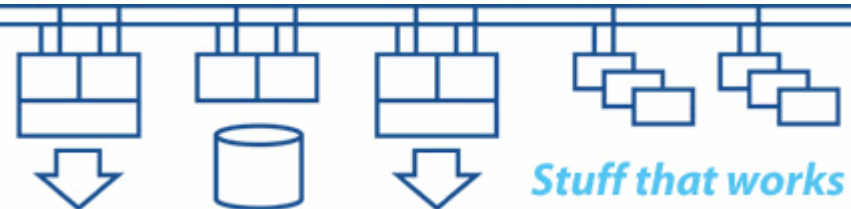
- **Console firmware**
- **Devices appear as a set of CSRs (Control and Status Registers) in physical memory - the IO space**
- **Devices have Interrupt Vectors which connect a device interrupt request to the device driver Interrupt Service Routine**
- **CSR addresses and contents indicate device type**
- **SYSMAN IO AUTO will scan IO space to find devices and set up OpenVMS device drivers to communicate with the hardware**



- **Itanium Processor Family architecture uses ACPI (Advanced Configuration and Power Interface) for device detection by firmware**
- **Devices appear as a set of CSRs (Control and Status Registers) in physical memory - the IO space**
- **Devices have Interrupt Vectors which connect a device interrupt request to the device driver Interrupt Service Routine. Device data obtained from ACPI data.**
- **ACPI data indicate device type**
- **SYSMAN IO AUTO will query ACPI data to find devices and set up OpenVMS device drivers to communicate with the hardware**



VAX (32 bit)

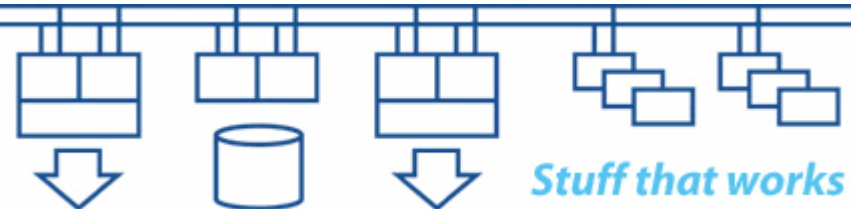


- **CDs:**
 - **Operating System (O.S.)**
 - **Layered Products (L.P.)**
- **Licence PAKs**
- **Installation requirements documentation (O.S. and L.P.)**
- **Hardware configuration information**
- **Console setup (SET BOOT DKA300 etc.)**

- **OpenVMS VAX V7.3 CD kit**
 - **DECnet-Plus V7.3**
 - **TCP/IP Services V5.3**
 - **Motif V1.2-5**
- **Licence data (PAKs)**
- **Base L.P. kits**
 - **DFU, LD, TSM, ZIP (from Freeware CDs)**
 - **DECnet-Plus V7.3 (from L.P. CDs)**
- **Patches**
- **Post install procedures**

- **Restore minimal install saveset (VMS073.B) to target system disc**
- **Boot target system disc minimal system**
- **Answer installation questions and proceed with installation process**
- **Install O.S. patches (read patch release notes)**
- **Set up system parameters ready for layered product installations (read product requirements)**
- **Install ‘base’ L.P. (e.g.: TCP/IP)**
- **Install ‘base’ L.P. patches**

Alpha (64 bit)



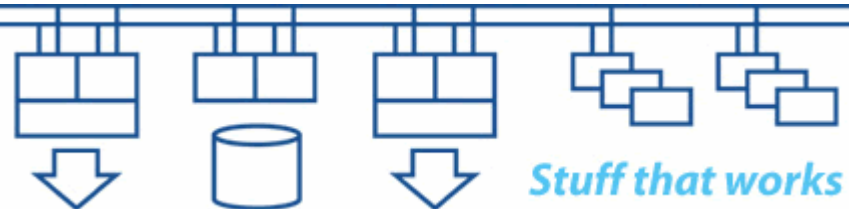
- **CDs:**
 - **Firmware Update (PALcode etc.) – V6.8 firmware CD is current**
 - **Operating System (O.S.)**
 - **Layered Products (L.P.)**
- **Licence PAKs**
- **Installation requirements documentation (O.S. and L.P.)**
- **Hardware configuration information**
- **Console setup (SET BOOTDEF_DEV DKA0 etc.)**

- **OpenVMS Alpha V7.3-2 / E8.2 CD kit**
 - **DECnet-Plus V7.3-2 / M8.2**
 - **TCP/IP Services V5.4 / T5.5**
 - **Motif V1.3-1 / K1.5**
- **Licence data (PAKs)**
- **Base L.P. kits:**
 - **DFU, LD, TSM, ZIP (from Freeware CDs)**
- **Patches**
- **Post install procedures**

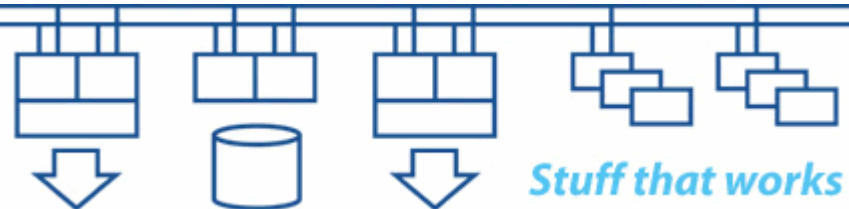
- **Boot firmware update CD**
- **Install firmware updates**
- **Boot OpenVMS installation CD**
- **Select installation menu (option 1)**
- **Answer installation questions**
- **Install O.S. patches**
- **Install ‘base’ L.P. patches (e.g. TCP/IP)**
- **Set up system parameters ready for layered product installations (read product requirements)**

Integrity (64 bit)

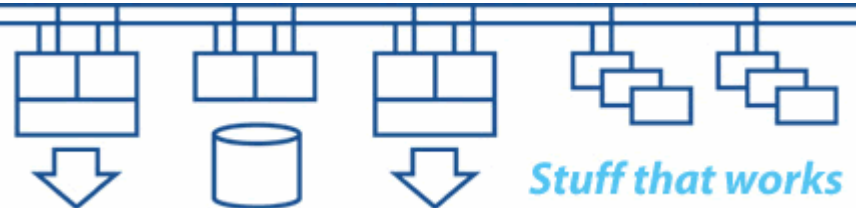
HP systems based on Intel Itanium Processor Family (Itanium II onwards)



- **DVDs / CDs / FTP:**
 - **Firmware Updates (MP console, BMC console etc.)**
 - **Operating System (O.S.)**
 - **Layered Products (L.P.)**
- **Installation requirements documentation (O.S. and L.P.)**
- **Hardware configuration information**
- **Consoles setup (MP console, BMC console)**
- **EFI configuration (VMS_LOADER etc.)**

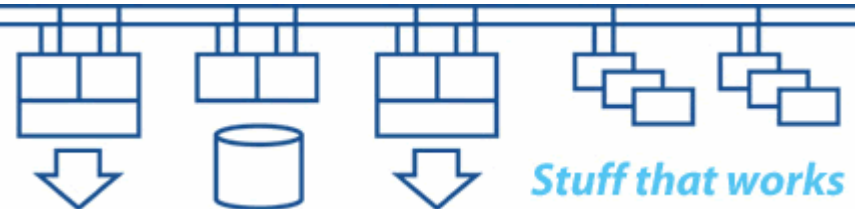


- **OpenVMS Integrity E8.2 DVD kit**
 - DECnet-Plus M8.2
 - TCP/IP Services T5.5
 - Motif K1.5
- **OS licences are different – “Operating Environment” based.**
- **Base L.P. kits:**
 - ZIP (from LP DVD)
- **Patches**
- **Post install procedures**



- **Read the documentation carefully, especially the console and EFI sections**
- **Install firmware updates**
- **Boot OpenVMS installation DVD**
- **Select installation menu (option 1)**
- **Answer installation questions**
- **Set up system parameters ready for layered product installations (read product requirements)**
- **Set up EFI boot menu for automatic boot – these can sometimes change when you use SET BOOTBLOCK or perform OpenVMS upgrades**

OpenVMS is OpenVMS, regardless of hardware platform



Configure base system:

- **Decompress library files**
- **Edit / create startup files**
- **Configure network software**
- **Set up / move page/swap/dump files**
- **Set up other system components (e.g.: LAT, MOP)**
- **Set up queues and queue manager database**
- **Set up / move user authorisation files**
- **Set up login actions (SYLOGIN)**
- **BACK IT UP**

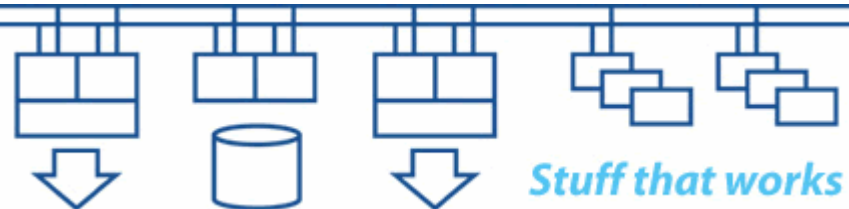
- **Backup and restore:**
 - **Image backups (/image/noalias)**
 - **File backups**
 - **Database backups (e.g.: Oracle)**
- **User management**
- **Hardware problem resolution (error logs etc.)**
- **Performance basics (AUTOGEN etc.)**

- **Security basics (Audit server, ACLs etc.)**
- **Disc management:**
 - **System log files (OPERATOR, NETSERVER etc.)**
 - **Version limits (file, directory, volume)**
 - **File and disc fragmentation**
 - **Quotas**
- **Queue management**
 - **Print queues (Generic and Execution)**
 - **Batch queues (Generic and Execution)**
 - **Common problems**

- **Use LD driver to create container files (built in to V7.3-2 and later, latent in V7.3-1, Freeware CD before that)**
- **Disable cache attribute for LD container files**
- **Container size as multiple of 4 blocks (allows for 2048 byte IDE/ATAPI native sector size)**
- **INIT/CLUSTER=4/ERASE/NOHIGH**
- **Use CDRECORD on VAX / Alpha (Freeware CD)**
- **Copy container to PC (FTP in binary mode) and burn CD / DVD as an image file (not ISO with 2048 byte sectors), eg: Nero Burning ROM**

- Use **STABACKIT** on **VAX**
- Use **AXPVMS\$PCSI_INSTALL_MIN.COM** on **Alpha**
- Use **SET BOOTBLOCK** command on **E8.2**
- **Integrity - EFI** needs to see the native sector size of the device, so:
 - **IDE/ATAPI** optical media needs a **2048** byte boot block
 - **Integrity SCSI** optical media needs a **512** byte boot block
- **VAX & Alpha** use **512** byte boot blocks
- Set **WLKSYSDSK** parameter appropriately

Q & A

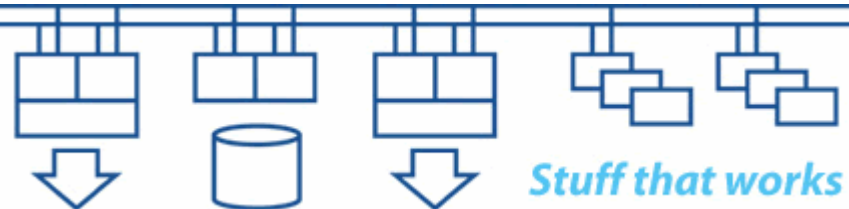


New date: week of June 6th 2005
Sheraton Tara hotel, Nashua, New England
(Original date was week of March 20th)

- **An “Ambassadors” style week for non-HP staff.**
- **An intense week of learning, information gathering and meeting people. Well worth going to.**
- **Contact Sue Skonetski (susan.skonetski@hp.com).**
- **See <http://www.hp.com/go/openvms/bootcamp>**

Most information is on the web somewhere:

- **OpenVMS main site** (<http://www.hp.com/go/openvms>)
- **Technical Journal** (<http://www.hp.com/go/openvms/journal>)
- **FAQ** (<http://www.hp.com/go/openvms/wizard>)
- **Ask The Wizard** (<http://www.hp.com/go/openvms/wizard>)
- **Integrity Servers** (<http://www.hp.com/go/integrity>)
- **Alpha Servers** (<http://h18002.www1.hp.com/alphaserver/>)
- **HP DSPP** (<http://www.hp.com/dspp>)





Contacts

HPUG – 8th Dec. 2004

OpenVMS on Integrity

Colin Butcher, XDelta Limited

Office: +44 117 904 8209

E-mail: colin.butcher@xdelta.co.uk

Web: <http://www.xdelta.co.uk>

Martin Riley, HP

E-mail: martin.riley@hp.com

Web: <http://www.hp.com/go/openvms>



Copyright © Colin Butcher, XDelta Limited, December 2004

