CHARON-AXP V4.6 for Linux - Users Guide

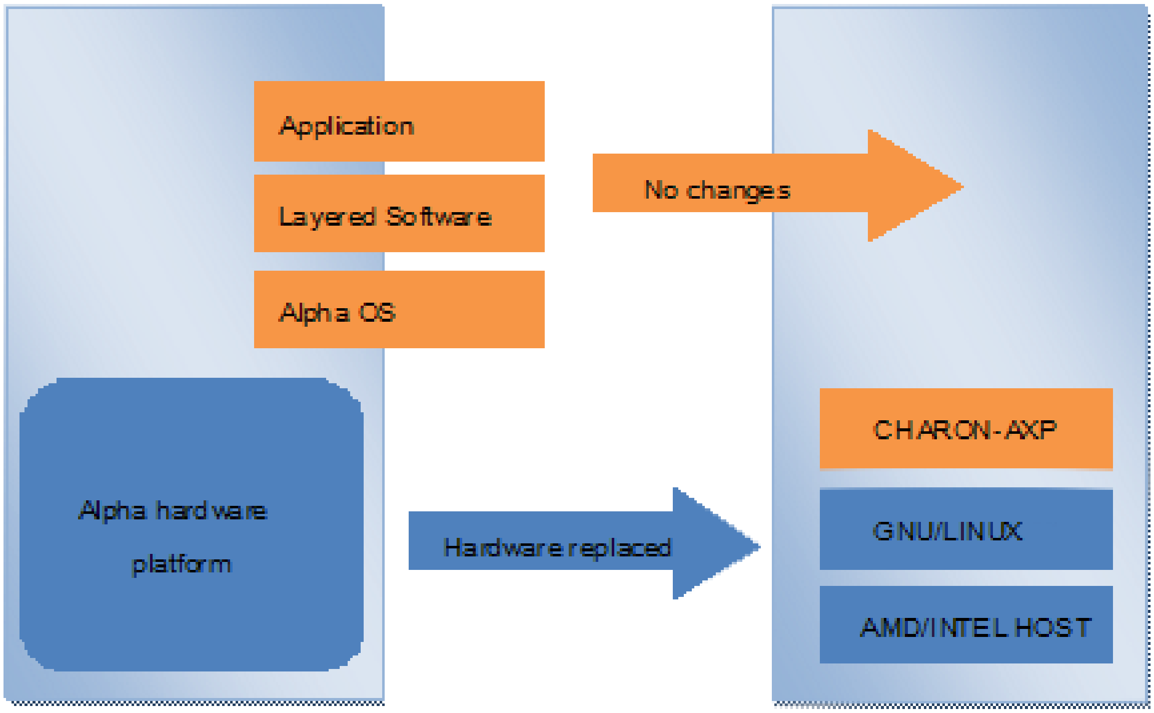
Contents

Introduction

Table of Contents

General Description

HP Alpha Hardware Virtualization allows users of HP Alpha (Previously known as DIGITAL Alpha) computers to move application software and user data to a modern Intel or AMD based x64 compatible platform without having to make changes to software and data. HP Alpha Hardware Virtualization is a software solution that replaces HP Alpha hardware.



This approach is best understood when the HP Alpha Hardware Virtualization Software is viewed as a special interface between the old HP Alpha software and a new hardware platform. Basically, the CHARON software presents a HP Alpha hardware interface to the original HP Alpha software, so that the existing software cannot detect a difference. This means no changes have to be made to the existing software. User programs and data can be copied to a new modern industry standard server (64-bit Intel or AMD) and continue to run for many more years.

The HP Alpha virtualization software is designed to replace single and multi-CPU HP Alpha computer systems, including:

- AlphaServer 400
- AlphaServer 800
- AlphaServer 1000
- AlphaServer 1000A
- AlphaServer 1200
- AlphaServer 2000
- AlphaServer 2100
- AlphaServer 4000
- AlphaServer 4100
- AlphaServer DS10
- AlphaServer DS10L
- AlphaServer DS15
- AlphaServer DS20
- AlphaServer DS25

Back to Table of Contents

The principles of HP Alpha Hardware Virtualization

Virtualized hardware

CHARON-AXP virtualizes various HP Alpha architectures and meets or exceeds the performance level of these HP Alpha systems when run on the recommended hardware platform. Our HP Alpha emulator product is currently available in the following variants:

CHARON-AXP/4100 includes:

- AlphaServer 400
- AlphaServer 800
- AlphaServer 1000
- AlphaServer 1000A
- AlphaServer 1200
- AlphaServer 2000
- AlphaServer 2100
- AlphaServer 4000
- AlphaServer 4100

CHARON-AXP/DS10 includes:

- AlphaServer DS10
- AlphaServer DS15

CHARON-AXP/DS20 includes:

- AlphaServer DS20
- AlphaServer DS25

CHARON-AXP/ES40 includes:

- AlphaServer ES40
- AlphaServer ES45

CHARON-AXP/GS80 include:

- AlphaServer GS80

CHARON-AXP/GS160 includes:

- AlphaServer GS160

CHARON-AXP/GS320 includes:

- AlphaServer GS320

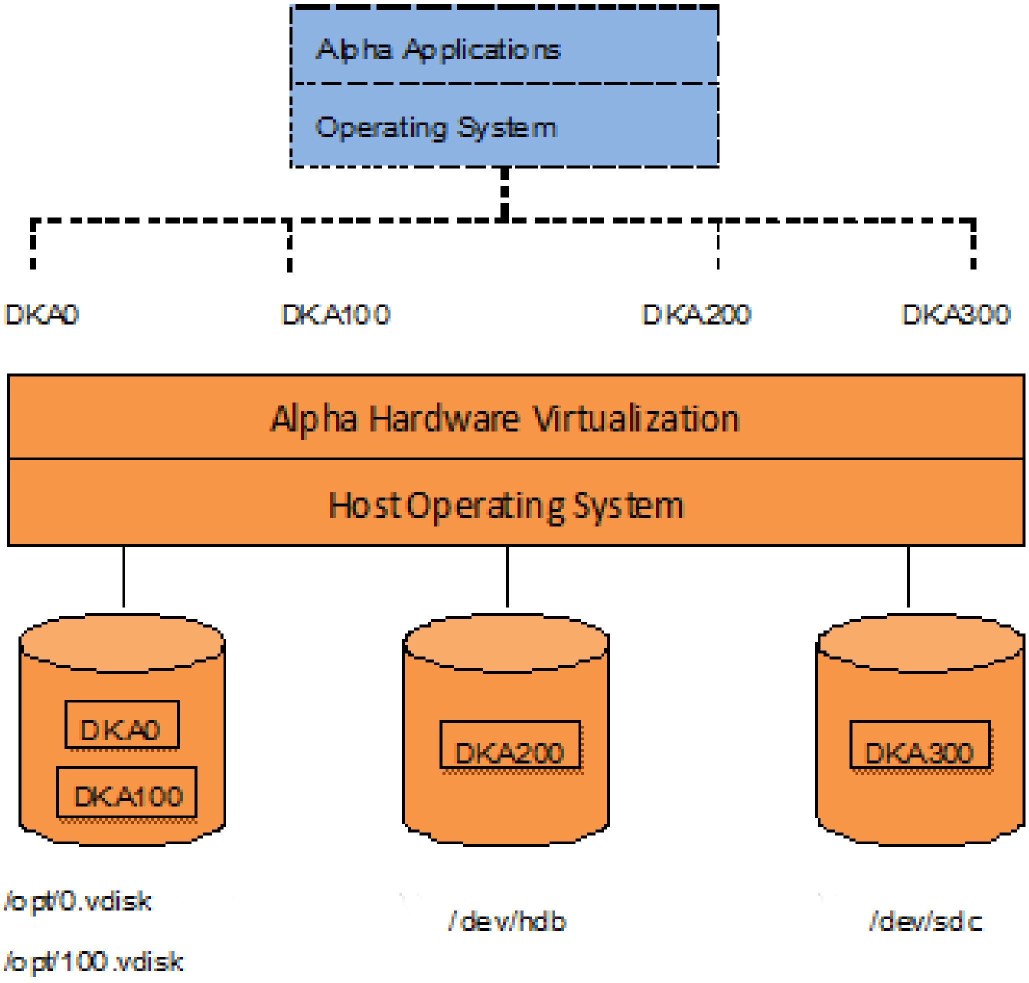
The following table explains which hardware boards CHARON virtualizes:

Subsystem	Covered HP Alpha hardware
Serial Lines Controllers	On-board serial line port COM2
IDE/ATAPI CD-ROM Controller	Virtual Acer Labs 1543C
PCI Fibre Channel Controller	KGPSA-CA
PCI SCSI Controller	KZPBA
PCI Network Controllers	DE435, DE450, DE500AA, DE500BA, DE602, DE602AA

Back to Table of Contents

Host platform

The Virtualization Software presents standard HP Alpha devices to the HP Alpha operating system, allowing the OS to function as though it were still running on a HP Alpha computer. For example, virtual disk container files in a directory or physical devices of the host Linux platform are presented by the Virtualization Software to the HP Alpha OS as emulated SCSI disks attached to a PCI SCSI adapter.



With the use of current storage technology, disks do not have to be physically attached to the Host platform, they can also reside on a SAN or iSCSI storage structure.

A similar translation process is also valid for other emulated hardware devices.

Back to Table of Contents

Conventions

Throughout the document(s) these conventions are followed:

Notation	Description
$	The dollar sign in interactive examples indicates an operating system prompt for VMS. The dollar sign can also indicate non superuser prompt for UNIX / Linux.
#	The number sign represents the superuser prompt for UNIX / Linux.
>	The right angle bracket in interactive examples indicates an operating system prompt for Windows command (cmd.exe).
`User input`	Bold monospace type in interactive examples indicates typed user input.
`<`**`path`**`>`	Bold monospace type enclosed by angle brackets indicates command parameters and parameter values.
`Output`	Monospace type in interactive examples, indicates command response output.
[ ]	In syntax definitions, brackets indicate items that are optional.
...	In syntax definitions, a horizontal ellipsis indicates that the preceding item can be repeated one or more times.
`dsk0`	Italic monospace type, in interactive examples, indicates typed context dependent user input.
↵	This symbol represents the Enter key without typed user input. Used, for example, to tell the user to select the default value by pressing enter.

The following definitions apply:

Term	Description
Host	The system on which the emulator runs, also called the CHARON server
Guest	The emulated system, in which the Tru64 or OpenVMS system runs

CHARON-AXP for Linux installation

Table of contents

Introduction

CHARON-AXP products are distributed in form of archive TAR.GZ files that contain RPM modules for different components. Generally it is recommended to install all the RPM modules, but it is possible to omit some RPMs if they are not needed.

CHARON installation consists of the following steps:

- Host system checks (hardware and software) to ensure the host platform meets minimum CHARON-AXP installation requirements
- Installation of any 3rd party material, for example, utilities required for CHARON-AXP
- Extracting CHARON-AXP RPM modules from the TAR.GZ archive, and their individual installation
- Installation of CHARON-AXP license (hardware dongle or software license)
- CHARON-AXP host system configuration. It assumes creating a specific user, network configuration etc.

Let's go through CHARON-AXP installation sequence step by step.

Back to Table of Contents

Hardware Requirements

Number of CPU cores

Each CHARON-AXP emulated CPU requires a corresponding physical core. So the total number of the host CPUs must exceed the number of emulated CPUs since some of the host CPUs must be dedicated to serving CHARON I/O operations and host operating system needs. If several CHARON instances run in parallel, the required number of CPU cores is summarized.

The following table shows the minimum and recommended number of CPUs required for each product for one instance (note that each product instance is able to run on 2 CPU cores hosts, but this configuration does not support emulation of all the virtual CPUs):

CHARON-AXP product	Minimal number of host CPU cores	Recommended number of host CPU cores
CHARON-AXP/4100	2	2
CHARON-AXP/DS10	2	2
CHARON-AXP/DS20	4	4
CHARON-AXP/ES40	6	8
CHARON-AXP/GS80	10	16
CHARON-AXP/GS160	18	32
CHARON-AXP/GS320	34	48

Hyperthreading must be switched off completely. Disable hyperthreading in the BIOS settings of the physical host or, for a VMware virtual machine, edit the virtual machine properties, select the Resources tab then select Advanced CPU. Set the Hyperthreaded Core Sharing mode to *None*.

CPU type and speed

Since CHARON-AXP utilizes LAHF instruction in HP Alpha CPU emulation please avoid usage of early AMD64 and Intel 64 CPUs in CHARON host system since they lack it. AMD introduced the instruction with their Athlon 64, Opteron and Turion 64 revision D processors in March 2005 w hile Intel introduced the instruction with the Pentium 4 G1 stepping in December 2005.

Concerning CPU speed the general recommendation is that higher CPU frequency is better since it allows better emulated HP Alpha performance. The minimal recommendation is at least 3 GHz.

Operative memory

The minimum host memory size depends on the amount of HP Alpha memory to be emulated and on the number of CHARON-AXP instances to be run on one host.
The minimum host memory is calculated according to the following formula:

The minimum host memory = (2Gb + the amount of HP Alpha memory emulated) per CHARON-AXP instance.

Disk storage

When installed, CHARON-AXP takes approximately *50* MB of disk space for its files, not counting any virtual HP Alpha disks/tapes (which appear as standard files).
When virtual HP Alpha disks/tapes are used to represent HP Alpha disk drives / magnetic tapes, the disk/tape image files have the same size as the equivalent HP Alpha diskhardware, regardless of their degree of utilization. So the total amount of a disk space required for CHARON-AXP can be calculated as a sum of the disk/tape images sizes plus 50 MB plus space required for the normal host system.

Ethernet adapters

CHARON-AXP networking assumes dedicated host Ethernet adapters; their number must be equal to the emulated adapters to be configured in CHARON-AXP. One adapter (optionally) can be left to the host for TCP/IP networking etc. It is also possible to use virtual network interfaces, but in consideration of performance, it is recommended to use physical ones only.

In case of VMware-based CHARON host it is mandatory to use "E1000" virtual network adapter. Please avoid usage of "E1000E" adapter since it may lead to problems with some TCP/IP services!

Back to Table of Contents

Software Requirements

- Fedora Core Linux version 20
- Red Hat Enterprise Linux version 6.2-6.5, 7 (excluding 'ncu' utility)

Back to Table of Contents

Host system preparation

In case network-wide license (red dongle or software license) is going to be used, do the following:

- *On server side (where network license will reside)*: open port 1947 for both TCP and UDP
- *On clients side*: open UDP ports 30000-65535
- *Both on server and client sides*: setup default gateway

Please consult with your Linux User's Guide on details.

> If stricter firewall rules are required, it is possible to open the ports 30000-65535 and 1947 only for the "/usr/sbin/hasplmd" daemon.

Back to Table of Contents

Before installation

1. Login as system administrator ("*root*") to the host system. Because Sentinel HASP runtime relies on 32-bit compatibility libraries to run on Linux, the 32-bit compatibility libraries should be installed first:

   ```
   # yum install glibc.i686
   ```

2. Create a special directory for the CHARON-AXP distribution and copy the TAR.GZ files there. Set this directory as the default with a "cd" command as shown in the following example:

   ```
   # mkdir /charon_dist
   # cp /temp/charon-axp-4.6-16803.68704.fc20.tar.gz /charon_dist
   # cd /charon_dist
   ```

Back to Table of Contents

Distribution preparation

1. Extract the content of the distribution TAR.GZ files to the current directory:

   ```
   # tar -xvzf charon-axp-<PRODUCT>-<VER>-<BN>.<VC>.<ZZ>.tar.gz
   ```

   where:

	Description
PRODUCT	Name of CHARON-AXP product, for example '*es40*'. ⚠ This field may be absent in recent kits
VER	Version of CHARON-AXP product, for example 4.6
BN	Build Number of CHARON-AXP product, for example 16803
VC	68704 - CHARON-AXP product vendor code
ZZ	CHARON-AXP target operating system identifier. For Fedora Core 20 '*ZZ*' value is '*fc20*', for Red Hat Linux v6.2-6.5 and 7 the value is '*el65*'

   Example (The "PRODUCT" field is absent)**:**

   ```
   # tar -xvzf charon-axp-4.6-16803.68704.fc20.tar.gz
   ```

   As result, a new directory "*charon-axp-<PRODUCT>-<VER>-<BN>.<VC>.<ZZ>"* will be created.

2. Switch to the directory, created by "*tar*" on the previous step:

   ```
   # cd charon-axp-<PRODUCT>-<VER>-<BN>.<VC>.<ZZ>
   ```

   Example:

   ```
   # cd charon-axp-4.6-16803.68704.fc20
   ```

3. The "main" RPM files of CHARON-AXP products are:

File name	Description
charon-axp-4100-VER-BN.VC.ZZ.x86_64.rpm	CHARON-AXP/4100
charon-axp-ds10-VER-BN.VC.ZZ.x86_64.rpm	CHARON-AXP/DS10
charon-axp-ds20-VER-BN.VC.ZZ.x86_64.rpm	CHARON-AXP/DS20
charon-axp-es40-VER-BN.VC.ZZ.x86_64.rpm	CHARON-AXP/ES40
charon-axp-gs80-VER-BN.VC.ZZ.x86_64.rpm	CHARON-AXP/GS80
charon-axp-gs160-VER-BN.VC.ZZ.x86_64.rpm	CHARON-AXP/GS160
charon-axp-gs320-VER-BN.VC.ZZ.x86_64.rpm	CHARON-AXP/GS320

   The distribution directory also contains the following RPM files with additional material, libraries and utilities:

File name	Description
aksusbd-2.4-1.i386.rpm	HASP Run-time
charon-base-VER-BN.VC.ZZ.x86_64.rpm	CHARON Libraries
charon-hasp-VER-BN.VC.ZZ.x86_64.rpm	hasp_srm_view utility and specific libraries for software licenses support

charon-ncu-VER-BN.VC.ZZ.x86_64.rpm	Network Configuration Utility ("ncu")

Example:

```
# ls
aksusbd-2.4-1.i386.rpm
charon-base-4.6-16803.68704.fc20.x86_64.rpm
charon-hasp-4.6-16803.68704.fc20.x86_64.rpm
charon-axp-4100-4.6-16803.68704.fc20.x86_64.rpm
charon-axp-ds10-4.6-16803.68704.fc20.x86_64.rpm
charon-axp-ds20-4.6-16803.68704.fc20.x86_64.rpm
charon-axp-es40-4.6-16803.68704.fc20.x86_64.rpm
charon-axp-gs80-4.6-16803.68704.fc20.x86_64.rpm
charon-axp-gs160-4.6-16803.68704.fc20.x86_64.rpm
charon-axp-gs320-4.6-16803.68704.fc20.x86_64.rpm
charon-ncu-4.6-16803.68704.fc20.x86_64.rpm
```

Back to Table of Contents

Installation

Issue the following command to install all the RPMs in the directory:

```
# yum install *.rpm
```

Network Configuration Utility ("ncu") package cannot be installed on Red Hat Linux 7.

To install all CHARON-AXP packages except the "ncu" one issue:

```
yum install *.rpm --exclude=*ncu*
```

Enter "**y**" to agree to install all the listed packages.

Example:

```
Dependencies Resolved
================================================================================
Package Arch Version Repository Size
================================================================================
Installing:
aksusbd i386 2.4-1 /aksusbd-2.4-1.i386 3.0 M
charon-base x86_64 4.6-16803 /charon-base-4.6-16803.68704.fc20.x86_64 15 M
charon-hasp x86_64 4.6-16803 /charon-hasp-4.6-16803.68704.fc20.x86_64 3.3 M
charon-axp-es40
x86_64 4.6-16803 /charon-axp-es40-4.6-16803.68704.fc20.x86_64 17 M
charon-ncu x86_64 4.6-16803 /charon-ncu-4.6-16803.68704.fc20.x86_64 15 M
Transaction Summary
 ================================================================================
Install 5 Packages
Total size: 39 M
Installed size: 39 M
Is this ok [y/N]: y
```

Check that the installation process has completed successfully.

Example:

```
Downloading Packages:
Running Transaction Check
Running Transaction Test
Transaction Test Succeeded
Running Transaction
Installing : aksusbd-2.4-1.i386 1/5
Starting aksusbd (via systemctl): [ OK ]
Installing : charon-base-4.6-16803.x86_64 2/5
Installing : charon-hasp-4.6-16803.x86_64 3/5
Installing : charon-axp-es40-4.6-16803.x86_64 4/5
Installing : charon-ncu-4.6-16803.x86_64 5/5
Verifying : charon-base-4.6-16803.x86_64 1/5
Verifying : charon-hasp-4.6-16803.x86_64 2/5
Verifying : aksusbd-2.4-1.i386 3/5
Verifying : charon-axp-es40-4.6-16803.x86_64 4/5
Verifying : charon-ncu-4.6-16803.x86_64 5/5
Installed:
aksusbd.i386 0:2.4-1 charon-base.x86_64 0:4.6-16803
charon-hasp.x86_64 0:4.6-16803 charon-axp-es40.x86_64 0:4.6-16803
charon-ncu-4.6-16803

Complete!
```

Re-login (as "*root*") to apply *PATH* settings or execute the following command:

```
# . /etc/profile.d/charon_*
```

Note that Network Configuration Utility ("ncu") package has the following dependencies:

- ethtool
- bridge-utils
- tunctl
- net-tools
- NetworkManager

During "ncu" installation using "yum", these packages will be installed automatically if some of them are absent on the host system.

Back to Table of Contents

CHARON-AXP home directory

By default CHARON is installed in the "/opt/charon" directory. It has the following subdirectories:

Directory	Description
/bin	Contains all executables
/cfg	Contains templates of configuration files
/lib	Contains product libraries
/doc	Contains documentation
/log	Contains log files
/disks	Contains disk containers
/drivers	Contains CHARON drivers

The most important at this stage is the "/cfg" directory since it contains template configuration files with examples of typical configuration parameters and commentaries. We will pay our attention to this subject in the next chapter.

Back to Table of Contents

Specific user account creation

Create a specific account "*charon*" for running CHARON:

```
# useradd -G disk,tape,cdrom,dialout,lock -c "Charon User" -m charon
# passwd charon
```

Any existing user can also be used to run CHARON. In this case issue the following command to include this existing user to specific groups:

```
# usermod -G disk,tape,cdrom,dialout,lock -g <user name> <user name>
```

Example:

```
# usermod -G disk,tape,cdrom,dialout,lock -g tommy tommy
```

The specific account created above does not allow to use physical consoles "`/dev/tty<N>`" as CHARON consoles. If you plan to map CHARON console to "`/dev/tty<N>`" use only "root" account for CHARON running.

Back to Table of Contents

License installation

Regular HASP USB dongle

If CHARON license represents a regular USB dongle just connect it to the host USB port.

If CHARON host is accessed remotely please note that regular HASP licenses cannot be displayed and used in this case. As workaround it is possible to install CHARON as daemon. This procedure will be described later.

Network HASP USB dongle

If CHARON license is a network license (red USB dongle) it is possible either to connect it to the host USB port (to use it locally providing it to other hosts on local network in the same time) or to install it on some local network "server" for remote access from this particular host.

In case of remote usage:

- Copy aksusbd-2.4-1.i386.rpm and charon-hasp-4.6-<*build*>.68704.<*OS identifier*>.x86_64.rpm files (see above) to the server to some directory, for example "/temp"
- Login as "*root*" to the server
- Switch to that directory
- Install the copied file using "yum"
  Example:

  ```
  # cd /temp
  # yum install aksusbd* charon-hasp-*
  ```

- Connect the network HASP dongle to the server USB port.

Network HASP (red dongles) licenses have no restrictions with remote access

Software license

If CHARON license is a software license (SL) it is required to install it on the host using the following procedure:

1. Run hasp_srm_view utility in the following way to get the host fingerprint file ("my_host.c2v" in this example):

   ```
   # hasp_srm_view -fgp my_host.c2v
   ```

2. Send the resulting file to STROMASYS. In return STROMASYS will provide you with a "*.v2c" file, for example "your_license.v2c"
3. Copy the received "your_license.v2c" file to CHARON host to any folder then invoke the system default web browser and enter URL http://localhost:1947 to display "**Sentinel Admin Control Center**" (**ACC**) web interface. This interface allows you to view and manage CHARON licenses.
4. In the **ACC** use the following menu items: first "**Browse**" for the "your_license.v2c" file and then secondly "**Apply File**"
5. Ensure that the software license appears now in the "**Sentinel Keys**" menu of the **ACC**.

Alternatively it is also possible to use "hasp_update" utility for applying ".v2c" file.

Network-wide software licenses have no restrictions with remote access, whereas regular software licenses cannot be displayed and used in this case

So called "Provisional" (demo) license does not require collecting fingerprint. For its installation proceed right with the action (4) of the sequence above

Back to Table of Contents

License validity verification

Check available CHARON license validity. To do that invoke the hasp_srm_view utility to make sure that CHARON license is visible and looks Ok:

- Text of the license is displayed correctly by the hasp_srm_view utility, no error messages are shown
- Content of the license looks correct. For example license number, major and minor versions, minimum and maximum build numbers, CHARON-AXP products and allowed hardware (CHARON-AXP models) should be checked. More details on the license content can be found in the CHARON-AXP Licensing chapter of this Guide.

Example:

```
# hasp_srm_view

License Manager running at host: XEON4WAYW7
License Manager IP address: 192.168.1.22

HASP Net key detected

The Physical KeyId: 354850588
CHARON Sentinel HASP License key section
Reading 4032 bytes

License Manager running at host: XEON4WAYW7
License Manager IP address: 192.168.1.22

The License Number: nes
The License KeyId: 354850588
The Master KeyId: 1712849125
Release date: 16-JAN-2014
Release time: 17:53:41
Update number: 8
End User name: Net-Time
...
```

Reminder: If CHARON host is accessed remotely please note that regular HASP licenses cannot be displayed and used in this case. As workaround it is possible to install CHARON as daemon. This procedure will be described later.

Troubleshooting

If CHARON license content cannot be displayed by hasp_srm_view utility or it is incorrect, check the license is available and correctly used:

1. Invoke the system default web browser and enter URL http://localhost:1947 to display "**Sentinel Admin Control Center**" (**ACC**) web interface.
2. Click on "**Sentinel Keys**" link to open up "**Sentinel Keys Section**" page
3. Make sure that one and only one CHARON HASP or SL license is present.

Problem	Action
No license is displayed	Make sure that all the recommendations above about remote access to the host are fulfilled (if remote access takes place), HASP USB key is not broken and its LED indicator is lit (meaning that it is used by the host).
Only one License key / SL is seen and its content is incorrect	Contact STROMASYS as soon as possible.
Several License keys / SLs are displayed	Remove all of them and leave only the one provided by STROMASYS for just installed version of CHARON.

Removing licenses can be done by physical disconnection of the corresponding USB HASP keys from CHARON host and physical disconnection of the network HASP keys from all hosts on local network (or by disabling remote access to network licenses from CHARON host - see detailed explanations below). Software licenses can also be uninstalled with hasp_srm_view utility "*-tfr*" option in the following way:

```
# hasp_srm_view -tfr <Key ID>
```

Example:

```
# hasp_srm_view -tfr 12345678
```

It is also possible to disable access to network licenses if just a local license must be used: Click on "**Configuration**" link to open up "**Configuration for Sentinel Manager**" page. Uncheck "**Allow Access to Remote Licenses**" and "**Broadcast Search for Remote Licenses**" checkboxes from the "**Access to Remote License Managers**" tab, then press "**Submit**" button to apply changes.

It is also possible to leave several licenses available to CHARON-AXP at the same time, but in this case you have to specify in CHARON-AXP configuration file what license must be used.

Example:

```
set session license_key_id[0]=1877752571
```

It is also possible to have one "main" and one "backup" licence in case if the main license becomes not accessible:

```
set session license_key_id[0]=1877752571 license_key_id[1]=354850588
```

CHARON-AXP checks its licences from time to time starting from main license and if it is not accessible it tries to access backup license

Back to Table of Contents

Network configuration

In most cases it is assumed that CHARON will use network. In this case some important steps must be performed, since CHARON requires a dedicated network interface cleared from any other protocols including TCP/IP.

Two ways of network configuration are possible:

- Manual
- With a help of "ncu" utility

The second way is a way simplier, so use manual approach only in absence of "ncu" utility or impossibility to use it.

Configuration with NCU utility

Login as root. Type "ncu" and press Enter. The following menu will appear:

```
# ncu
CHARON Network Configuration Utility, STROMASYS (c) 2015 Version 1.4

Interfaces Dedicated to State
---------- ------------ ------------
eth0       host         connected to host
eth1       host         connected from host
lo         host         unmanaged from host

==================================================================
bridge name     bridge id        STP enabled            interfaces

select action:
1 - Dedicate to CHARON
2 - Release to host
3 - Create Bridge with TAPs
4 - Remove Bridge
5 - Print status
6 - Exit
:> 1
```

The utility lists available network interfaces (both physical and virtual) and indicates whether they are dedicated to the host or to CHARON and whether they are currently in use by host operating system.

"ncu" offers several options:

- Dedicate interface to CHARON (press "1")
- Release interface to host (press "2")
- Create a bridge between a chosen physical network interface and the Linux virtual network and create a number of virtual network interfaces (press "3")
- Remove the Linux virtual network and all the created virtual network interfaces (press "4")
- Print status (press "5") - use it to display status of network interfaces and the menu shown above
- Exit (press "6")

In the example above we see 2 network interfaces - "eth0" and "eth1", both of them are dedicated to host, but host uses only the interface "eth0".

Let's dedicate the interface "eth1" to CHARON-AXP.

Enter "1", then type "eth1" and press Enter:

```
Specify the interface to dedicate to CHARON:eth1
Turning off offloading for eth1.. Please wait

select action:
1 - Dedicate to CHARON
2 - Release to host
3 - Create Bridge with TAPs
4 - Remove Bridge
5 - Print status
6 - Exit
:> 5
```

Now the interface "eth1" is dedicated to CHARON-AXP:

```
Interfaces Dedicated to State
---------- ------------ ------------
eth0       host         connected to host
eth1       CHARON       disconnected from host
lo         host         unmanaged from host

==================================================================
bridge name     bridge id        STP enabled
interfaces

select action:
1 - Dedicate to CHARON
2 - Release to host
3 - Create Bridge with TAPs
4 - Remove Bridge
5 - Print status
6 - Exit
```

Enter "6" to return to console prompt.

Now "eth1" can be used by CHARON-AXP.

Manual Configuration

Choosing network interface

To choose an interface to be used for CHARON networking do the following:

```
# ifconfig
eth0 Link encap:Ethernet HWaddr 00:60:52:0A:A9:1E
...
eth1 Link encap:Ethernet HWaddr 00:C0:26:60:FB:15
...
eth2 Link encap:Ethernet HWaddr 00:1A:92:E1:3F:7F
```

Choose some interface to be used by CHARON, for example "*eth1*"

Designation of network interface to CHARON

To designate the chosen interface to CHARON open up the file "/etc/sysconfig/network-scripts/ifcfg-eth*N*" (where *N* is the number of the interface to be used for CHARON, in our case it is "1") and make sure that all the IP-setup related parameters are removed. Basically the file must look like this ("eth1" is used as example):

```
DEVICE="eth1"
HWADDR="00:06:2B:00:6A:87"
NM_CONTROLLED="no"
ONBOOT="no"
```

Switching off the offload parameters

A first step is to find what additional parameters are currently set to "*on*" on the host network adapter to be used by CHARON. To do that issue:

```
# ethtool -k <device>
```

Example:

```
# ethtool -k eth1
Offload parameters for eth1:
rx-checksumming: on
tx-checksumming: on
scatter-gather: on
tcp-segmentation-offload: off
udp-fragmentation-offload: off
generic-segmentation-offload: on
generic-receive-offload: off
large-receive-offload: off
```

Then use ethtool to switch off all the offload parameters:

```
# ethtool -K <device> <parameter> off
```

Example:

```
# ethtool -k eth1
Offload parameters for eth1:
rx-checksumming: on
tx-checksumming: on
scatter-gather: on
tcp-segmentation-offload: off
udp-fragmentation-offload: off
generic-segmentation-offload: on
generic-receive-offload: off
large-receive-offload: off
```

For the example above let's create a temporary file containing the commands to be run on system startup, since the offload parameters must be switched off on each reboot:

```
ethtool -K eth1 rx off
ethtool -K eth1 tx off
ethtool -K eth1 sg off
ethtool -K eth1 gso off
ethtool -K eth1 gro off
```

Let's suppose that the name of the file is "offload_off_eth1.txt". In this case running it on system startup can be done in the following ways:

On Red Hat Linux:

```
# cat offload_off_eth1.txt >> /etc/rc.d/rc.local
```

On Fedora Core:

```
# echo '#!/usr/bin/bash' > /etc/rc.d/rc.local       ⚠ Do not use double quotes, use single ones
# cat offload_off_eth1.txt >> /etc/rc.d/rc.local
# chmod 755 /etc/rc.d/rc.local
# ln -s /usr/lib/systemd/system/rc-local.service /etc/systemd/system/rc-local.service
# systemctl daemon-reload
```

Back to Table of Contents

Final steps

- Reboot the host system to apply the offload parameters switching off
- Login as user "*charon*"

Back to Table of Contents

Upgrade to new version

To upgrade already installed CHARON-AXP kit to more recent one:

1. Ensure your license allows you to upgrade. If not, please generate a C2V file and send it to STROMASYS for update. See CHARON-AXP for Linux utilities - 'hasp_srm_view' utility
2. Prepare the new kit RPM files as it is described in "Before Installation" and "Distribution preparation" sections.
3. Stop all running CHARON-AXP instances.
4. Make sure that no template files (i.e. "es40.cfg.template") have been used for your specific configuration. Otherwise copy those files to some save place.
5. Login as "root" user.
6. Proceed with the same instructions on the new kit installation as described in "Installation" section.
   ⓘ Note that this time "yum" will request you to confirm the existing kit update. Confirm that.
7. If a new license is required for the new CHARON-AXP kit install it as described in "License installation" section. Otherwise just skip this step.
8. Start all the CHARON-AXP services stopped on the step (2).

> The upgrade procedure above is applicable only to CHARON-AXP installed kits starting from the V4.6 Build 16803. For older kit use the standard deinstallation of the old kit and then the standard installation of the new one sequence instead of the step (6).

Back to Table of Contents

Running CHARON-AXP for Linux

Table of Contents

Back to Table of Contents

CHARON-AXP symbolic links

Use the following symbolic link to run different models of CHARON-AXP:

Link name	Emulator to run
as400	AlphaServer 400
as800	AlphaServer 800
as1000	AlphaServer 1000
as1000a	AlphaServer 1000A
as2000	AlphaServer 2000
as2100	AlphaServer 2100
as4000	AlphaServer 4000
as4100	AlphaServer 4100
ds10	AlphaServer DS10
ds10l	AlphaServer DS10L
ds15	AlphaServer DS15
ds20	AlphaServer DS20
ds25	AlphaServer DS25
es40	AlphaServer ES40
es45	AlphaServer ES45
gs80	AlphaServer GS80
gs160	AlphaServer GS160
gs320	AlphaServer GS320

Back to Table of Contents

Running CHARON-AXP emulators

It is possible to run one or several instances of CHARON-AXP at the same time if your license allows it.

In case of multiple instances, please use only absolute paths and unique names to all the files referenced in the configuration file of each

CHARON-AXP instance (log, toy clock, rom files and all the other data such as disk images - all these objects to be discussed later in this document). Also, hardware devices (e.g., CD-ROM) may be used by only one instance at a time (not shared).

For example:

```
...

set session log="/CharonInstances/1st_es40.log"
set rom container="/CharonInstances/1st_es40.bin"
set toy container="/CharonInstances/1st_es40.dat"

load KZPBA PKA scsi_id = 7
set PKA container[0]="/CharonInstances/1st_es40_boot_disk.vdisk"

...
```

Please refer to the next chapters for more details concerning CHARON-AXP configuration details.

Back to Table of Contents

Running from console

Copy required configuration template from "/opt/charon/cfg/" directory to some local file and set correct privileges to that file to be able to edit it:

```
$ cp /opt/charon/cfg/es40.cfg.template my_es40.cfg
$ chmod 644 my_es40.cfg
```

Now let's execute CHARON using this template configuration file:

```
$ es40 my_es40.cfg
```

You will see normal HP Alpha test sequence, followed by prompt sign (">>>"):

```
initializing ...

polling for units on kzpba0, slot 2, bus 0, hose 1 ...
pka0.0.0.2.1 PKA0 Q-Logic/ISP PCI SCSI HBA

... enter console

CHARON-AXP/ES40 for Linux (AlphaServer ES40 6/667), Version 4.6.16803
(C) 2009-2015 STROMASYS SA.
All rights reserved.

P00>>>
```

The next stage can be either installation of new HP Alpha/VMS system using a distributive provided by HP or transfer of data from some existing HP Alpha system. These possibilities will be discussed in details in next chapters.

If for some reason CHARON-AXP refuses to start please look for files with .log extension (CHARON-AXP log files) located in the directory from which CHARON-AXP starts, open them in some editing tool and analyze their content. In most cases those files contain very helpful information on what may possible went wrong.

To exit from CHARON-AXP emulator use the following methods:

<table>
<tr><th>Configuration</th><th>How to exit</th></tr>
<tr><td>No changes to the template configuration file</td><td>Type "<b>power off</b>" in CHARON console</td></tr>
<tr><td>Enable "<b>F6</b>" button in configuration file to trigger exit from CHARON:
<pre>
#--
#
# Uncomment to allow 'F6' to terminate the running emulator.
#
#--

set OPA0 stop_on = F6
</pre></td><td>Press "<b>F6</b>"</td></tr>
</table>

⚠ Please note that before stopping CHARON-AXP, one must shutdown the operating system running by CHARON-AXP.

Back to Table of Contents

Running as system service (daemon)

It is possible to run CHARON-AXP as a daemon. In this case CHARON-AXP process will be detached from its parent process and from the terminal window in which it runs.

Follow the description below to establish and run CHARON-AXP as daemon:

Installation and start of CHARON-AXP service

1. Copy the sample script "/opt/charon/bin/charon" ("/opt/charon/bin/charon.service" for Fedora Core Linux) to your home directory (Red Hat Linux) or to "/usr/lib/systemd/system/" directory (Fedora Core), for example:

Red Hat Linux	`$ cp /opt/charon/bin/charon /my_services/es40_service`
	`$ chmod 755 /my_services/es40_service`

Fedora Core	`$ cp /opt/charon/bin/charon.service /usr/lib/systemd/system/es40.service`
	`$ chmod 755 /usr/lib/systemd/system/es40.service`

2. Edit the renamed file to replace sample values of the following parameters, for example:

Red Hat Linux	`exec="/opt/charon/bin/es40"`
	`prog="my_es40"`
	`config="/my_services/es40-service.cfg"`

Fedora Core	`ExecStart=/opt/charon/bin/es40 -d /my_services/es40-service.cfg`
	`WorkingDirectory=/my_services`

3. Create and edit configuration file ("/my_services/es40-service.cfg" in the examples above) the way it was described before and make sure that the following pre-requisites are met:
   - OPA0 must be configured as virtual port or physical console, not as operator console, for example:

     ```
     load virtual_serial_line OPA0 port=10003
     #load operator_console OPA0
     ```

   - Use only absolute paths to log, toy clock, nvram files and all the other data such as disk images etc. The names of the references files must be unique too, for example

     ```
     ...

     set session log="/CharonInstances/1st_es40.log"
     set rom container="/CharonInstances/1st_es40.bin"
     set toy container="/CharonInstances/1st_es40.dat"

     set PKA container[0]="/CharonInstances/1st_es40_boot_disk.vdisk"

     ...
     ```

   - Make sure the same physical devices are not used by other CHARON-AXP daemons, same for the OPA0 console port number.

Once configuration file is ready issue the following commands (the specifics belongs to the examples above) to install and start CHARON-AXP as daemon:

Red Hat Linux	`# ln -sf /my_services/es40_service /etc/init.d/es40_service`
	`# chkconfig es40_service on`
	`# service es40_service start`

Fedora Core	`# systemctl enable es40.service`
	`# systemctl start es40.service`

Note that a certain delay may appear in finding network license by Sentinel Run-time on CHARON-AXP host system startup. So if CHARON-AXP service is starting automatically on host system startup it may report "License not found" error and exit.

This problem can be avoided by specifying "license_key_lookup_retry" parameter in the following way:

```
set session license_key_lookup_retry = "N [, T]"
```

where:

- N - Number of retries looking for license key (or keys)
- T - Time between retries in seconds. If not specified 60 seconds is used

Example:

```
set session license_key_lookup_retry = 5
```

In this example if the license key is not found during initial scan, CHARON-AXP will do 5 more attempts waiting 60 seconds between them.

See General Settings section for more details.

Back to Table of Contents

Stopping CHARON-AXP service

To stop CHARON-AXP daemon use the following command, for example:

Red Hat Linux	`# service es40_service stop`
Fedora Core	`# systemctl stop es40`

⚠ Please note that before stopping CHARON-AXP service, one must shutdown the operating system running by CHARON-AXP.

Back to Table of Contents

Removing CHARON-AXP service

To remove CHARON-AXP daemon use the following commands, for example:

Red Hat Linux	`# chkconfig es40_service off`
	`# chkconfig --del es40_service`
	`# rm -f /etc/init.d/es40_service`

Fedora Core	`# systemctl disable es40.service`
	`# rm -f /usr/lib/systemd/system/es40.service`

⚠ Please note that before removing CHARON-AXP service one must shutdown the operating system running by CHARON-AXP and then stop corresponding CHARON-AXP service.

ℹ Please refer to the next chapters for more details concerning CHARON-AXP configuration details

Back to Table of Contents

CHARON-AXP for Linux configuration

Table of Contents

Creation of your own configuration file using a template

By default, all the CHARON templates are located in the "/opt/charon/cfg" folder. Copy the appropriate template configuration file(s) to your home directory (or to any directory intended for CHARON-AXP). Name them meaningfully and set proper privileges.

For example:

```
$ cp /opt/charon/cfg/es40.cfg.template /my_charon_cfg/my_es40.cfg
$ chmod 644 /my_charon_cfg/my_es40.cfg
```

Please do not edit the original template configuration files since they can be updated or even removed on update/deinstallation of CHARON-AXP

Once the file has been created you can open it in your favorite editing tool and proceed with modification to reflect the exact features of the system you are going to emulate.

We will review all the parameters step by step issuing some recommendations and guidelines.

Note: lines preceeded by the comment sign "#" inside the configuration files will not be interpreted. You can use this sign to debug your configuration.

Back to Table of Contents

HP Alpha model specification

The first configuration statement is the specification of the exact HP Alpha hardware model to emulate, for example:

```
set session hw_model = AlphaServer_ES40
```

You must leave this line untouched.

If you create the CHARON-AXP configuration file from scratch it must be the very first uncommented line in the configuration file.

Back to Table of Contents

Configuration name

The next configuration statement is the "Configuration name" option:

```
#set session configuration_name = My_ES40
```

You can optionally uncomment this line to differentiate this CHARON-AXP instance from all others in a multi-instance environment. The configuration name can be any label that is meaningful.

The configuration name is reported in the log file and is used to set the log file name for rotating log (see further: Rotating log (default)).

Back to Table of Contents

Log file parameters

Execution of CHARON-AXP creates one log file or a set of log files reflecting the progress of its start-up and ongoing operation - start and end time of execution, system information, license and configuration details, warnings, reports on problems that may occur, etc. In case of possible problems either with the running CHARON-AXP or the emulated system configuration (such as the absence or malfunction of certain devices), the log file(s) is the primary source to be analyzed for troubleshooting. If it becomes necessary to contact Stromasys for support, the configuration and log files, plus the license number, will be requested to begin problem resolution.

Here is an example of a field test CHARON-AXP log file:

```
20150319:090329:INFO :0:000003A5:hexane.cxx(5357): session is loading built-in configuration "AlphaServer_ES40"...
20150319:090329:INFO :0:000003A6:hexane.cxx(5381): session has finished loading built-in configuration "AlphaServer_ES40".
20150319:090329:INFO :0:000003AA:hexane.cxx(5470): session is loading configuration file "es40.cfg"...
20150319:090329:INFO :0:000003AB:hexane.cxx(5500): session has finished loading configuration file "es40.cfg".
20150319:090329:INFO :0:0000032B:hexane.cxx(2588): Start request received.
20150319:090330:INFO :0:000003AC:hexane.cxx(1283): session's process affinity is 000000000000000F, system affinity is 000000000000000F.
20150319:090330:INFO :0:000003D1:hexane.cxx(1537): session's I/O domain affinity is 0000000000000001, CPU domain affinity is 000000000000000E
20150319:090330:INFO :0:000003BA:ll_sentine( 639): Looking for a license key ...
20150319:090330:INFO :0:000003DC:ll_sentine( 734): ... found license key 1422726238.
20150319:090330:INFO :0:0000024D:hexane.cxx(2729): STROMASYS SA, (C) 2009-2015
20150319:090330:INFO :0:00000350:hexane.cxx(2775): CHARON-AXP (AlphaServer ES40), V 4.6 B 16803, Jan 27 2015 / net-nes / 1422726238
20150319:090330:INFO :0:00000336:hexane.cxx(2802): The end user of this software has agreed to STROMASYS' Terms and Conditions for Software License and Limited Warranty, as described at: http://www.stromasys.com/pub/doc/30-17-033.pdf
20150319:090330:INFO :0:0000009D:hexane.cxx(2879): License info:
CHARON product code: "CHAXP-460xx-WI".
Licensed to: "net-nes".

20150319:090330:INFO :0:00000097:hexane.cxx(2888): OS Environment: Linux 3.17.8-200.fc20.x86_64 #1 SMP Thu Jan 8 23:26:57 UTC 2015 x86_64.
20150319:090330:INFO :0:00000098:hexane.cxx(2893): Host CPU: GenuineIntel, Family 6, Model 42, Stepping 1, Intel Xeon E312xx (Sandy Bridge), 1 Cores per Chip, 1 Threads per Core, at ~2593 MHz, 4 cpu's available
20150319:090330:INFO :0:00000099:hexane.cxx(2898): Host Memory: 4096Mb
20150319:090330:WARN :1:00000354:hexane.cxx(3050): 4 host CPUs detected but 8 recommended, performance might be limited.
20150319:090330:WARN :1:00000353:hexane.cxx(3068): The host system is below recommended specifications.
20150319:090332:ERROR:2:00000100:lnxpackpor( 753): (95) Operation not supported: EWA0: Failed to query for max frame size. Assume 1518.
20150319:090332:ERROR:2:00000101:lnxpackpor( 754): EWA0: Failed to query for link speed. Assume 10Mbps.
20150319:090332:WARN :1:000003D6:cpuemul.cx( 290): cpu_3 does not start as it unable to allocate host cpu
20150319:090333:INFO :0:0000034B:scsi_disk.( 566): PKA_0 is being set ONLINE
container = "/home/charon/Charon/test/performancecomparison-axp.vdisk"
20150319:090333:INFO :0:0000032C:hexane.cxx(2630): "AlphaServer_ES40" started.
20150319:090710:INFO :0:000003C3:21264_vms_( 43): CPU0 Halted (HALT INSTRUCTION @PC = FFFFFFFF800D3884)
20150319:090718:INFO :0:000003D7:hexane.cxx(4973): All virtual CPUs of "AlphaServer_ES40" have been stopped by now.
20150319:090718:INFO :0:0000032D:hexane.cxx(2674): "AlphaServer ES40" stop request received.
20150319:090718:INFO :0:0000014C:lnxpackpor( 416): EWA0: Stopping network interface ... please wait.
20150319:090718:INFO :0:00000348:ataunit.cx(1738): ide0 is being set OFFLINE
20150319:090718:INFO :0:00000348:ataunit.cx(1738): ide1 is being set OFFLINE
20150319:090718:INFO :0:0000032E:hexane.cxx(2692): Stopped.
```

The next group of parameters defines the name of the CHARON-AXP log file and how CHARON-AXP will use it:

```
#set session log_method = append
#set session log_method = overwrite
#set session log = "AlphaServer_ES40.log"
```

Rotating log (default)

By default CHARON-AXP utilizes a so-called "rotating log". This means that a new default log file is always created each time CHARON starts and can switch to another log file if the size of the log file exceeds 64Kb.

This mode is turned on if all the log parameters above are disabled (commented out) or the "session_log" parameter is pointing to an existing directory rather than to a file. If a directory is specified, the log files will be created in that directory.

Names of the rotating log files are composed as follows:

```
configuration_name-YYYY-MM-DD-hh-mm-ss-xxxxxxxxx.log
```

If the "Configuration name" parameter described before is omitted (commented out), the log name has the following format instead:

```
hw_model-YYYY-MM-DD-hh-mm-ss-xxxxxxxxx.log
```

Note that "xxxxxxxxx" is an increasing decimal number starting from "000000000" to separate log files with the same time of creation.

> Only existing directory can be used as a value of the "log" parameter.

Single log

Alternatively it is possible to use just a single log file. Uncomment the "set session log" line and specify the desired CHARON-AXP log file name. Optionally, a path can be added to the log file name. If the path is not specified, the log file is created in the directory from where the guest (emulated machine) is started.

The log file can be extended ("log_method = append") or overwritten ("log_method = overwrite") by CHARON-AXP.

Below is a specification of a CHARON-AXP log file located in the "/my_logs" directory which will be overwritten each time CHARON-AXP starts:

```
set session log_method = overwrite
set session log = "/my_logs/my_es40.log"
```

Back to Table of Contents

CPU affinity

This setting binds the running instance of the emulator CPUs to particular host CPUs.This should be used for soft partitioning host CPU resources or for isolating multiple CHARON instances on the same host from each other. By default the emulator instance allocates as many host CPUs as possible.

“Affinity” overrides the default and allows explicit specification of which host CPUs will be used by the instance. Affinity does not reserve the CPU for exclusive use.

```
set session affinity="0, 1, 2, 3"
```

The example above directs CHARON-AXP to use CPU 0,1,2 and 3.

If this parameter is omitted CHARON host will allocate available CPUs automatically.

⚠ Note that the number of the specified host CPUs must correspond to the number of the emulated CPUs (one host CPU for one emulated CPU; this value is specific for each HP Alpha model) and number of CPUs needed for CHARON application itself ("n_of_io_cpus").

Back to Table of Contents

Number of host CPUs dedicated to CHARON I/O

This setting reserves host CPUs (of those specified by “affinity” parameter, if any) for use by the emulator for I/O handling. By default the emulator instance reserves one third of available host CPUs for I/O processing (round down, at least one).

The “n_of_io_cpus” overrides the default by specifying the number of I/O host CPUs explicitly

Example:

```
set session n_of_io_cpus=2
```

The example above directs CHARON-AXP to use 2 CPUs for CHARON I/O operations.

⚠ Note that the number of the specified CPUs dedicated to CHARON I/O operations must correspond to the total number of available for CHARON CPUs (restricted by "affinity" parameter if needed) and the number of the virtual HP Alpha CPUs to be emulated.

Back to Table of Contents

Setting a specific HP Alpha model

CHARON-AXP allows to specify an exact model of HP Alpha.

For example for HP AlphaServer ES40 family the "es40.cfg" sample configuration file contains the following options:

```
#============================================================================
#
# AlphaServer ES40 6/500
#
#----------------------------------------------------------------------------

#set ace cpu_architecture = EV6
#set rom dsrdb[0] = 1816 system_name = "AlphaServer ES40 6/500"
#set rom version[1] = 1.98-4 version[2] = 1.92-5

#============================================================================
#
# AlphaServer ES40 6/667
#
#----------------------------------------------------------------------------

set ace cpu_architecture = EV67
set rom dsrdb[0] = 1820 system_name = "AlphaServer ES40 6/667"
```

Just uncomment the provided lines to apply a certain model (It is "AlphaServer ES40 6/667" in the example above).

Full description of the parameters and other models that can be also configured is available in the "Configuration details" chapter of this User's Guide.

Back to Table of Contents

Reducing number of emulated CPUs

If CHARON host contains not enough CPUs to emulate full range of the CPUs provided by a certain HP Alpha model it is possibe to direct CHARON-AXP to reduce number of the emulated Alpha CPUs in the configuration:

```
set session n_of_cpus=1
```

Back to Table of Contents

Setting system serial number

The next configuration option that can be applied is setting a specific system serial number instead of the default one:

```
set rom system_serial_number = SN01234567
```

Back to Table of Contents

TOY and ROM containers

The next objects to be configured are TOY and ROM containers (their presence depends on the HP Alpha model). It is always recommended to enable them. If a container file of the given name does not exist, CHARON-AXP will create it. It is recommended to specify the path for each file so that time and console parameters will be kept whatever the current directory is when starting the guest.

TOY means "Time of Year"; its container records time, date and some console parameters while CHARON-AXP is not running. To enable, uncomment the following line:

```
set toy container="clipper.dat"
```

The ROM container stores an intermediate state of the Flash ROM and some console parameters. It is highly recommended to define its location:

```
set rom container="clipper.bin"
```

Back to Table of Contents

Emulated memory (RAM) size

The next parameter defines the amount of host memory the chosen CHARON-AXP model reserves for the emulation:

```
#set ram size=4096
set ram size=32768
```

The amount of RAM is specified in MB. It cannot exceed or be lower than certain values specific for each HP Alpha model. It is very important to keep the listed predefined increment between possible memory values.

The following table shows all the parameters:

Hardware Model	RAM size (in MB)			
	Min	Max	Default	Increment
AlphaServer 400	64	1024	512	64
AlphaServer 800	256	8192	512	256
AlphaServer 1000	256	1024	512	256
AlphaServer 1000A	256	1024	512	256
AlphaServer 1200	256	32768	512	256
AlphaServer 2000	64	2048	512	64
AlphaServer 2100	64	2048	512	64
AlphaServer 4000	64	32768	512	64
AlphaServer 4100	64	32768	512	64
AlphaServer DS10	64	32768	512	64
AlphaServer DS15	64	32768	512	64
AlphaServer DS20	64	32768	512	64
AlphaServer DS25	64	32768	512	64
AlphaServer ES40	64	32768	512	64
AlphaServer ES45	64	32768	512	64
AlphaServer GS80	256	65536	512	256
AlphaServer GS160	512	131072	512	512
AlphaServer GS320	1024	262144	1024	1024

It is possible to leave the RAM line commented out. In this case the model's default RAM amount is used.

Note that in some particular orders your license may restrict the maximum RAM amount of each HP Alpha model.

Back to Table of Contents

Console

Mapping to system resources

The next step is the specification of the HP Alpha console (OPA0) serial line:

```
#load physical_serial_line OPA0 line="/dev/ttyN"
#load virtual_serial_line OPA0 port=10003
load operator_console OPA0
```

The goal of this configuration step is to tell CHARON-AXP what host device to use as the virtual system console. The following options are available:

<table>
<tr><th>Option</th><th>Description</th></tr>
<tr><td>physical_serial_line</td><td>Mapping to host serial line, both physical and virtual. Use the following mapping for different types of host serial lines:
<table>
<tr><th>Mapping</th><th>Description</th></tr>
<tr><td>/dev/tty&lt;N&gt;</td><td>Virtual serial lines</td></tr>
<tr><td>/dev/ttyS&lt;N&gt;</td><td>Onboard serial lines</td></tr>
<tr><td>/dev/ttyUSB&lt;N&gt;</td><td>Modem or usb serial lines adapters</td></tr>
</table>
A specific account for running CHARON ("charon") does not allow usage of physical consoles "<code>/dev/tty&lt;N&gt;</code>" as CHARON consoles. If you plan to map CHARON console to "<code>/dev/tty&lt;N&gt;</code>" use only "root" account for CHARON running.</td></tr>
<tr><td>virtual_serial_line</td><td>Mapping to an IP port of CHARON-VAX host.
Using this mapping it is possible to connect to CHARON-VAX console and disconnect from it at any time.</td></tr>
<tr><td>operator_console</td><td>Mapping to the current TTY console</td></tr>
</table>

The default setting is "operator_console".

The second console line "TTA0" can be also optionally configured (for 1 CPU models such as HP AlphaServer 400, HP AlphaServer 800, HP AlphaServer 1000, HP AlphaServer 1000A, HP AlphaServer DS10, HP AlphaServer DS10L and HP AlphaServer DS15):

```
load physical_serial_line TTA0 line="/dev/tty7"
set COM2 line=TTA0
```

⚠ Note there are a number of additional parameters for CHARON-AXP serial lines configuration. Follow this link for details.

Exit on pressing F6 button

Despite the fact that CHARON-AXP can exit with "power off" command given in its SRM console it is also recommended to set a hot key to stop the guest from the console in addition to the system tray icon:

```
set OPA0 stop_on = F6
```

This line provides CHARON-AXP the ability to exit by pressing the "**F6**" button.

Back to Table of Contents

Improve granularity of emulated timer

The next configuration option can be applied for improving granularity of emulated CHARON-AXP timer:

```
#set isa clock_period=1000
```

Do not uncomment this parameter unless there are some problems with system time or system clock intervals in guest OS.

Back to Table of Contents

ATAPI CD/DVD-ROM configuration

If the sample configuration file provides this parameter it is possible to map this particular CHARON-AXP emulator's "DQA0" CD-ROM to the host CD/DVD-ROM with the following setting:

```
set ide container="/dev/sg<N>"
```

How to find proper "/dev/sg" device is explained in this section.

Back to Table of Contents

Networking

CHARON-AXP supports DE435, DE450, DE500AA, DE500BA and DE602 virtual network adapters.

All of them are configured in a similar way:

```
load DE500BA/dec21x4x EWA interface=EWA0
load packet_port/chnetwrk EWA0 interface="eth0"
```

In the example above the first line loads DE500BA virtual adapter with a name "EWA"; the following line maps it to host network interface "eth0". Note that the mapping is performed in 2 steps:

1. A mapping object "packet_port" with a name "EWA0" is loaded and connected to host interface "eth0", so CHARON-AXP will use this interface for its networking
2. The loaded DE500BA virtual adapter "EWA" is connected to the "packet_port" object "EWA0"

It is possible to load several DE435, DE450, DE500AA, DE500BA or DE602 controllers, for example:

```
load DE500BA/dec21x4x EWA interface=EWA0
load packet_port/chnetwrk EWA0 interface="eth0"

load DE500BA/dec21x4x EWB interface=EWB0
load packet_port/chnetwrk EWB0 interface="eth1"
```

Some network adapters available in CHARON-AXP are preloaded (for example, HP AlphaServer DS15 contains 2 preloaded adapters EWA and EWB), so their configuration is even more simple:

```
load packet_port/chnetwrk EWA0 interface = "eth0"
```

> CHARON supports VLAN adapters. If used, proceed with their installation and configuration according to the network adapter vendor User's Guide and then use the resulting VLAN interface the same way as the regular network interface.

Follow this link for more details of CHARON-AXP network controllers configuration.

Back to Table of Contents

Disk/tape subsystem

The next step is configuration of the disk/tape subsystem and mapping it to system resources using the samples given in the template configuration files.

CHARON-AXP supports KZPBA and KGPSA-CA adapters.

KZPBA PCI SCSI disk/tape controller

Below is the typical configuration options for KZPBA PCI SCSI disk/tape controller:

```
load KZPBA PKA scsi_id = 7

# Disks
#set PKA container[0] = "<file-name>.vdisk"
#set PKA container[100] = "/dev/sd<L>"

# Unknown SCSI device
#set PKA container[200] = "/dev/sg<N>"

# CD-ROM
#set PKA container[300] = "/dev/cdrom"
#set PKA container[300] = "/dev/cdrom1"
#set PKA container[300] = "/dev/cdrom<N>"
#set PKA container[300] = "/dev/sr0"
#set PKA container[300] = "/dev/sr<N>"

# CD-ROM image
#set PKA container[400] = "<file-name>.iso"

# Tape
#set PKA container[500] = "/dev/sg<N>"
#set PKA container[600] = "<file-name>.vtape"
```

The first line ("load KZPBA PKA") loads disk controller KZPBA with name "PKA", followed by 8 group of lines showing different ways of mapping to the host resources:

Type of mapping	Description
"<file-name>.vdisk"	Mapping to files representing physical disks of the HP Alpha system (disk images). These files can be created from scratch with "mkdskcmd" utility. Data and OS disk backups are transferred from the original system via tapes or network and restored into these container files. Mapping may also include the full path, for example: "/my_disks/my_boot_disk.vdisk"
"/dev/sd<L>"	Mapping to physical disk. "L" is letter here. ⚠ Be careful not to destroy all the information from the disk dedicated to CHARON-AXP by mistake. These disks must not be formatted by the host OS. ℹ It is also possible to use not a whole disk, but previously created partitions on it. In this case the syntax is the following: "/dev/sd<L><N>" where N is the number of partition to be used.
"/dev/dm-<N>" "/dev/mapper/mpath<N>" "/dev/mapper/disk<N>"	Mapping to multipath disk. ⚠ Be careful not to destroy all the information from the disk dedicated to CHARON-AXP by mistake. These disks must not be formatted by the host OS.
"/dev/sg<N>"	Direct mapping to some SCSI device, for example, a SCSI disk or tape reader. How to find proper "/dev/sg" device is explined in this section.
"/dev/sr<N>" "/dev/cdrom" "/dev/cdrom<N>	Mapping to host CD-ROM device.
"<file-name>.iso"	Mapping to an ISO file for reading distribution CD-ROM image. Mapping may also include the full path (recommended), for example: "/my_disks/vms_distributive.iso"

"<file-name>.vtape"	Mapping to the file representing the tape (tape image). These files are created automatically. Mapping may also include a full path (recommended), for example: "/my_tapes/backup.vtape"

Additionally it is possible to specify a parameter "media_type" to assign the type of the attached media explicitly.

Example:

```
set PKA media_type[600]="RX23"
```

Numbers in the square brackets represent SCSI addresses and LUNs associated with each container of the KZBPA controller. They have the following structure:

[XXYY], where

Parameter	Range	Description
XX	0...15	Stands for SCSI ID of each connected unit. ℹ Note that KZPBA itself has some ID associated with it. By default it is 7, but it can be changed in the following way: `load KZPBA PKA scsi_id = 0` In this example an instance "PKA" of KZPBA controller is assigned with SCSI ID 0.
YY	00...07	Stands for LUN.

It is possible to load several KZPBA controllers: DKB, DKC, etc. by configuring specific placement for them on the PCI bus. It is discussed in details in the "Configuration details" chapter of this Guide.

Some HP Alpha systems emulated by CHARON-AXP have already had one or two KZPBA controllers preloaded. If the system has only one preloaded controller, the template configuration file usually provides some sample line on how to add another one, for example:

```
load KZPBA PKA bus=pci_1 device=1 function=0 irq_bus=isa irq=24
```

Follow this link for details of KZPBA controllers configuration.

Back to Table of Contents

KGPSA-CA PCI FC disk controller

Optionally it is possible to configure KGPSA-CA FC disk controller.

It can be configured in 2 modes:

- Direct mapping to the host resources
- Pass Through mode

Below is an example of KGPSA-CA controller loading:

```
load KGPSA FGA
```

Optionally another KGPSA-CA adapter can be loaded similar way:

```
load KGPSA FGB
```

Follow this link for details of KGPSA-CA controllers configuration.

Back to Table of Contents

KGPSA-CA mapping to the host resources

Below is the typical configuration options for KGPSA-CA PCI FC disk controller, mapped to the host resources ("L" is letter here):

```
load KGPSA FGA
#set FGA container[0] = "<file-name>.vdisk"
#set FGA container[100]="/dev/sd<L>"
```

The first line ("load KGPSA FGA") loads disk controller KGPSA with name "FGA", followed by 2 groups of lines showing different ways of mapping to the host resources:

Type of mapping	Description
"<file-name>.vdisk"	Mapping to the file representing a physical disk of the HP Alpha system (disk image). These files can be created from scratch with "mkdskcmd" utility. Data and OS disk backups are transferred from the original system via tapes or network and restored into these container files. Mapping may also include the full path (recommended), for example: "/my_disks/my_boot_disk.vdisk"
"/dev/sd<L>"	Mapping to physical disk. "L" is letter here ⚠ Be careful not to destroy all the information from the disk dedicated to CHARON-AXP by mistake. These disks must not be formatted by the host OS. ℹ It is also possible to use not a whole disk, but previously created partitions on it. In this case the syntax is the following: "/dev/sd<L><N>" where N is the number of partition to be used.
"/dev/dm-<N>" "/dev/mapper/mpath<N>" "/dev/mapper/disk<N>"	Mapping to multipath disk. ⚠ Be careful not to destroy all the information from the disk dedicated to CHARON-AXP by mistake. These disks must not be formatted by the host OS.

Numbers in the square brackets represent KGPSA-CA units. They can be in the range 0..32766, but no more than 255 units can be configured on a single controller.

Back to Table of Contents

KGPSA-CA pass through mode

It is also possible to use the emulated KGPSA-CA in "pass through" mode to address a physical EMULEX LightPulse PCI/PCI-X/PCIe FC adapter plugged into the host’s PCI/PCI-X/PCIe slot.

The sample configuration file provides a template for this type of mapping:

```
#set FGA host_bus_location = "/dev/kgpsaX"
#set FGB host_bus_location = "/dev/kgpsaY"
```

Follow this link for detailed description of building and installation of EMULEX LightPulse PCI/PCI-X/PCIe FC adapter driver.

Back to Table of Contents

Auto boot

CHARON-AXP can be configured to automatically boot an operating system at start up by specifying the default boot device and setting the 'auto_action' parameter to 'restart' from the console.

Example: dka0 is defined as the default boot device

```
>>>set bootdef_dev dka0
>>>set auto_action restart
```

Back to Table of Contents

Migration to CHARON-AXP for Linux

Table of Contents

Introduction

This section describes how to migrate your HP Alpha system to CHARON-AXP. We will use a sample HP AlphaServer ES40 system running OpenVMS to demonstrate the migration procedure. The process is similar for all CHARON-AXP models.

Back to Table of Contents

Collecting information about the source HP Alpha system

The first step is to determine the exact configuration of your HP Alpha hardware in order to create the CHARON-AXP configuration file.

Turn on your source HP Alpha system. At the ">>>" prompt, issue "show device" command:

```
>>>show device

sys0.0.0.0.0 SYS0 System ROOT Device
ewa0.0.0.1.1 EWA0 F8-D1-11-00-67-E6
pka0.0.0.2.1 PKA0 Q-Logic/ISP PCI SCSI HBA
pga0.0.0.3.1 PGA0 WWN 1000-0000-0248-C550
pqa0.0.0.15.0 PQA0 ALi 1553C Integrated IDE Controller
pqb0.0.1.15.0 PQB0 ALi 1553C Integrated IDE Controller
dqa0.0.0.15.0 DQA0 TSSTcorpCDDVDW SH-222BB
dka0.0.0.2.1 DKA0 DEC RZ28 (C)DEC
dka100.1.0.2.1 DKA100 DEC RZ22 (C)DEC
dka200.2.0.2.1 DKA200 DEC RZ23 (C)DEC
mka600.6.0.2.1 MKA600 Virtual SCSI Tape

>>>
```

To get more detailed information, boot OpenVMS and issue a "show device /full" command:

```
$ show device /full

Disk PFCAXP$DKA0:, device type RZ28, is online, mounted, file-oriented device,
shareable, available to cluster, error logging is enabled.
...

Disk PFCAXP$DKA100:, device type RZ22, is online, file-oriented device,
shareable, available to cluster, error logging is enabled.
...

Disk PFCAXP$DKA200:, device type RZ23, is online, file-oriented device,
shareable, available to cluster, error logging is enabled.
...

Disk PFCAXP$DQA0:, device type TSSTcorpCDDVDW SH-222BB, is online,
file-oriented
device, shareable, available to cluster, error logging is enabled.
...

Disk $1$DGA0: (PFCAXP), device type RZ24, is online, file-oriented device,
shareable, available to cluster, error logging is enabled.
...

Magtape PFCAXP$MKA600:, device type Virtual SCSI Tape, is online, file-oriented
device, available to cluster, error logging is enabled, device supports
fastskip (per_io).
...

Terminal OPA0:, device type VT102, is online, record-oriented device, carriage
control.
...

Device EWA0:, device type DE500, is online, network device, device is a
template
only.
...

Device FGA0:, device type KGPSA Fibre Channel, is online, shareable, error
logging is enabled.
...

Device PGA0:, device type SCSI FCP, is online, error logging is enabled.

...

Device PKA0:, device type Qlogic ISP1020 SCSI port, is online, error logging is
enabled.
...

Device $1$GGA32767:, device type Generic SCSI device, is online, shareable.

$
```

In case of Tru64 UNIX V5 running on the host system it is recommended to use the following commands to get information on the host configuration:

Command	Description
`#/sbin/hwmgr view devices`	Get detailed information about the host hardware configuration
`#/sbin/hwmgr show scsi`	Get specific information about the host SCSI controllers and attached disks
`#/sbin/hwmgr view hierarchy`	Get information about the host controllers

Please reference to the Tru64 UNIX User's Guide for more details.

The source HP Alpha peripheral configuration in this example is:

Controller	Devices on controller	Description
KZPBA	-DKA0 (RZ28) -DKA100 (RZ22) -DKA200 (RZ23) -MKA600 (tape)	SCSI disk/tape controller
KGPSA-CA	-DGA0 (RZ24)	FC disk controller
OPA0		System console
TSSTcorpCDDVDW SH-222BB	-DQA0	IDE CD-ROM controller
EWA0		Network interface, MAC address: "F8-D1-11-00-67-E6"

Now collect some general information about the HP AlphaServer ES40 system:

```
>>>show cpu /full

System: PFCAXP, AlphaServer ES40 6/667

SMP execlet = 3 : Enabled : Streamlined.
Config tree = None
Primary CPU = 0
HWRPB CPUs = 4
Page Size = 8192
Revision Code =
Serial Number = SN01234567
Default CPU Capabilities:
System: QUORUM RUN
Default Process Capabilities:
System: QUORUM RUN

....

>>>
```

```
>>>show mem

System Memory Resources on 5-FEB-2015 09:29:16.42

Physical Memory Usage (pages): Total Free In Use Modified
Main Memory (512.00MB) 65536 56496 8610 430

...

>>>
```

So the collected information about the HP AlphaServer ES40 system is:

Component	Value
System Type	AlphaServer ES40 6/667
Serial Number	SN01234567
Number of CPUs	4
System memory	512 Mb

In some particular situations it is also important to know the exact placement of all the peripheral devices on HP Alpha PCI bus. To get that information issue a "show config" command at ">>>" prompt of HP Alpha console, for example:

```
>>>show config

...

PCI Bus

Bus 00 Slot 03: DECchip 21142 Network Controller
ewa0.0.0.3.0 00-00-F8-03-9A-6D

Bus 00 Slot 07: Cypress PCI Peripheral Controller

Bus 00 Slot 07: Function 1: PCI IDE

Bus 00 Slot 07: Function 2: PCI IDE

Bus 00 Slot 07: PCI USB

Bus 00 Slot 08: DECchip 21052 PCI to PCI Bridge

Bus 01 Slot 08: ISP1040 Scsi Controller
pka0.7.0.1008.0 SCSI Bus ID 7
dka0.0.0.1008.0 RZ2DD-KS
dka400.4.0.1008.0 RRD45

>>>
```

The "show config" command collects the following information of placement of peripheral devices on PCI bus:

- Bus number
- Slot number
- Function number

To find out the exact types of controllers and other useful information refer to the source HP Alpha system documentation.

Back to Table of Contents

Creation of CHARON-AXP configuration file

Using the above info, the following configuration can be created:

```
#
# HP AlphaServer model: AlphaServer ES40 6/667
#

set session hw_model = AlphaServer_ES40
set ace cpu_architecture = EV67
set rom dsrdb[0] = 1820 system_name = "AlphaServer ES40 6/667"

...

#
# Override default System Serial Number, set it to "SN01234567"
#

set rom system_serial_number = SN01234567

#
# Specify RAM size: 512 Mb
#

set ram size=512

#
# Map OPA0 console to the xtem from which CHARON-AXP runs
#

load operator_console OPA0

#
# Connect the emulator's DQA0 to the host's ATAPI CD/DVD-ROM drive.
#

set ide container="/dev/cdrom"

#
# Load optional DE500BA PCI Ethernet Adapter (EWA0) and map it to the "eth1" host network interface
#

load DE500BA/dec21x4x EWA interface=EWA0
load packet_port/chnetwrk EWA0 interface="eth1"

#
# Load DEC-KZPBA SCSI controller and map it to 3 disk containers and 1 tape container
#

load KZPBA PKA scsi_id = 7

set PKA container[0] = "/my_disks/bootable.vdisk"
set PKA container[100] = "/my_disks/RZ22.vdisk"
set PKA container[200] = "/my_disks/RZ23.vdisk"

set PKA container[600] = "/my_tapes/my_tape.vtape"

#
# Load DEC-KGPSA-CA PCI FC adapter and map it to a disk container
#

load KGPSA FGA

set FGA container[0] = "/my_disks/RZ24.vdisk"

...
```

Back to Table of Contents

Making disk images

In our example, possible mappings of KZPBA SCSI controller include disk and tape images. Tape images have not to be manually created whereas you have to provision disk images, as described below.

Our example creates disk images of the original physical type. In reality, this step is the best opportunity in the migration to provision bigger disks to get extra storage space.

Create special directories for storing disk and tape images, as needed. Created directories are referenced in the sample configuration file above.

```
$ mkdir /my_disks
$ mkdir /my_tapes
```

Next, create disk images using the "mkdskcmd" utility:

```
$ mkdskcmd -d rz24 -o /my_disks/rz24.vdisk
Please wait...
100% done
Success.
$ mkdskcmd -d rz23 -o /my_disks/rz23.vdisk
Please wait...
100% done
Success.
$ mkdskcmd -d rz22 -o /my_disks/rz22.vdisk
Please wait...
100% done
Success.
$ mkdskcmd -d rz28 -o /my_disks/bootable.vdisk
Please wait...100% done
Success.
```

Back to Table of Contents

Installation of HP Alpha operating system

The next step is to transfer the data from the source HP Alpha system to CHARON-AXP. The easiest way to do this is via backup over the network. But for this operation we need a bootable, network-enabled operating system on a CHARON-AXP disk image or physical disk.

The example configures the CHARON-AXP AlphaServer ES40 system for installation of HP OpenVMS from a distribution CD-ROM (usually it is "/dev/cdrom" if the host has only one CD-ROM drive):

```
#
# DEC-KZPBA SCSI controller is mapped to 5 disk containers; one of them (DKA300) - for migration
purposes;
# another one (DKA400) - for installation of fresh HP OpenVMS system from distributive
#

load KZPBA PKA scsi_id = 7

set PKA container[0] = "C:\my_disks\bootable.vdisk"
set PKA container[100] = "C:\my_disks\RZ22.vdisk"
set PKA container[200] = "C:\my_disks\RZ23.vdisk"
set PKA container[300] = "C:\my_disks\migration.vdisk"
set PKA container[400] = "C:\my_disks\fresh_openvms.vdisk"

#
# CD-ROM for HP OpenVMS installation (DQA0)
#

set ide container="/dev/cdrom"
```

⚠ DKA300 will be the disk where all the source disks will be copied, so its size needs to be big enough to store all the disk backup images

Create an empty disk image for installation of HP OpenVMS and another one for storing backups from the source HP Alpha system as it is shown in the section above.

Run CHARON-AXP and boot from the CDROM named "dqa0" ("migration.cfg" is the configuration file we use in this example):

```
$ es40.exe migration.cfg

initializing ...

polling for units on kzpba0, slot 2, bus 0, hose 1 ...
pka0.0.0.2.1 PKA0 Q-Logic/ISP PCI SCSI HBA

... enter console

CHARON-AXP/ES40 for Linux (AlphaServer ES40 6/667), Version 4.6.16803
(C) 2009-2015 STROMASYS SA.
All rights reserved.

P00>>>boot dqa0
```

Install HP Alpha/VMS including DECnet on "dka400". DECnet address must belong to the same area as the source HP Alpha system.

Login to the newly installed OpenVMS system and initialize the disk intended for backups storage. Let's assume it's prompt is "`newvms$` "

```
newvms$ INIT DKA300: SCRATCH
newvms$ MOUNT/SYSTEM/NOASSIST DKA300: SCRATCH
```

Back to Table of Contents

Making remote backups

Now we are ready to create disk backups from the source HP Alpha system to CHARON-AXP.

Boot CHARON-AXP and make sure that the source HP Alpha system is available via DECnet.

Login to the source HP Alpha system. Stop all the batch queues, kick off the users, stop all applications and close databases if any. The commands listed in SYS$MANAGER:SYSHUTDWN.COM may be helpful. The goal is to close as many files as possible. The system disk will have several files opened (pagefile, swapfile, etc.), but this is normal.

ℹ The use of the "SHOW DEVICE /FILES" command would be of help to know files opened on a disk

Let's assume the CHARON-AXP system is node 1.400 in this example. Issue then the following commands from the source HP Alpha whose prompt is set to "`source$` "

```
source$ BACKUP/IMAGE/IGNORE=INTERLOCK DKA0: 1.400::DKA300:[000000]DKA0.BCK/SAVE
source$ BACKUP/IMAGE/IGNORE=INTERLOCK DKA100: 1.400::DKA300:[000000]DKA100.BCK/SAVE
source$ BACKUP/IMAGE/IGNORE=INTERLOCK DKA200: 1.400::DKA300:[000000]DKA200.BCK/SAVE
```

Once the backup procedure completes, the disk "DKA300" of CHARON-AXP will contain 3 savesets: "DKA0.BCK", "DKA100.BCK" and "DKA200.BCK"

Back to Table of Contents

Restore backups to CHARON-AXP disks

Next, restore the new savesets to their corresponding virtual disks. Login to CHARON-AXP and issue this sequence of commands to restore all the savesets created on the previous step:

```
newvms$ MOUNT/FOREIGN DKA0:
newvms$ BACKUP/IMAGE DKA300:[000000]DKA0.BCK/SAVE DKA0:
newvms$ DISMOUNT DKA0:
newvms$ MOUNT/FOREIGN DKA100:
newvms$ BACKUP/IMAGE DKA300:[000000]DKA100.BCK/SAVE DKA100:
newvms$ DISMOUNT DKA100:
newvms$ MOUNT/FOREIGN DKA200:
newvms$ BACKUP/IMAGE DKA300:[000000]DKA200.BCK/SAVE DKA200:
newvms$ DISMOUNT DKA200:
```

If you are going to have CHARON-AXP and the original physical HP Alpha on the network at the same time, you must change the network identity of one (usually the CHARON-AXP).

The easiest way is to boot the CHARON-AXP virtualized system on the restored system disk with the network Disabled and configure new addresses, as needed.

ℹ The NIC can be disabled with a "disabled" statement in the CHARON configuration file.

Then Enable the network and reboot.

Back to Table of Contents

Alternative ways of data transfer

Some alternative methods of data transfer are also possible. For example:

- Connect a SCSI tape drive to CHARON-AXP host via a PCI card
    - Map the tape drive in the CHARON-AXP configuration file
        a. Restore source HP Alpha system backups from tape to disk images via OpenVMS running on CHARON-AXP.
        b. Boot from standalone backups and restore its content to CHARON-AXP virtual disks
    - Dump source HP Alpha system backups to tape images with "mtd" utility and:
        a. Boot from freshly installed OpenVMS system and restore the tape images to CHARON-AXP virtual disks
        b. Boot from standalone backups and restore its content to CHARON-AXP virtual disks
- Create a network cluster between the source HP Alpha system and CHARON-AXP (it is possible to use the source system as boot server); then simple backup from one disk to another:

    ```
    $ BACKUP/IMAGE/IGNORE=INTERLOCK REAL$DKA0: DKA0:
    ```

Back to Table of Contents

CHARON-AXP for Linux virtual network

Table of Contents

General description

It is strongly recommended to use only physical network adapters for CHARON-AXP networking to gain maximum performance. In situations where the host has only one network adapter, you can use the LINUX virtual network Interfaces ("TUN/TAP") and map individual CHARON-AXP instances to their own virtual interfaces.

There are 2 ways to create the Linux virtual network Interfaces ("TUN/TAP"):

- Using "ncu" utility
- Manually

Back to Table of Contents

Using "ncu" utility to establish CHARON virtual network

Login as root. Start "ncu" utility:

```
# ncu

CHARON Network Configuration Utility, STROMASYS (c) 2015 Version
1.4

Interfaces Dedicated to State
---------- ------------ ------------
eth0       host         connected to host
eth1       host         connected to host
lo         host         unmanaged from host

==================================================================
bridge name     bridge id        STP enabled            interfaces

select action:
1 - Dedicate to CHARON
2 - Release to host
3 - Create Bridge with TAPs
4 - Destroy Bridge
5 - Print status
6 - Exit
:> 3
```

Enter "3" to create a bridge between the host physical network adapter and the Linux virtual network interfaces (TAP) and specify the physical network interface ("eth1" in our example) and the number of virtual network interfaces to be created (2 in our example):

```
Specify the interface to be used for BRIDGE:eth1
How many tap should be created:2
Forming the bridge: ..1..2..3..4..5.. addif tap0 .. addif tap1 ..7..8 done!
Formed bridge br0_eth1 attached over eth1...

select action:
1 - Dedicate to CHARON
2 - Release to host
3 - Create Bridge with TAPs
4 - Remove Bridge
5 - Print status
6 - Exit
:> 5
```

Now enter "5" to see the created virtual interfaces:

```
Interfaces   Dedicated to    State
----------   ------------    ------------
eth0         host            connected to host
eth1         bridge          connected to bridge
lo           host            unmanaged from host
tap0         bridge          connected to bridge
tap1         bridge          connected to bridge

==================================================================
bridge name     bridge id           STP enabled        interfaces
br0_eth1        8000.525400698995a  no                 tap0
tap1

select action:
1 - Dedicate to CHARON
2 - Release to host
3 - Create Bridge with TAPs
4 - Remove Bridge
5 - Print status
6 - Exit
:> E
```

In the example above we see 2 virtual network interfaces "tap0" and "tap1" connected to the created bridge. The physical network interface "eth1" is used for the bridge to the virtual network interfaces.

The interfaces "tap0" and "tap1" are ready to be used in CHARON configurations - they do not need to be additionally dedicated to CHARON.

Enter "6" to quit "ncu" utility.

Back to Table of Contents

Manual configuration of CHARON virtual network

Host preparation

1. Login as "root" user.

2. Configure the physical network interface to run in promiscuous mode using the following command. This interface will be dedicated to the whole network bridge (created later).

   ```
   # ifconfig eth<N> 0.0.0.0 promisc up
   ```

   Promiscuous mode allows the physical (or virtual) network interface to accept the entire volume of incoming packets. This mode is essential for consistency of the information transfer.

3. In case the firewall is enabled on the host system, the following command should be executed to allow the bridge to forward IP packets:

   ```
   # /sbin/iptable -I FORWARD -m physdev --physdev-is-bridged -j ACCEPT
   ```

   This command can also be performed from the bridge configuration script. It has to be executed each time the *iptables* service is (re)started.

   It is also possible to make this setting system-wide. Either:
   a. Issue the given command from the firewall control panel.
   b. Add the following line to the end of the "/etc/sysconfig/iptables" file:

      ```
      -I FORWARD -m physdev --physdev-is-bridged -j ACCEPT
      ```

Back to Table of Contents

Virtual interface creation

Creation of the desired number of virtual network interfaces (TAPs) can be performed in the following way:

```
# tunctl [-t tap<N>]
```

where “tap<N>” is a name of an instance of the virtual network interface, i.e. “tap0”, “tap1” etc.

Once each virtual network interface instance is created it must be set to promiscuous mode:

```
# /sbin/ifconfig tap<N> promisc up
```

Back to Table of Contents

Bridge creation

To interconnect the physical and virtual network interfaces created in the previous step, the network bridge must be introduced in the following way:

```
# /usr/sbin/brctl addbr br0
```

where “*br0*” stands for a name of the created bridge.

Now it is possible to add the network interfaces to the created bridge:

```
# /usr/sbin/brctl addif br0 eth<N>
# /usr/sbin/brctl addif br0 tap0
…
# /usr/sbin/brctl addif br0 tap<N>
```

Example:

```
# /usr/sbin/brctl addif br0 eth1

# /usr/sbin/brctl addif br0 tap0
```

The proposed configuration assumes one and only one network bridge, so loops are not possible. It is required to turn off the spanning tree protocol with the following command:

```
# /usr/sbin/brctl stp br0 off
```

Back to Table of Contents

Starting bridge

To start the created bridge "*br0*" use the following command:

```
# /sbin/ifconfig br0 up
```

Back to Table of Contents

Usage of the virtual interface in CHARON-AXP configuration

Once the "*tap<N>*" interfaces have been created, the load command maps those interfaces to CHARON-AXP:

```
...
load tap_port/chnetwrk XQA0 interface="tap<N>"
...
```

Back to Table of Contents

CHARON-AXP for Linux licensing

Table of Contents

General description

CHARON-AXP products are protected by licenses, issued by STROMASYS for each customer individually. The CHARON-AXP license defines all the specifics of the particular CHARON-AXP distribution and its usage.

The license is implemented in the form of a hardware dongle (a Sentinel HASP key) or a software license. Please be careful with your license key. In case of loss or damage, CHARON-AXP will not run or start unless the license key is replaced. For extra protection, STROMASYS recommends the use of a backup license key (purchased separately) that can replace the main license key for a restricted period of time. It is possible to specify the backup license in the CHARON-AXP configuration file to prevent CHARON-AXP from stopping in case its main license is no longer accessible.

The CHARON-AXP license is read upon the start of each instance of CHARON-AXP and at a specified interval (defined by the license content) during the emulated system execution. If CHARON-AXP detects the absence (or malfunction) of the license key / software license, CHARON will try to use a backup license (if specified in the configuration file). If the license is not available / not specified, CHARON displays a warning message in the log file requesting license key reconnection or software license reactivation. If the license is not reconnected within a given period of time (the check interval), CHARON-AXP exits.

Note that if the time-restricted license is used and it expires, CHARON-AXP tries to find its replacement automatically and, if found, CHARON-AXP proceeds using the replacement license.

> The present CHARON-AXP implementation requires that the expired license be removed to allow the running CHARON-AXP to switch to some other (valid) one.
>
> The CHARON-AXP software license is not distributed in case of Proof-of-Concept and evaluation installations. Only hardware dongles are used in this case.
>
> It is important to connect HASP license keys to the computer from time to time even if CHARON-AXP is not used. The keys contain a built-in accumulator that needs to be charged. If the accumulator is completely discharged, a license key can be fatally damaged.

Update of the CHARON-AXP license can be performed on the fly without stopping CHARON-AXP. At the next license check, CHARON-AXP will use the updated license normally.

The following sections list all the main parameters of the CHARON-AXP licensing mechanism.

Back to Table of Contents

Parameters defined by CHARON-AXP license

The following table represents all the parameters defined by CHARON-AXP license:

General	Products relevant	Optional
• Physical key ID • License Number • End user name • Master key ID • License release date and time • Update Number • Purchasing Company name. In most cases the company to which the key was issued originally	• Commercial product name • Commercial product code • Commercial product version and range of build numbers suitable for running • Range of CHARON-AXP virtual models available for running • Type of host CPU required • Host operating system required • Number of virtual CPUs enabled for virtual SMP systems • Minimum number of host CPU cores required • Minimum host memory required • Maximum memory emulated. If not present the value defaults to the maximum memory possible for the particular virtual system. Note that the maximum memory may not be available to the virtual system if the host computer has insufficient physical memory. • Maximum number of CHARON-AXP instances that can be run concurrently • Whether or not CHAPI (CHARON-AXP API) can be used with this product • Product and Field Test expiration dates (if any) • Product and Field Test executions counter (if any) • Maximum number of hosts that may run CHARON-AXP concurrently (in the case of a networking license) • Level of support (if any), end date of any support contract, the "First Line" Service Provider • Frequency of CHARON-AXP license checking during CHARON-AXP execution	• Parameter that reduces the maximum speed of the program • Parameter that prohibits use of Advanced CPU Emulation. If not present the Advanced CPU Emulation is enabled

Back to Table of Contents

CHARON-AXP licensing models

CHARON-AXP licensing models are divided in 3 groups:

Regular Sentinel HASP keys

This is most common way of CHARON-AXP licensing.

The CHARON-AXP license is embedded in a Sentinel HASP dongle. This license is available only on the host where the dongle is physically installed.

The CHARON-AXP installation procedure takes care of the Sentinel HASP run-time (driver) installation. Once the CHARON-AXP product has been installed, it is possible to plug-in the regular license key and proceed with CHARON-AXP usage without additional configuration steps.

> The number of CHARON-AXP instances allowed to run on a particular host may be restricted by the license content (see above).

Network Sentinel HASP keys

The Network Sentinel HASP key (red dongle) can be shared between several hosts running CHARON-AXP (including the host on which the network license is installed).

If CHARON-AXP is installed on the host where the network key is connected, no additional steps are required. The Sentinel driver is activated as part of the CHARON-AXP installation. If the host does not have CHARON-AXP installed, the host can still distribute the connected network license to CHARON-AXP instances running on other hosts. In this case the Sentinel driver must be installed on the host manually.

The Sentinel run-time driver is distributed as a separate RPM package in the CHARON-AXP kit. Please see the "License installation" section of this chapter for details.

Once the Sentinel run-time driver is installed and the network license is connected, CHARON-AXP can be started on any appropriate host on the LAN network segment.

> The Network license key contains a specific parameter to restrict the number of hosts allowed to run CHARON-AXP at the same time. Together with a parameter defining the number of CHARON-AXP instances that may run at the same time, the network license sets the total number of running CHARON-AXP instances on the allowed number of hosts.

Software licenses

The CHARON-AXP Software License is a "virtual" key with exactly the same functionality as the hardware dongle.

The CHARON-AXP software license does not require any hardware but it requires installation of the Sentinel run-time environment.

Software licenses are always network-wide on Linux, so they behave the same way as Network HASP keys.

> Software Licenses are highly dependent on hardware configuration of CHARON host. Do not change hardware configuration since it leads to disabling of installed Software License!
>
> If CHARON host has to be upgraded use the following procedure:
>
> 1. Transfer Software License to some other host.
> 2. Upgrade CHARON host.
> 3. Transfer Software License back to CHARON host.

Back to Table of Contents

Multiple licenses configuration and backup license

For any type of licensing, CHARON-AXP can use **only one valid ("active") license (of given vendor code) at a time**.

The "hasp_srm_view" utility displays the "active" license only. The utility provides the license number and ID / IP address of the host where the active license is installed.

> CHARON-AXP **cannot:** check all the available license keys / software licenses, choose one, automatically switch from one key to another, etc.

The general recommendation is to avoid usage of multiple keys in one network segment. Use only one locally installed license per host or one network license per local network segment containing several CHARON-AXP hosts.

When needed, it is possible to use a special parameter in the CHARON-AXP configuration file to specify exactly which license must be used by each particular instance of CHARON-AXP:

Parameter	Type	Value

license_key_id[N], N=0 or 1	Numeric	A number (decimal Sentinel key ID) that specifies regular (N=0) and backup (N=1) license keys to be used by CHARON-AXP.
		Example:
		`set session license_key_id[0]=1877752571`
		`set session license_key_id[1]=354850588`
		It is also possible to specify both regular and backup key in one line.
		Example:
		`set session license_key_id[0]=1877752571 license_key_id[1]=354850588`
		Depending on the presence of the regular and/or backup license key IDs in the configuration file, CHARON-AXP behaves differently:
		1. **No keys are specified** CHARON-AXP behaves as usual (performs unqualified search for any suitable key). If no keys are found, CHARON-AXP exits.
		2. **Both keys are specified** CHARON-AXP performs qualified search for regular license key. If it is not found, CHARON-AXP performs qualified search for backup license key. If it is not found, CHARON-AXP exits.
		3. **Only regular key is specified** CHARON-AXP performs qualified search for regular license key. If it is not found, CHARON-AXP performs unqualified search for any suitable key. If it is not found, CHARON-AXP exits.
		4. **Only backup key is specified** CHARON-AXP behaves as usual (performs unqualified search for any suitable key). If no keys are found, CHARON-AXP exits.

Back to Table of Contents

License installation

Installation of Regular and Network license keys

Installation of CHARON-AXP regular and network licenses consists of:

1. Installation of the Sentinel run-time environment on the CHARON-AXP host (regular and network keys) or on the host that will distribute CHARON-AXP licenses over a local network segment (network key only). The Sentinel software (the “aksusbd” RPM package) is installed automatically by CHARON-AXP for Linux.
2. Physical connection of the HASP license dongle to the CHARON-AXP host or to the host distributing the CHARON-AXP license over the local network segment.

When manual installation of Sentinel run-time is required (in the case of the network license server that does not have CHARON-AXP installed), open the CHARON-AXP kit folder and proceed the following way:

```
# rpm --nodeps -ihv aksusbd-2.4-1.i386.rpm charon-hasp-4.6-16803.68704.el65.x86_64.rpm
```

In case of network-wide license (red dongle) do the following:

- *On server side (where network license will reside)*: open port 1947 for both TCP and UDP
- *On clients side*: open UDP ports 30000-65535
- *Both on server and client sides*: setup default gateway

Please consult with your Linux User's Guide on details.

If stricter firewall rules are required, it is possible to open the ports 30000-65535 and 1947 only for the "/usr/sbin/hasplmd" daemon.

Some additional packages may be needed in certain cases, for example "glibc.i686"

Back to Table of Contents

Replacement of currently installed Sentinel run-time

Replacement of currently installed Sentinel Run-time may be needed in case of:

- Upgrade to a newer version of CHARON-AXP
- Installation of a specific CHARON-AXP license Run-time provided by STROMASYS

Run-time replacement is a two step process:

- Remove the current run-time (and the package "charon-hasp-<...>.rpm" containing the run-time customization) with the command

```
# rpm --nodeps -e aksusbd charon-hasp-<...>
```

- Change to the directory where the new run-time RPM resides (along with the corresponding "charon-hasp-<...>.rpm" customization package) and issue the command:

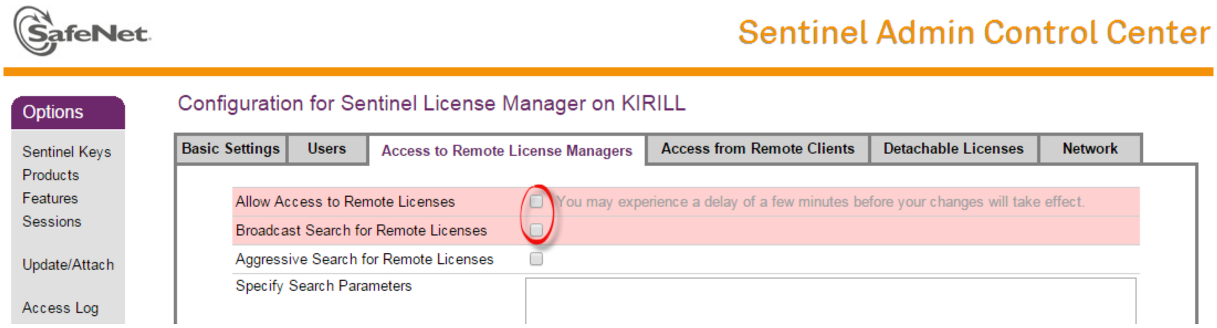
```
# rpm --nodeps -ihv aksusbd<…>.rpm charon-hasp-<...>.rpm
```

Back to Table of Contents

Installation and update of CHARON-AXP Software License or HL/HASP dongle License

CHARON-AXP software licenses can be installed / updated according to the procedure described below. This procedure is also applicable for update of a license in case of HL/HASP dongles.

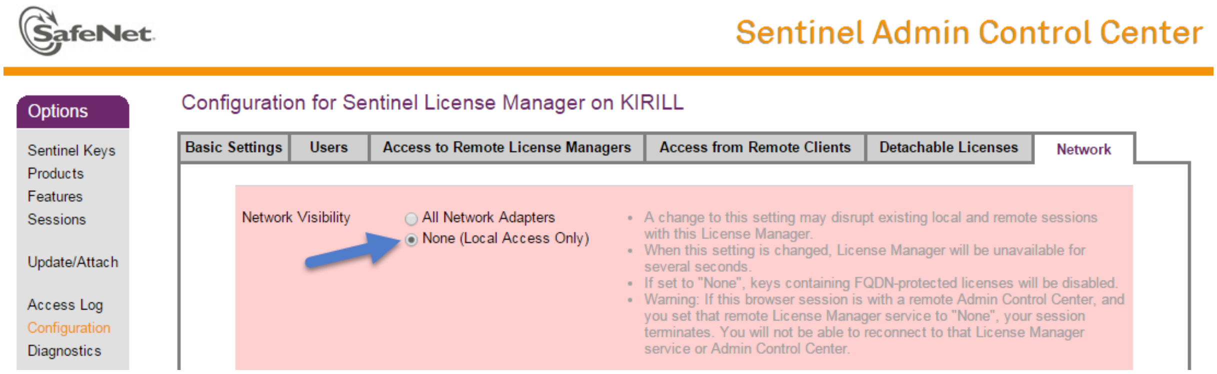
- Install CHARON-AXP together with Sentinel run-time (Sentinel run-time is an essential part of CHARON-AXP for Linux distribution)
- Reboot host system
- In case of Software License installation and if there are already installed network-wide SL's in local network disable access to network licenses in the following way:
  - Go to http://localhost:1947 to access the "Sentinel HASP Admin Control Center" (ACC).
  - Select "Configuration" option at the left panel, then "Access to Remote License Managers" tab.
  - Uncheck the highlighted options:



Configuration for Sentinel License Manager on KIRILL

Allow Access to Remote Licenses □ You may experience a delay of a few minutes before your changes will take effect.
Broadcast Search for Remote Licenses □
Aggressive Search for Remote Licenses □
Specify Search Parameters

  - Press "Submit" button to apply settings
  - Select "Network" tab.
  - Switch "Network visibility" to "None":



Configuration for Sentinel License Manager on KIRILL

Network Visibility ○ All Network Adapters
● None (Local Access Only)

- A change to this setting may disrupt existing local and remote sessions with this License Manager.
- When this setting is changed, License Manager will be unavailable for several seconds.
- If set to "None", keys containing FQDN-protected licenses will be disabled.
- Warning: If this browser session is with a remote Admin Control Center, and you set that remote License Manager service to "None", your session terminates. You will not be able to reconnect to that License Manager service or Admin Control Center.

  - Press "Submit" button to apply setting.

    - Do not forget to return these settings back after SL installation.
- Connect HASP dongle to host system (in case of update of a license in a dongle)
- Collect CHARON-AXP host fingerprint file (".c2v") - in case of first installation of Software License:

  ```
  # hasp_srm_view -fgp my_host.c2v
  ```

  or collect ".c2v" file in case if already installed Software License or connected HL/HASP dongle needs updating:

  ```
  # hasp_srm_view -c2v current_license.c2v
  ```

- Send the ".c2v" file ("*my_host.c2v*" / "current_license.*c2v*" in the examples above) to STROMASYS
- Receive a ".v2c" file in return and put it somewhere on the CHARON-AXP host.
- Start any web browser on this system and go to http://localhost:1947 to access the "Sentinel HASP Admin Control Center" (ACC) or configure ACC for remote access (see the details below).
- In ACC, under the Options menu, select Update/Attach, "Browse" for the "*.v2c" file and then "Apply File".
- Ensure that the license appears in the “Sentinel Keys” menu.

Alternatively it is also possible to use "hasp_update" command line utility for applying the ".v2c" file.

Content of the installed software license is not shown by the Sentinel HASP Admin Control Center. To see it please run "hasp_srm_view" utility from local console or configure remote access according to the instructions given in the "hasp_srm_view" utility section

In case of network-wide software license do the following:

- *On server side (where network license will reside)*: open port 1947 for both TCP and UDP
- *On clients side*: open UDP ports 30000-65535
- *Both on server and client sides*: setup default gateway

Please consult with your Linux User's Guide on details.

If stricter firewall rules are required, it is possible to open the ports 30000-65535 and 1947 only for the "/usr/sbin/hasplmd" daemon.

Back to Table of Contents

License management

CHARON-AXP license management is performed by the Sentinel Admin Control Center and specific utilities. These are described in the sub-sections below.

Sentinel Admin Control Center

General Description

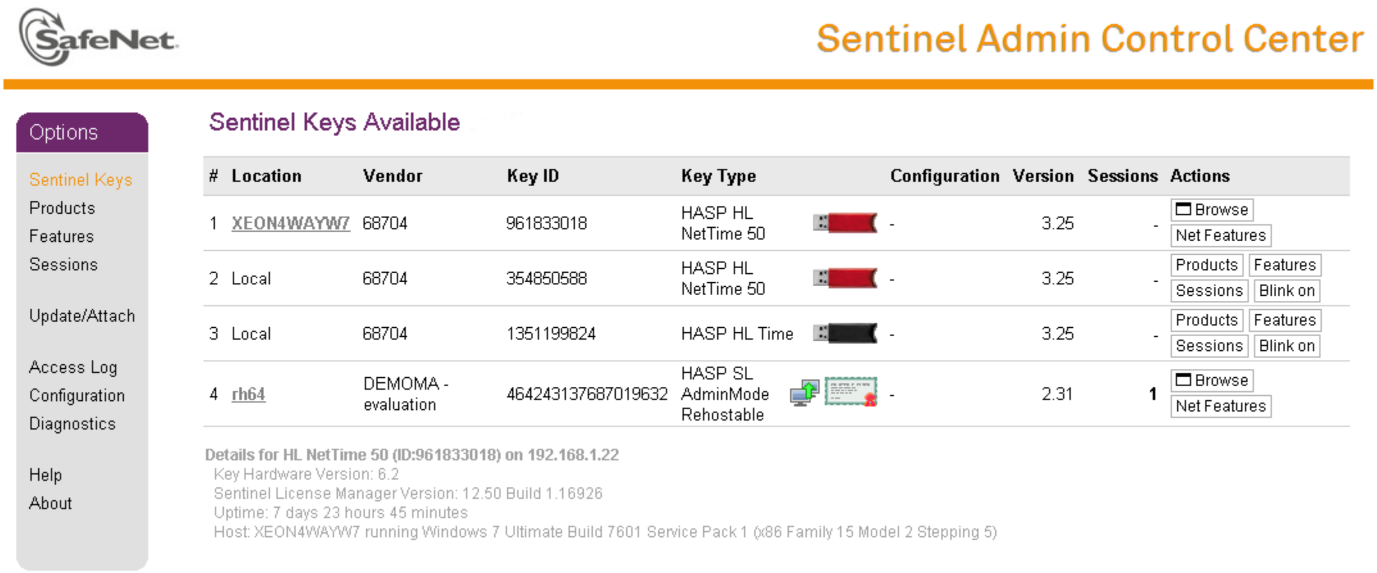
The Sentinel Admin Control Center (ACC) is the web-interface to the Sentinel run-time environment. It allows viewing/managing available keys, enabling and disabling them, controlling usage of remote keys etc.

To access the ACC, start any web browser and go to http://localhost:1947

> Sentinel Admin Control Center is not able to display CHARON-AXP licenses - to view key contents, use the "hasp_srm_view" utility.

To access Sentinel Admin Control Center start any web browser, enter http://localhost:1947 and press Enter. web interface of the Sentinel Admin Control Center will appear.

The screenshot below gives an example of its interface:



#	Location	Vendor	Key ID	Key Type	Configuration	Version	Sessions	Actions
1	XEON4WAYW7	68704	961833018	HASP HL NetTime 50	-	3.25	-	Browse Net Features
2	Local	68704	354850588	HASP HL NetTime 50	-	3.25	-	Products Features Sessions Blink on
3	Local	68704	1351199824	HASP HL Time	-	3.25	-	Products Features Sessions Blink on
4	rh64	DEMOMA - evaluation	464243137687019632	HASP SL AdminMode Rehostable	-	2.31	1	Browse Net Features

Details for HL NetTime 50 (ID:961833018) on 192.168.1.22
Key Hardware Version: 6.2
Sentinel License Manager Version: 12.50 Build 1.16926
Uptime: 7 days 23 hours 45 minutes
Host: XEON4WAYW7 running Windows 7 Ultimate Build 7601 Service Pack 1 (x86 Family 15 Model 2 Stepping 5)

This example demonstrates that 4 license keys are available:

1. Network key ("*HASP-HL NetTime*") on the host "XEON4WAYW7"
2. Network key installed locally
3. HASP-HL installed locally
4. Network-wide software license on the host "RH64"

Sentinel Admin Control Center reports that there is one opened session on key (4). The other keys are not being used at the moment

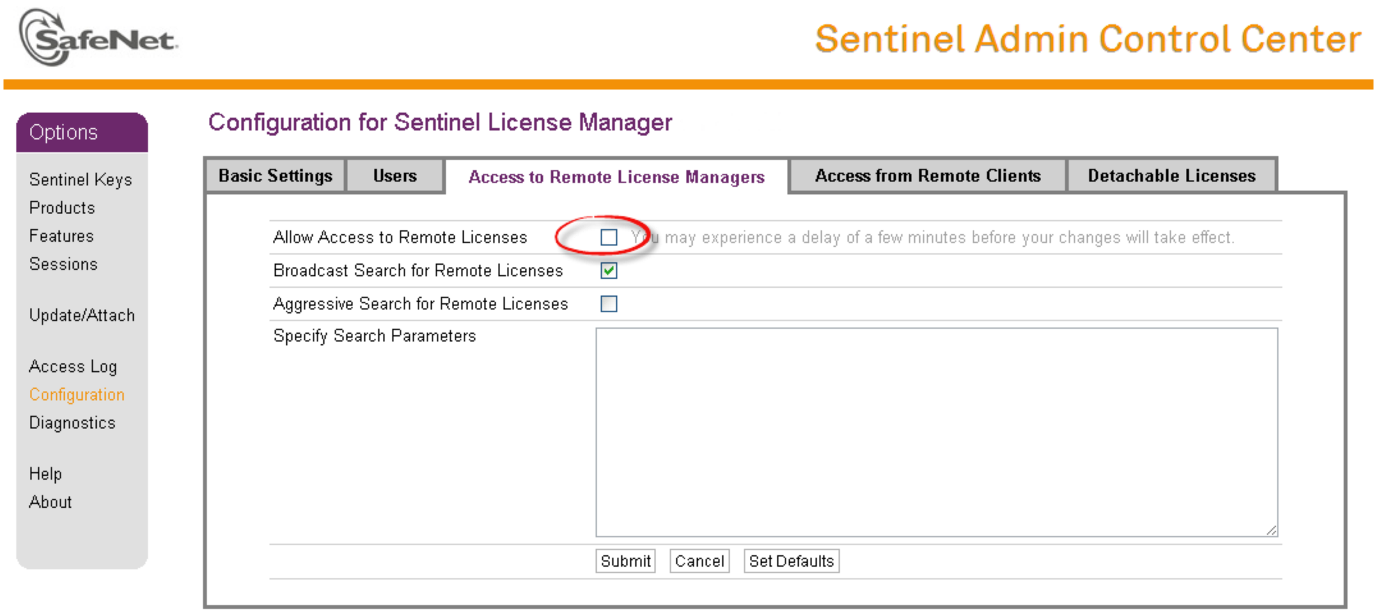
Using Sentinel Admin Control Center it is possible to check available keys, verify hosts on which they reside, verify opened sessions etc. For a more detailed description of Sentinel Admin Control Center, please refer to its "Help" section.

Back to Table of Contents

Disable remote keys access

A helpful feature of Sentinel Admin Control Center is the ability to disable access to remote keys. If the network key is installed locally, access to the key from remote hosts can be disabled. The following examples demonstrate how this can be done.

To disable access to remote keys switch to the "Access to Remote License managers" tab and uncheck the "Allow Access to Remote Licenses" checkbox. Then press "Submit" button to apply this setting:



Configuration for Sentinel License Manager

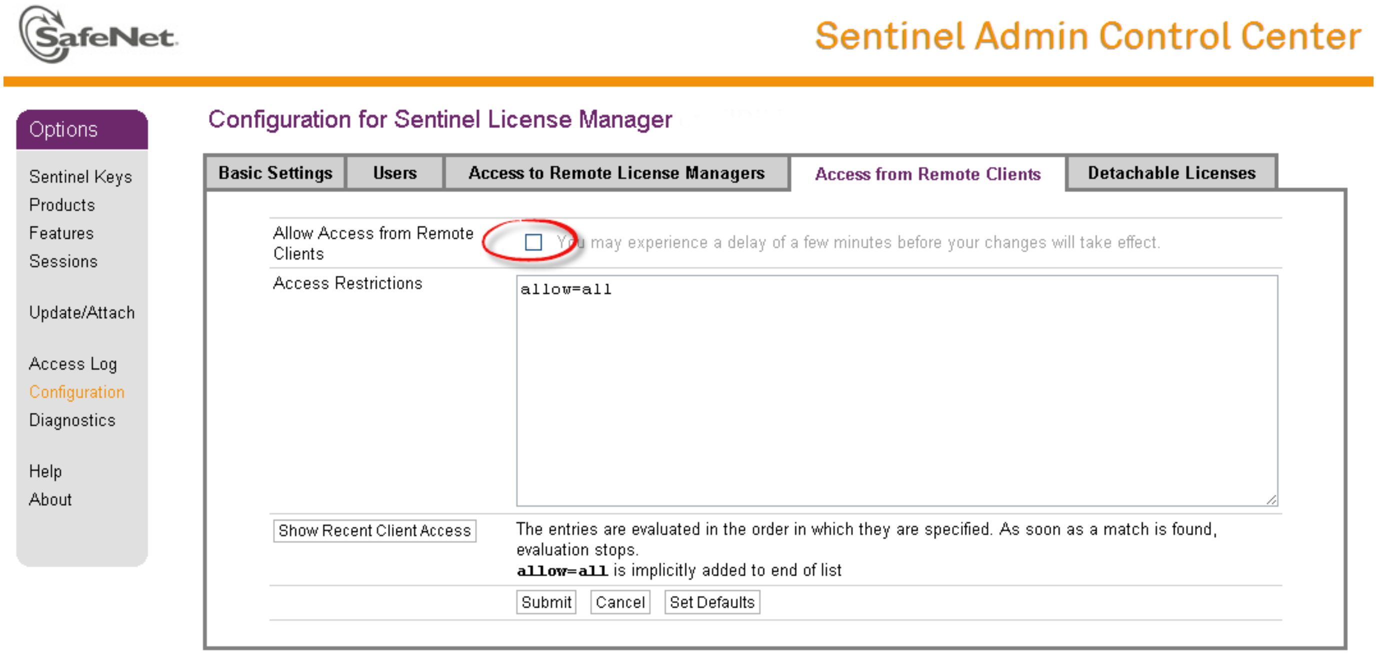
Allow Access to Remote Licenses ☐ You may experience a delay of a few minutes before your changes will take effect.

Broadcast Search for Remote Licenses ☑

Aggressive Search for Remote Licenses ☐

Specify Search Parameters

To disable access to the locally installed license key from remote hosts switch to the "Access from Remote Clients" tab and uncheck the "Allow Access from Remote Clients" checkbox. Then press "Submit" button to apply this setting:



Configuration for Sentinel License Manager

Allow Access from Remote Clients ☐ You may experience a delay of a few minutes before your changes will take effect.

Access Restrictions

```
allow=all
```

The entries are evaluated in the order in which they are specified. As soon as a match is found, evaluation stops.
allow=all is implicitly added to end of list

Back to Table of Contents

Accessing Sentinel Admin Control Center from remote hosts

By default, Sentinel Admin Control Center forbids accessing its web interface from remote machines. To allow access, configure ACC for remote management.

The first step is to edit the "`hasplm.ini`" file:

```
# vi /etc/hasplm/hasplm.ini
```

Allow remote access by changing the "ACCremote" parameter from "0" to "1".Then restart Sentinel Admin Control Center run-time:

```
# /etc/init.d/aksusbd restart
```

If the CHARON-AXP host firewall is blocking remote access to the Sentinel Admin Control Center, please configure the firewall to open the port 1947 (TCP protocol). Refer to Linux documentation for details on how to configure the firewall. It is also possible to use SSH port forwarding with the following command (put the real CHARON-AXP host name instead of "*CHARON_MACHINE*"):

```
# ssh -L8080:CHARON_MACHINE:1947 root@CHARON_MACHINE
```

This will expose Sentinel Admin Control Center on port 8080 to any computer, and Sentinel Admin Control Center will believe commands are coming from the local host.

Back to Table of Contents

License management utilities

CHARON-AXP for Linux provides a specific utility for license management - "hasp_srm_view". This utility is used to display CHARON-AXP license content, and to collect key status information and host fingerprint (C2V) files.

> Applying updates (".v2c" files) is typically done using Sentinel Admin Control Center (see above), but alternatively it is also possible to use a specific "hasp_update" utility for that.

Please refer to the Utilities section of this Guide for more details.

Back to Table of Contents

Transferring and removing CHARON-AXP software licenses

Software Licenses Transfer

Software Licenses (SL) can be transferred from one host to another using the "hasp_srm_view" utility and "Sentinel Admin Control Center**"** (ACC) .

The following example demonstrates the transfer procedure. Let's suppose a Software License must be transferred from a host "SourceHost" to a host "RecipientHost":

1. Collect the specific information about the "RecipientHost" to issue a transfer license. To do that run "hasp_srm_view" utility on the "RecipientHost" with the following parameters:

   ```
   $ hasp_srm_view -idf
   ```

   The file "recipient.id" will be created in the current directory.

2. Copy the "recipient.id" file to the "SourceHost".

   > "recipient.id" file is an ASCII file, so use "ascii" option in case of FTP transfer.

3. On "SourceHost", open "Sentinel Admin Control Center" (ACC) (browse to http://localhost:1947). Note the number of the software license you are going to transfer.

4. Run the "hasp_srm_view" utility in the following way to create a transfer license for the host "RecipientHost":

   ```
   $ hasp_srm_view -tfr <license number> recipient.id
   ```

   The "license number" is the value collected at step 3. Example of collecting a transfer license:

   ```
   $ hasp_srm_view -tfr 12345678 recipient.id
   ```

The file "<license number>.v2c" will then be created in the current directory. In the example above the name of the transfer license will be "*12345678*.v2c"

5. Copy the resulting "<license number>.v2c" file to the "RecipientHost".

> "<license number>.v2c" file is an ASCII file, so use "ascii" option in case of FTP transfer.

6. On "RecipientHost", open "Sentinel Admin Control Center" (ACC) (browse to http://localhost:1947). Apply the "<license number>.v2c" file as described above

Back to Table of Contents

Software License Removal

It is also possible to remove Software License completely from a host, the license will then be dumped to a specific license file "*.v2c", so it can be re-applied if needed.
To remove the Software License completely from a host, do the following::

1. Open "Sentinel Admin Control Center" (ACC) (browse to http://localhost:1947). Note the number of the software license you are going to remove.
2. Run the "hasp_srm_view" utility in the following way to remove the license:

```
$ hasp_srm_view -tfr <license number>
```

The "license number" is the value collected at step 1. Example:

```
$ hasp_srm_view -tfr 12345678
```

The "<license number>.v2c" file will then be created in the current directory. In the example above the name of the transfer license will be "*12345678*.v2c"

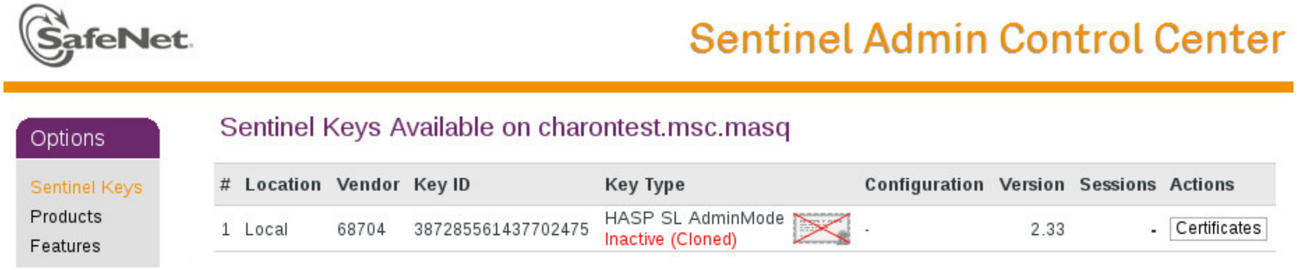
3. It is always possible to re-apply the created ".v2c" file to restore the deleted software license.

Back to Table of Contents

Cloned Software License Removal

In certain situations Software License may become "Cloned" (disabled). In this case the following procedure must be applied to remove the cloned license:

1. Go to http://localhost:1947 to access the "Sentinel HASP Admin Control Center" (ACC).
2. In the "Sentinel HASP Admin Control Center" (ACC), locate the target "Sentinel SL AdminMode" license.
3. Press the "Certificates" button at the right side of the SL description:



Sentinel Keys Available on charontest.msc.masq

#	Location	Vendor	Key ID	Key Type	Configuration	Version	Sessions	Actions
1	Local	68704	387285561437702475	HASP SL AdminMode Inactive (Cloned)	-	2.33	-	Certificates

4. Note the name of the correspondent certificate and path to the certificates base in the "Certificates" section.
5. Remove the target certificate file from the specified directory (in most cases it is "/var/hasplm/installed/68704/").
6. Reboot CHARON host.
7. Start "Sentinel HASP Admin Control Center" (ACC) again to ensure that the SL has been removed.

Back to Table of Contents

License Deinstallation

To completely remove a CHARON-AXP license from a host, it is enough to remove the Sentinel run-time daemon (and the package "charon-hasp-<...>.rpm" containing the run-time customization) using the following command:

```
# rpm --nodeps -e aksusbd charon-hasp-<...>
```

Then just physically disconnect the license key (in the case of protection by dongles).

Back to Table of Contents

Special "backup" license keys

Backup keys are provided by STROMASYS along with standard license dongles. It is strongly recommended to order a backup key to recover immediately from damage or loss of the main license key. Backup keys use a counter (integer) value hardcoded inside the key. This integer value is a number of hours CHARON-AXP is allowed to run. Each time CHARON-AXP checks the license (every hour), the value is decreased (by 1 hour). Please note that backup keys have restricted functionality:

- CHARON run time is typically limited to 720 hours (30 days). This should be more than enough time to get a replacement key from STROMASYS.
- Backup license may be valid only until a certain date. Please check with STROMASYS management.

Back to Table of Contents

CHARON-AXP for Linux utilities

Table of Contents

General description

CHARON-AXP provides the following set of utilities:

Utility	Description
mkdskcmd	Used to create CHARON virtual disk containers of custom or standard types. This utility also may be used to transfer virtual disks of one type to virtual disks of another type.
hasp_srm_view	Used to display the CHARON license contents, to collect the host system fingerprint and to transfer software licenses from one host to another.
hasp_update	Sentinel standard utility used to retrieve Sentinel protection key information, detach a license from a Sentinel SL key and rehost a license from a Sentinel SL key
ncu	Used to dedicate a host interface to CHARON-AXP, to release it back to the host and to manage CHARON virtual interfaces (TAPs).
mtd	Used to create CHARON tape images from physical tapes and to write tape images back to physical tapes.

All these utilities are invoked from Linux console command line.

Back to Table of Contents

'mkdskcmd' utility

Creating disk images

The "mkdskcmd" utility:

- Creates empty disk images of a given standard disk type or a custom disk size
- Transfers existing disk images of one type to disk images of another type.

The first step is to obtain the name of the disk that needs to be created:

```
$ mkdskcmd --list
```

This command results in a list of all supported disk types.

Choose the desired disk (for example "RZ22"), then use the "mkdskcmd" command to create the virtual disk image as shown below:

```
$ mkdskcmd --disk rz22 --output rz22.vdisk
```

A disk container "rz22.vdisk" will be created in the current directory.

> A file "rz22.avdisk" will also be created. This file helps CHARON accurately recognize a specific disk image type. It is recommended to put the ".avdisk" file in the same directory as the created disk image.

It is also possible to create custom disk images using "--blcount" (blocks count) and "--blsize" (blocks size) switches.

To get all the available parameters please use the "–help"switch:

```
Usage:
   mkdskcmd [Options]

Options:

   --help - to see help screen

   --h - to see help screen

   --output <full name> - to specify output file name

   --o <full name> - to specify output file name

   --disk <disk name> - to specify the disk name from Disk table
   -d <disk name> - to specify the disk name from Disk table

   --blsize <number> - to specify the block size in bytes (custom disk image)
   -z <number> - to specify the block size in bytes (custom disk image)

   --blcount <number> - to specify number of the blocks (custom disk image)
   -c <number> - to specify number of the blocks (custom disk image)

   --avtable <full_name> - to specify AVDISK table file
   -a <full_name> - to specify AVDISK table file

   -t - please see the '-transform' options description
  --transform <source_disk_name> <source_disk_params> - to transform the the disk image (change actual
size)

   <source_disk_name> - the file name of the disk image to be transformed
   <source_disk_params> - the name of the disk from the list of available at the Disk table

   The source disk size will changed accordingly the reach the specified parameters.
   To specify the transform parameters manually, follow the option below:

   --transform <source_disk_name> --blsize <number> --blcount <number>

   --shrink - parameter which needs to be EXPLICITLY provided, if the disk size is to be decreased
   -k - parameter which needs to be EXPLICITLY provided, if the disk size is to be decreased

   --list <full_name> - to display AVDISK table
   -l <full_name> - to display AVDISK table

   --silent - silent mode running
   -s - silent mode running

   --transfer - please see the '--transfer' options description
   -t - please see the '-t' options description

Return value:
   0 - for Success
   Non zero - in case of failure

Examples:
   mkdskcmd -h
   mkdskcmd -l
   mkdskcmd -a /opt/charon/bin/mkdsk.vtable -o /etc/rk07.vdisk -d rk07
   mkdskcmd -o /etc/custom.vdisk -z 512 -c 16384
```

The "--avtable" parameter is used to work with an alternative disk specification database (or to point to the standard database ("mkdsk.vtable") if it is in a location other than the current directory).

The "--blcount" (blocks count) and "--blsize" (blocks size) switches are used to create custom disk images.

Back to Table of Contents

Transferring disk images

The "mkdskcmd" utility is able to transfer (copy) disk images of one type to a disk image of another type.

This operation is needed, for example, to obtain more free space on a disk image that already contains data.

Note: it is not possible to add more free space dynamically. CHARON-AXP must be stopped before performing this operation.

> If a source disk image is larger than the target disk image, the extra data is lost. If the source disk image is smaller, it will be extended and padded with null bytes ('\0').

An example of the syntax follows:

```
$ mkdskcmd --transfer <source disk file name> <source disk parameters> [--shrink] [-k]
```

where:

- <source disk file name> - a file name of the disk image to be transferred
- <source disk parameters> - the name of the disk from the list provided by the "mkdskcmd --list" command execution or the disk geometry specification (see below).
- --shrink or -k - used in the case where the target disk is transferred to a smaller disk.

Example:

```
$ mkdskcmd --transfer /etc/rz22.vdisk rz25
```

It is also possible to specify the disk parameters manually with "--blcount / -c" (blocks count) and "--blsize / -z" (blocks size) switches:

```
$ mkdskcmd --transfer <source disk file name> -blsize <number> -blcount <number>
```

Example:

```
$ mkdskcmd -t /etc/custom.vdisk -z 512 -c 262134
```

> There is a certain delay between the moment when the utility reports that a disk image has been transferred and its actual availability to CHARON. This delay can reach to several minutes in case of very big disks transfers. It happens because the host operating systems needs some time for actual allocation of the enlarged file on HDD.

Back to Table of Contents

'mtd' utility

The "mtd" utility is used to:

- Create a CHARON tape image from a physical tape
- Write a tape image to a physical tape.

Usage is the following:

```
$ mtd [options] <tape device name> <tape container name>
```

where the options are:

Parameter	Description
`-l <file name>`	Creates an execution log “file name”.
`-r <number>`	Specifies a number of attempts to read a damaged data bock
`-i`	Directs to ignore bad blocks and continue processing without interruption. It implies "-r 0"
`-n`	Do not rewind tape
`-p`	Disable progress reporting
`-v`	Enable verbose trace of data transfer (implies "-p")

Example:

```
$ mtd -l tape1.txt -r 10 /dev/st5 /charon/tapes/tape1.vtape
```

Use the following syntax to write the content of a tape container to a physical tape:

```
$ mtd <tape container name> <tape device name>
```

Example:

```
$ mtd /charon/tapes/tape1.vtape /dev/st5
```

Back to Table of Contents

'hasp_srm_view' utility

The "hasp_srm_view" utility displays content of CHARON-AXP license.

Just run the utility without any parameter to see the license details.

The "hasp_srm_view" utility provides the following functionality:

- Display the CHARON-AXP license details
- Collecting license status information
- Collecting host fingerprint information
- Managing software license transfer procedure.

Run the utility without any options to display the license details.

```
# hasp_srm_view -help

CHARON Sentinel HASP utility
Copyright: STROMASYS SA, 2013

Options:
  -? or -h or -help - to see help screen
  -l - to see CHARON license details
  -c2v <C2V file> - to collect the key status information (C2V file)
  -fgp <C2V file> - to collect the host fingerprint information (C2V file)
  -tfr <LicenseID> <recipient file> - to transfer HASP SL license (V2C file)
  -tfr <LicenseID> - to remove HASP SL license (V2C file) from the local host
  -idf - to get transfer recipient (ID) file "recipient.id"
```

The specific type of CHARON license defines what switches may be used in each case.

Collecting the "c2v" file can be done only from the CHARON host console.

Remote collection of status information

For remote collection of status information it is recommended to use "ssh" as shown in the following examples:

```
# ssh root@CHARON_HOST /opt/charon/bin/hasp_srm_view -c2v /opt/charon/bin/my_hasp_key.c2v
# ssh root@CHARON_HOST /opt/charon/bin/hasp_srm_view -fgp /opt/charon/bin/my_host_fingerprint.c2v
```

To see the license text on the console:

```
# ssh root@localhost /opt/charon/bin/hasp_srm_view
```

To collect license text to an output file on host server:

```
# ssh root@localhost /opt/charon/bin/hasp_srm_view > /opt/charon/bin/hasp_srm_view.txt
```

The "hasp_srm_view" utility always reports the ID and IP address of the host(s) where active licenses are found.

Back to Table of Contents

Software Licenses Transfer

Software Licenses (SL) can be transferred from one host to another one with the help of "hasp_srm_view" utility and "Sentinel Admin Control Center**"** (ACC).

The following example demonstrates the transfer procedure.

Let's suppose a Software License must be transferred from a host "SourceHost" to a host "RecipientHost":

1. Run "hasp_srm_view" utility on the "RecipientHost" with the following parameters to collect the host ID info:

   ```
   $ hasp_srm_view -idf
   ```

   The "recipient.id" file will be created in the current directory.

2. Copy the "recipient.id" file to the "SourceHost".

   > "recipient.id" is an ASCII file, so use the "ascii" option for FTP transfer.

3. On the "SourceHost", open the "Sentinel Admin Control Center" (ACC) (http://localhost:1947). Note the number of the software license you are going to transfer.

4. Run the "hasp_srm_view" utility in the following way to create a transfer license for the host "RecipientHost":

   ```
   $ hasp_srm_view -tfr <license number> recipient.id
   ```

   The "license number" is the value collected at the step 3.

   Example of collecting a transfer license:

   ```
   $ hasp_srm_view -tfr 12345678 recipient.id
   ```

   A "<license number>.v2c" file will be created in the current directory. In the example above, the name of the transfer license will be "*12345678*.v2c"

5. Copy the resulting "<license number>.v2c" file to the "RecipientHost".

   > "<license number>.v2c" is an ASCII file, so use the "ascii" option for FTP transfer.

6. On the "RecipientHost", open "Sentinel Admin Control Center" (ACC) (http://localhost:1947) and apply the "<license number>.v2c" file as described above.

Back to Table of Contents

Software Licenses Removal

When a Software License is removed completely from a host, the license is dumped to a specific license file ".v2c". The license is not destroyed and can be re-applied if needed.

To remove a software license from a host do the following:

1. Open "Sentinel Admin Control Center" (ACC) (http://localhost:1947). Note the number of the software license you are going to remove.

2. Run the "hasp_srm_view" utility in the following way to remove the license:

   ```
   $ hasp_srm_view -tfr <license number>
   ```

   The "license number" is the value collected at step 1.

   Example:

   ```
   $ hasp_srm_view -tfr 12345678
   ```

   The "<license number>.v2c" file will then be created in the current directory. In the example above the name of the transfer license will be "*12345678*.v2c"

3. It is always possible to re-apply the created ".v2c" file to restore the deleted software license.

Back to Table of Contents

'hasp_update' utility

The "hasp_update" is a Sentinel standard utility for license management included in CHARON-AXP kit.

To invoke the "hasp_update" utility login as "root" and use the following syntax:

```
# hasp_update <option> [filename]
```

where:

Parameter	Value	Description
<option>	u	Updates a Sentinel protection key / attaches a detached license
	i	Retrieves Sentinel protection key information
	d	Detaches a license from a Sentinel Software License (SL) key
	r	Rehost a license from a Sentinel Software License (SL) key
	h	Display help
[filename]		Path to the V2C/H2R file when used with the 'u' option
		Optional path to the C2V file when used with the 'i' option Uses "stdout" if file name is not specified

Example:

```
# hasp_update u license_update.v2c
```

We recommend to use this tool only for "Update a Sentinel protection key / attach a detached license" function ("u" option). For the rest use "hasp_srm_view" utility.

Back to Table of Contents

'ncu' utility

The "ncu" ("Network Control Utility") is used to dedicate a host interface to CHARON-AXP, to release it back to the host and to manage CHARON virtual interfaces (TAPs).

The utility allocates chosen network interfaces (both physical and virtual) and configures the offload parameters.

Dedication of a host physical interface to CHARON

Login as root and enter "ncu". The following menu will appear:

```
# ncu
CHARON Network Configuration Utility, STROMASYS (c) 2015 Version 1.4

Interfaces Dedicated to State
---------- ------------ ------------
eth0       host         connected to host
eth1       host         connected from host
lo         host         unmanaged from host

==================================================================
bridge name     bridge id        STP enabled            interfaces

select action:
1 - Dedicate to CHARON
2 - Release to host
3 - Create Bridge with TAPs
4 - Remove Bridge
5 - Print status
6 - Exit
:> 6
```

The utility lists available network interfaces (both physical and virtual) and indicates whether they are dedicated to the host or to CHARON and whether they are currently in use by host operating system.

"ncu" offers several options:

1. Dedicate interface to CHARON
2. Release interface to host
3. Create a bridge between a chosen physical network interface and the Linux virtual network and create a number of virtual network interfaces
4. Remove the Linux virtual network and all the created virtual network interfaces
5. Print status - use it to display status of network interfaces and the menu shown above
6. Exit

In the example above we see 2 network interfaces - "eth0" and "eth1", both of them are dedicated to host, but host uses only the interface "eth0".

Let's dedicate the interface "eth1" to CHARON-AXP.

Enter "1", then type "eth1" and press Enter:

```
Specify the interface to dedicate to CHARON:eth1
Turning off offloading for eth1.. Please wait

select action:
1 - Dedicate to CHARON
2 - Release to host
3 - Create Bridge with TAPs
4 - Remove Bridge
5 - Print status
6 - Exit
:> 5
```

Now the interface "eth1" is dedicated to CHARON-AXP:

```
Interfaces Dedicated to State
---------- ------------ ------------
eth0       host         connected to host
eth1       CHARON       disconnected from host
lo         host         unmanaged from host

==================================================================
bridge name     bridge id        STP enabled            interfaces

select action:
1 - Dedicate to CHARON
2 - Release to host
3 - Create Bridge with TAPs
4 - Remove Bridge
5 - Print status
6 - Exit
```

Enter "6" to return to console prompt.

Now "eth1" can be used by CHARON-AXP.

Back to Table of Contents

Release of a host physical interface back to host

Login as root and enter "ncu". The following menu will appear:

```
# ncu
CHARON Network Configuration Utility, STROMASYS (c) 2015 Version 1.4

Interfaces Dedicated to State
---------- ------------ ------------
eth0       host         connected to host
eth1       CHARON       disconnected from host
lo         host         unmanaged from host

==================================================================
bridge name     bridge id        STP enabled            interfaces

select action:
1 - Dedicate to CHARON
2 - Release to host
3 - Create Bridge with TAPs
4 - Remove Bridge
5 - Print status
6 - Exit
:> 2
```

Let's say that we want to return the interface "eth1" (currently dedicated to CHARON) back to host. To do that enter "2" then "eth1":

```
Specify the interface to release to HOST:eth1
Connection successfully activated (D-Bus active path: /org/freedesktop/NetworkManager/ActiveConnection/3)

select action:
1 - Dedicate to CHARON
2 - Release to host
3 - Create Bridge with TAPs
4 - Remove Bridge
5 - Print status
6 - Exit
:> 6
```

Enter "6" to quit the "ncu" utility.

The interface "eth1" is released back to host system now.

Back to Table of Contents

Creation of a virtual network

Login as root and enter "ncu":

```
# ncu
CHARON Network Configuration Utility, STROMASYS (c) 2015 Version 1.4

Interfaces Dedicated to State
---------- ------------ ------------
eth0       host         connected to host
eth1       host         connected to host
lo         host         unmanaged from host

==================================================================
bridge name     bridge id        STP enabled            interfaces

select action:
1 - Dedicate to CHARON
2 - Release to host
3 - Create Bridge with TAPs
4 - Destroy Bridge
5 - Print status
6 - Exit
:> 3
```

Enter "3" to create a bridge between the host physical network adapter and the LINUX virtual network interfaces (TAP) and specify the physical network interface ("eth1" in our example) and the number of virtual network interfaces to be created (2 in our example):

```
Specify the interface to be used for BRIDGE:eth1
How many tap should be created:2
Forming the bridge: ..1..2..3..4..5.. addif tap0 .. addif tap1 ..7..8 done!
Formed bridge br0_eth1 attached over eth1...

select action:
1 - Dedicate to CHARON
2 - Release to host
3 - Create Bridge with TAPs
4 - Remove Bridge
5 - Print status
6 - Exit
:> 5
```

Now enter "5" to see the created virtual interfaces:

```
Interfaces   Dedicated to    State
----------   ------------    ------------
eth0         host            connected to host
eth1         bridge          connected to bridge
lo           host            unmanaged from host
tap0         bridge          connected to bridge
tap1         bridge          connected to bridge
==================================================================
bridge name     bridge id           STP enabled        interfaces
br0_eth1        8000.525400698995a  no                 tap0
                                                       tap1

select action:
1 - Dedicate to CHARON
2 - Release to host
3 - Create Bridge with TAPs
4 - Remove Bridge
5 - Print status
6 - Exit
:> E
```

In the example above we see 2 virtual network Interfaces "tap0" and "tap1" connected to the created bridge. The physical network interface "eth1" is used for the bridge to the virtual network interfaces.

The interfaces "tap0" and "tap1" are ready to be used in CHARON configurations - they do not need to be additionally dedicated to CHARON.

Enter "6" to quit "ncu" utility.

Back to Table of Contents

Removal of a virtual network

Login a root. Start "ncu" utility:

```
# ncu

CHARON Network Configuration Utility, STROMASYS (c) 2015 Version 1.4

Interfaces   Dedicated to    State
----------   ------------    ------------
eth0         host            connected to host
eth1         bridge          connected to bridge
lo           host            unmanaged from host
tap0         bridge          connected to bridge
tap1         bridge          connected to bridge
==================================================================
bridge name     bridge id           STP enabled        interfaces
br0_eth1        8000.525400698995a  no                 tap0
                                                       tap1

select action:

1 - Dedicate to CHARON
2 - Release to host
3 - Create Bridge with TAPs
4 - Remove Bridge
5 - Print status
6 - Exit
:> 4
```

Enter "4" then enter the interface name that is a bridge to the Linux virtual network on this host ("eth1" in our example):

```
Specify the phys interface used for BRIDGE:eth1
Cleanup bridge br0_eth1 with ip over eth1...
Removing the bridge: ..1..2 delif eth1
delif tap0
delif tap1
..5..6..7..8 done!

select action:
1 - Dedicate to CHARON
2 - Release to host
3 - Create Bridge with TAPs
4 - Remove Bridge
5 - Print status
6 - Exit
:> 6
```

Enter "6" to quit "ncu" utility.

Back to Table of Contents

CHARON-AXP for Linux configuration details

Introduction

This chapter describes, in detail, all of the configuration parameters of the devices emulated by CHARON-AXP for Linux, with corresponding examples and parameters.

Emulated devices are loaded with the "load" command (if a device has not been already loaded) and parameters are made active with the "set" command. Parameters can be specified directly in the "load" command.

Example:

```
load KZPBA DKA
set DKA container[0]="/my_disks/bootable_disk.vdisk"
```

In this example, an instance of a KZPBA controller is loaded with the name "DKA". Its first unit, "container[0]", is mapped to the "/my_disks/bootable_disk.vdisk" disk image.

> The Controller name is accompanied with a "/<module name>". The module name is a CHARON-AXP component that specifies the controller load module. Its name can be the same as the loaded controller, however this is not mandatory. Once a module name is specified, there is no need to specify it again for additional references of the same controller.

Details of CHARON-AXP configuration

Back to Table of Contents

General Settings

Session

General settings that control the execution of CHARON-AXP belong to an object called the "session". It is a preloaded object; therefore, only "set" commands apply.

Example:

```
set session <parameter>=<value>
```

The following table describes all available "session" parameters, their meaning and examples of their usage:

hw_model

Parameter	hw_model
Type	Text string
Value	Virtual HP Alpha system hardware model to be emulated. Use a default configuration template for each particular model as a starting point for a custom configuration. This would ensure that the parameter is set correctly. Example: `set session hw_model="AlphaServer_ES40"` Available models are: • AlphaServer_AS400 • AlphaServer_AS800 • AlphaServer_AS1000 • AlphaServer_AS1000A • AlphaServer_AS1200 • AlphaServer_AS2000 • AlphaServer_AS2100 • AlphaServer_AS4000 • AlphaServer_AS4100 • AlphaServer_DS10 • AlphaServer_DS15 • AlphaServer_DS20 • AlphaServer_DS25 • AlphaServer_ES40 • AlphaServer_ES45 • AlphaServer_GS80 • AlphaServer_GS160 • AlphaServer_GS320 ⓘ Refer to this section to find how to set a particular HP Alpha model supported by CHARON-AXP.

configuration_name

Parameter	configuration_name
Type	Text string

Value	Name of the CHARON-AXP instance (ir must be unique):
	`set session configuration_name="MSCDV1"`
	The value of this parameter is used as a prefix to the event log file name. (see below).
	From the example above, the CHARON-AXP log file will have the following name:
	`MSCDV1-YYYY-MM-DD-hh-mm-ss-xxxxxxxxx.log`
	xxxxxxxxx is an increasing decimal number starting from 000000000 to separate log files with the same time of creation (in case the log is being written faster than one log file per second).
	⚠ It is strictly recommended to use the "configuration_name" parameter if more than one CHARON instance runs on the same server.

log

Parameter	log
Type	Text string

Value	The log file or directory name is where the log file for each CHARON-AXP execution session is stored. If an existing directory is specified, CHARON-AXP automatically creates individual log files for each CHARON-AXP execution session. If the log parameter is omitted CHARON-AXP creates a log file for each CHARON-AXP execution session in the directory where the emulator was started.
	It is possible to overwrite the existing log file or extend it using the "log_method" parameter. See below for details. ℹ The "log_method" parameter is effective only in the case where a single log file is specified, not a directory.
	Examples:
	`set session log="log.txt"`
	`set session log="/CharonLogs/Logs"`
	If only a directory name is specified in the "log" parameter and the "configuration_name" parameter of the session is specified, the log file name will have the following format:
	`<configuration_name>-YYYY-MM-DD-hh-mm-ss-xxxxxxxxx.log`
	If only a directory name is specified in the "log" parameter and the "configuration_name" parameter is omitted, the log file name will have the following format:
	`<hw_model>-YYYY-MM-DD-hh-mm-ss-xxxxxxxxx.log`
	xxxxxxxxx is an increasing decimal integer, starting from 000000000 to separate log files with the same time of creation (in case the log is being created faster than one log file per second).
	Only existing directory can be used as a value of the "log" parameter.

log_method

Parameter	log_method
Type	Text string
Value	• "overwrite" (default) • "append" Determines if the previous log information is maintained or overwritten. This parameter must be specified only in addition to "log" parameter on the same line. This parameter is applicable only if the CHARON-AXP log is stored to a file that is specified explicitly with the "log" parameter. Example: `set session log="log.txt" log_method="append"`

license_key_id

Parameter	license_key_id[N] N=0 or 1
Type	Numeric

Value	An integer (decimal Sentinel Key ID) that specifies the regular (N=0) and backup (N=1) license keys to be used by CHARON-AXP.
	`set session license_key_id[0]=1877752571`
	`set session license_key_id[1]=354850588`
	It is also possible to specify both regular and backup keys in one line:
	`set session license_key_id[0]=1877752571 license_key_id[1]=354850588`
	Based on the presence of the regular and/or backup license key IDs in the configuration file, CHARON-AXP behaves as follows:
	1. **No keys are specified** CHARON-AXP performs an unqualified search for any suitable key. If no key is found, CHARON-AXP exits.
	2. **Both keys are specified** CHARON-AXP performs a qualified search for a regular license key. If it is not found, CHARON-AXP performs a qualified search for backup license key. If it is not found, CHARON-AXP exits.
	3. **Only regular key is specified** CHARON-AXP performs a qualified search for a regular license key. If it is not found, CHARON-AXP performs an unqualified search for any suitable key. If none are found, CHARON-AXP exits.
	4. **Only backup key is specified** CHARON-AXP performs an unqualified search for any suitable key. If no key is found, CHARON-AXP exits.

affinity

Parameter	affinity
Type	Text string
Value	Overrides any initial process affinity mask provided by the host operating system. Once specified it binds the running instance of the emulator to particular host CPUs. Used for soft partitioning of the host CPU resources and/or for isolating host CPUs for other applications. By default the CHARON-AXP emulator instance allocates as many host CPUs as possible. The “affinity” parameter overrides that and allows explicit specification on which host CPU the instance must run on. The "affinity" parameter defines the total number of host CPUs to be used both for emulated Alpha CPUs and for CHARON-AXP application itself (including the CPUs to be used for I/O - they are controlled by "n_of_io_cpus" parameter described below). Host CPUs are enumerated as a comma separated list of host system assigned CPU numbers: `set session affinity="0, 2, 4, 6"`

n_of_cpus

<table>
<tr><td><b>Parameter</b></td><td>n_of_cpus</td></tr>
<tr><td><b>Type</b></td><td>Numeric</td></tr>
<tr><td><b>Value</b></td><td>Limits the number of emulated CPUs.
Example:
<code>set session n_of_cpus=3</code>
The maximum number of CPUs enabled by CHARON-AXP is specified by the license key, but cannot exceed the original hardware restrictions:
<table>
<tr><th>HP Alpha Model</th><th>Number of emulated CPUs</th></tr>
<tr><td>AlphaServer_AS400</td><td>1</td></tr>
<tr><td>AlphaServer_AS800</td><td>1</td></tr>
<tr><td>AlphaServer_AS1000</td><td>1</td></tr>
<tr><td>AlphaServer_AS1000A</td><td>1</td></tr>
<tr><td>AlphaServer_AS1200</td><td>2</td></tr>
<tr><td>AlphaServer_AS2000</td><td>2</td></tr>
<tr><td>AlphaServer_AS2100</td><td>4</td></tr>
<tr><td>AlphaServer_AS4000</td><td>2</td></tr>
<tr><td>AlphaServer_AS4100</td><td>4</td></tr>
<tr><td>AlphaServer_DS10</td><td>1</td></tr>
<tr><td>AlphaServer_DS15</td><td>1</td></tr>
<tr><td>AlphaServer_DS20</td><td>2</td></tr>
<tr><td>AlphaServer_DS25</td><td>2</td></tr>
<tr><td>AlphaServer_ES40</td><td>4</td></tr>
<tr><td>AlphaServer_ES45</td><td>4</td></tr>
<tr><td>AlphaServer_GS80</td><td>8</td></tr>
<tr><td>AlphaServer_GS160</td><td>16</td></tr>
<tr><td>AlphaServer_GS320</td><td>32</td></tr>
</table>
</td></tr>
</table>

n_of_io_cpus

Parameter n_of_io_cpus

Type Numeric

Value This parameter specifies how many host CPUs CHARON-AXP must use for I/O handling. Use of the “affinity” parameter may limit the number of CPUs available.

By default the CHARON-AXP instance reserves one third of all available host CPUs for I/O processing (round down, at least one). The “n_of_io_cpus” parameter overrides that by specifying the number of CHARON I/O CPUs explicitly.

Example:

```
set session n_of_io_cpus=2
```

license_key_lookup_retry

Parameter license_key_lookup_retry

Type Text String

Value In case the CHARON-AXP license connection is not present when the guest starts up, this parameter specifies how many times CHARON-AXP will try to establish the connection and, optionally, a period of time between retries.

Syntax:

```
set session license_key_lookup_retry = "N [, T]"
```

Options:

- N - Number of retries to look for license keys.
- T - Time between retries in seconds. If not specified 60 seconds are used

Example 1

```
set session license_key_lookup_retry = 1
```

If license key is not found during initial scan, do only one more attempt after 60 seconds.

Example 2

```
set session license_key_lookup_retry = "1,30"
```

Same as above but retry in 30 seconds.

Example 3

```
set session license_key_lookup_retry = "3,10"
```

If license key is not found during initial scan, do 3 more attempts waiting 10 seconds between them.

Example 4

```
set session license_key_lookup_retry = "5"
```

If license key is not found during the initial scan, do 5 more attempts waiting 60 seconds between them.

Back to Table of Contents

Core Devices

Table of Contents

Back to Table of Contents

CPU

The CHARON-AXP CPU can be calibrated with "set ace" directive and the following parameters:

enabled

Parameter	enabled
Type	Boolean
Value	A CHARON-AXP emulated CPU is configured with the "enabled" command enabling the high performance Advanced CPU Emulation mode ("ACE"). The ACE option optimizes the HP Alpha instruction interpretation and significantly improves performance. It also requires approximately twice the amount of host memory allocated by CHARON instance itself to store the optimized code (Note that 2Gb of host memory + the amount of HP Alpha memory emulated per each CHARON instance is required). ACE optimization is performed dynamically during execution. It does not need to write optimized code back to disk, ACE provides its full capability instantly. The optimization does not compromise the HP Alpha instruction decoding; CHARON-AXP remains fully HP Alpha hardware compatible and completely transparent to the HP Alpha operating systems and applications. This configuration setting enables the ACE mode if the CHARON-AXP license permits it. If this configuration setting is omitted from the CHARON-AXP configuration file and the license permits it, "true" is the default, otherwise "false" is the default. Example: `set ace enabled = false` "`set ace enabled=true`" is ignored when the license does not permit ACE operation.

cpu_architecture

Parameter	cpu_architecture
Type	Text String
Value	Specifies the architecture of the virtual Alpha CPU. Can be one of the following: EV4, EV45, EV5, EV56, EV6, EV67, EV68 Example: `set ace cpu_architecture = EV6` Refer to this section to find an appropriate value of the HP Alpha architecture per each HP Alpha model supported by CHARON-AXP.

cache_size

Parameter	cache_size
Type	Value
Value	"cache_size" defines the amount of memory in megabytes allocated to the ACE cache. Default value is 1GB (1024 MB). Example: `set ace cache_size=2048`

Back to Table of Contents

RAM

The CHARON-AXP memory subsystem is permanently loaded and has the logical name "ram".

size

Parameter	size
Type	Numeric
Value	Size of the emulated memory in MB.

Example:

```
set ram size = 2048
```

The amount of memory is capped at a maximum, this is defined in the CHARON license key. If the host system cannot allocate enough memory to map the requested emulated memory, CHARON-AXP generates an error message in the log file and reduces its effective memory size.

The following table lists the values of emulated RAM for various hardware models of virtual HP Alpha systems:

Hardware Model	**RAM size (in MB)**			
	Min	**Max**	**Default**	**Increment**
AlphaServer 400	64	1024	512	64
AlphaServer 800	256	8192	512	256
AlphaServer 1000	256	1024	512	256
AlphaServer 1000A	256	1024	512	256
AlphaServer 1200	256	32768	512	256
AlphaServer 2000	64	2048	512	64
AlphaServer 2100	64	2048	512	64
AlphaServer 4000	64	32768	512	64
AlphaServer 4100	64	32768	512	64
AlphaServer DS10	64	32768	512	64
AlphaServer DS15	64	32768	512	64
AlphaServer DS20	64	32768	512	64
AlphaServer DS25	64	32768	512	64
AlphaServer ES40	64	32768	512	64
AlphaServer ES45	64	32768	512	64
AlphaServer GS80	256	65536	512	256
AlphaServer GS160	512	131072	512	512
AlphaServer GS320	1024	262144	1024	1024

Back to Table of Contents

TOY

CHARON-AXP maintains its time and date using the "toy" (time-of-year) component. In order to preserve the time and date while a virtual system is not running, the TOY component uses a binary file on the host system to store the date and time relevant data. The name of the file is specified by the “container” option of the "toy" component.

container

<table>
<tr><td><b>Parameter</b></td><td>container</td></tr>
<tr><td><b>Type</b></td><td>Text string</td></tr>
<tr><td><b>Value</b></td><td>

Specifies a name for the file in which CHARON-AXP preserves the time and date during its “offline” period. This file also keeps some console parameters (such as the default boot device).

By default it is left unspecified.

Example:

```
set toy container="/Charon/my_virtual_system.dat"
```

ⓘ It is recommended to specify the full path to the TOY file.

</td></tr>
</table>

sync_to_host

<table>
<tr><td><b>Parameter</b></td><td>sync_to_host</td></tr>
<tr><td><b>Type</b></td><td>Text string</td></tr>
<tr><td><b>Value</b></td><td>

Specifies whether and how the guest OS time is synchronized with the CHARON host time.

Syntax:

```
set TOY sync_to_host = "{as_vms | as_tru64 | as_is}[, nowrite]"
```

where:

Parameter	Description
as_vms	If the guest OS is OpenVMS/AXP and its date and time must be set to the host's date and time each time it boots.
as_tru64	If the guest OS is Tru64 UNIX and its date and time must be set to the host's date and time each time it boots.
as_is	If the TOY date and time must be set to the host's UTC date and time
nowrite	Disable undesirable updates to the TOY from the guest OS.

Example:

```
set TOY sync_to_host = "as_vms, nowrite"
```

To synchronize the guest OS with TOY, use the following commands (from "SYSTEM"/"root" account):

<table>
<tr><th>On OpenVMS/AXP</th><th>On Tru64 UNIX</th></tr>
<tr><td>

```
$ set time
```

</td><td>

```
# date -u `consvar -g date | cut -f 3 -d ' '`
```

</td></tr>
</table>

The default value is "not specified" - it means that by default CHARON does not synchronize its guest OS time with the CHARON host time but collects date and time from the file specified with "container" parameter.

⚠ If "sync_to_host" parameter is specified there is no need to specify "container" parameter in addition.

</td></tr>
</table>

The CHARON-AXP time zone may be different from that of the host system. Correct CHARON time relies on the correctness of the host system time to calculate the duration of any CHARON “offline” periods. (i.e. while the virtual system is not running). Every time CHARON comes on line it calculates a Delta time (the system time is used if there is no TOY file). Therefore, if the host system time is changed while CHARON is not running, the CHARON time may be incorrect when CHARON is restarted and the CHARON time must be set manually.

Back to Table of Contents

ROM

The System Flash ROM file conserves specific parameters between reboots.

container

Parameter	container
Type	Text string
Value	Specifies the name of a file in which CHARON-AXP stores an intermediate state of its Flash ROM. This state includes, for example, most of the console parameters. By default it is left unspecified. ⓘ it is recommended to specify the full path to this file Example: `set rom container="/Charon/my_virtual_system.rom"`

system_name

Parameter	system_name
Type	Text string
Value	Allows changing the system name, for example: `set rom system_name="Alpha Server 1000 4/200"` Refer to this section to find an appropriate value of the HP Alpha system name per each HP Alpha model supported by CHARON-AXP

system_serial_number

Parameter	system_serial_number
Type	Text string
Value	Allows changing the system serial number, for example: `set rom system_serial_number = NY12345678` Any sequence of characters can be used as a serial number. Sequences longer than 16 symbols are truncated to 16 symbols. Serial Numbers should be according to DEC standard: 10 characters. First two characters are capital letters, remaining 8 characters are decimal digits. By default it is set to SN01234567

dsrdb

Parameter	dsrdb[n]
Type	Numeric
Value	DSRDB - Dynamic System Recognition Data Block. These parameters allow changing the emulated hardware model type. dsrdb[0] stands for SMM - System Marketing Model. Example: `set rom dsrdb[0]=1090` This section describes connection between "dsrdb" parameter and the rest of the parameters defining an exact HP Alpha model - including SMM.

version

<table>
<tr><td><b>Parameter</b></td><td>version</td></tr>
<tr><td><b>Type</b></td><td>Text string</td></tr>
<tr><td><b>Value</b></td><td>Sets Console and PAL code versions in the following way:
<table>
<tr><th>Function</th><th>Command</th></tr>
<tr><td>Set SRM Console version to X.Y-Z</td><td><code>set rom version[0] = x.y-z</code></td></tr>
<tr><td>Set OpenVMS PAL code version to X.Y-Z</td><td><code>set rom version[1] = x.y-z</code></td></tr>
<tr><td>Set Tru64 UNIX PAL code version to X.Y-Z</td><td><code>set rom version[2] = x.y-z</code></td></tr>
</table>
Example:
<pre>set rom version[0] = 7.3-1 version[1] = 1.98-104 version[2] = 1.92-105</pre>
</td></tr>
</table>

Back to Table of Contents

Virtual HP Alpha interval timer

The CHARON-AXP provides interval timer interrupts to virtual Alpha CPU(s) at frequency 100Hz (100 interrupts a second).

This is default behavior which may be changed through “clock_period” configuration parameter of virtual ISA or EISA bus, depending on emulated hardware model of virtual HP Alpha system.

Value of the parameter is interval timer period in microseconds. By default it is set to 10000. By changing it to 1000 frequency of virtual interval timer interrupts may be increased to 1000Hz (1000 interrupts per second).

Parameter	clock_period
Type	Numeric
Value	Specifies period of interval timer, in microseconds. Only two values are supported: • 10000 (which corresponds to 100Hz interval timer) • 1000 (which corresponds to 1000Hz interval timer) By default it is set to 10000.

Example for AlphaServer 400, DS, ES, GS:

```
set ISA clock_period=1000
```

Example for AlphaServer 800, 1000, 1000A, 1200, 2000, 2100, 4000, 4100:

```
set EISA clock_period=1000
```

⚠ Higher interval timer frequency creates higher load for virtual Alpha CPU which may cause degradation of overall virtual system performance.

Back to Table of Contents

Setting of a particular HP Alpha model

It is important to have the "system_name", "hw_model", "cpu_architecture" and "dsrdb[n]" (DSRDB - Dynamic System Recognition Data Block) parameters in sync. (see above for details) to configure CHARON-AXP for emulation of a particular HP Alpha model.

The following tables illustrate how to synchronize those values:

HP AlphaStation 200 - 400

set session hw_model=	set rom system_name=	set ace cpu_architecture=	set rom dsrdb[0]=
AlphaServer_400	AlphaStation 200 4/100	EV4	1156
AlphaServer_400	AlphaStation 200 4/133	EV4	1088
AlphaServer_400	AlphaStation 205 4/133	EV4	1250
AlphaServer_400	AlphaStation 255 4/133	EV4	1257
AlphaServer_400	AlphaStation 200 4/166	EV4	1087
AlphaServer_400	AlphaStation 205 4/166	EV4	1251
AlphaServer_400	AlphaStation 255 4/166	EV4	1258
AlphaServer_400	AlphaStation 400 4/166	EV4	1086
AlphaServer_400	AlphaStation 205 4/200	EV4	1252
AlphaServer_400	AlphaStation 255 4/200	EV4	1259
AlphaServer_400	AlphaStation 200 4/233	EV45	1151
AlphaServer_400	AlphaStation 205 4/233	EV45	1253
AlphaServer_400	AlphaStation 255 4/233	EV45	1260
AlphaServer_400	AlphaStation 400 4/233	EV45	1152
AlphaServer_400	AlphaStation 205 4/266	EV45	1254
AlphaServer_400	AlphaStation 255 4/266	EV45	1261
AlphaServer_400	AlphaServer 300 4/266	EV45	1593
AlphaServer_400	AlphaStation 400 4/266	EV45	1153
AlphaServer_400	AlphaStation 400 4/266	EV45	1154
AlphaServer_400	AlphaStation 200 4/300	EV45	1157
AlphaServer_400	AlphaStation 205 4/300	EV45	1255
AlphaServer_400	AlphaStation 255 4/300	EV45	1262
AlphaServer_400	AlphaStation 400 4/300	EV45	1160
AlphaServer_400	AlphaStation 205 4/333	EV45	1256
AlphaServer_400	AlphaStation 255 4/333	EV45	1263

Back to Table of Contents

HP AlphaServer 600 - 800

set session hw_model=	set rom system_name=	set ace cpu_architecture=	set rom dsrdb[0]=
AlphaServer_800	AlphaServer 600 5/333	EV56	1310
AlphaServer_800	AlphaServer 800 5/333	EV56	1310
AlphaServer_800	AlphaServer 800 5/400	EV56	1584
AlphaServer_800	AlphaStation 600A 5/500	EV56	1590
AlphaServer_800	AlphaServer 800 5/500	EV56	1585

Back to Table of Contents

HP AlphaServer 1000

set session hw_model=	set rom system_name=	set ace cpu_architecture=	set rom dsrdb[0]=
AlphaServer_1000	AlphaServer 1000 4/200	EV4	1090
AlphaServer_1000	AlphaServer 1000 4/233	EV45	1091
AlphaServer_1000	AlphaServer 1000 4/266	EV45	1264

Back to Table of Contents

HP AlphaServer 1000A

set session hw_model=	set rom system_name=	set ace cpu_architecture=	set rom dsrdb[0]=
AlphaServer_1000A	AlphaServer 1000A 4/266	EV45	1265

Back to Table of Contents

HP AlphaServer 1200 and AlphaStation 1200

set session hw_model=	set rom system_name=	set ace cpu_architecture=	set rom dsrdb[0]=
AlphaServer_1200	AlphaServer 1200 5/300	EV5	1722
AlphaServer_1200	AlphaServer 1200 5/300	EV5	1724
AlphaServer_1200	AlphaServer 1200 5/400	EV56	1726
AlphaServer_1200	AlphaServer 1200 5/400	EV56	1728
AlphaServer_1200	AlphaStation 1200 5/400	EV56	1758
AlphaServer_1200	AlphaStation 1200 5/400	EV56	1760
AlphaServer_1200	AlphaServer 1200 5/466	EV56	1730
AlphaServer_1200	AlphaServer 1200 5/466	EV56	1732
AlphaServer_1200	AlphaStation 1200 5/466	EV56	1762
AlphaServer_1200	AlphaStation 1200 5/466	EV56	1764
AlphaServer_1200	AlphaServer 1200 5/533	EV56	1734
AlphaServer_1200	AlphaServer 1200 5/533	EV56	1736
AlphaServer_1200	AlphaServer 1200 5/533	EV56	1746
AlphaServer_1200	AlphaServer 1200 5/533	EV56	1748
AlphaServer_1200	AlphaStation 1200 5/533	EV56	1766
AlphaServer_1200	AlphaStation 1200 5/533	EV56	1768
AlphaServer_1200	AlphaStation 1200 5/533	EV56	1778
AlphaServer_1200	AlphaStation 1200 5/533	EV56	1780
AlphaServer_1200	AlphaServer 1200 5/600	EV56	1738
AlphaServer_1200	AlphaServer 1200 5/600	EV56	1740
AlphaServer_1200	AlphaServer 1200 5/600	EV56	1750
AlphaServer_1200	AlphaStation 1200 5/600	EV56	1752
AlphaServer_1200	AlphaStation 1200 5/600	EV56	1770
AlphaServer_1200	AlphaStation 1200 5/600	EV56	1772
AlphaServer_1200	AlphaStation 1200 5/600	EV56	1782
AlphaServer_1200	AlphaStation 1200 5/600	EV56	1784
AlphaServer_1200	AlphaServer 1200 5/666	EV56	1742
AlphaServer_1200	AlphaServer 1200 5/666	EV56	1744
AlphaServer_1200	AlphaServer 1200 5/666	EV56	1754
AlphaServer_1200	AlphaServer 1200 5/666	EV56	1756
AlphaServer_1200	AlphaStation 1200 5/666	EV56	1774
AlphaServer_1200	AlphaStation 1200 5/666	EV56	1776
AlphaServer_1200	AlphaStation 1200 5/666	EV56	1786
AlphaServer_1200	AlphaStation 1200 5/666	EV56	1788

Back to Table of Contents

HP AlphaServer 2000

set session hw_model=	set rom system_name=	set ace cpu_architecture=	set rom dsrdb[0]=
AlphaServer_2000	AlphaServer 2000 4/200	EV4	1123
AlphaServer_2000	AlphaServer 2000 4/233	EV45	1171
AlphaServer_2000	AlphaServer 2000 4/275	EV45	1127

Back to Table of Contents

HP AlphaServer 2100

set session hw_model=	set rom system_name=	set ace cpu_architecture=	set rom dsrdb[0]=
AlphaServer_2100	AlphaServer 2100 4/200	EV4	1059
AlphaServer_2100	AlphaServer 2100 4/200	EV4	1135
AlphaServer_2100	AlphaServer 2100 4/233	EV45	1179
AlphaServer_2100	AlphaServer 2100 4/233	EV45	1187
AlphaServer_2100	AlphaServer 2100 4/275	EV45	1115
AlphaServer_2100	AlphaServer 2100 4/275	EV45	1139

Back to Table of Contents

HP AlphaServer 4000

set session hw_model=	set rom system_name=	set ace cpu_architecture=	set rom dsrdb[0]=
AlphaServer_4000	AlphaServer 4000 5/266	EV5	1409
AlphaServer_4000	AlphaServer 4000 5/266	EV5	1411
AlphaServer_4000	AlphaServer 4000 5/266	EV5	1421
AlphaServer_4000	AlphaServer 4000 5/266	EV5	1423
AlphaServer_4000	AlphaServer 4000 5/266	EV5	1433
AlphaServer_4000	AlphaServer 4000 5/266	EV5	1435
AlphaServer_4000	AlphaServer 4000 5/266	EV5	1445
AlphaServer_4000	AlphaServer 4000 5/266	EV5	1447
AlphaServer_4000	AlphaServer 4000 5/300	EV5	1413
AlphaServer_4000	AlphaServer 4000 5/300	EV5	1415
AlphaServer_4000	AlphaServer 4000 5/300	EV5	1425
AlphaServer_4000	AlphaServer 4000 5/300	EV5	1427
AlphaServer_4000	AlphaServer 4000 5/300	EV5	1437
AlphaServer_4000	AlphaServer 4000 5/300	EV5	1439
AlphaServer_4000	AlphaServer 4000 5/300	EV5	1449
AlphaServer_4000	AlphaServer 4000 5/300	EV5	1451
AlphaServer_4000	AlphaServer 4000 5/400	EV56	1417
AlphaServer_4000	AlphaServer 4000 5/400	EV56	1419
AlphaServer_4000	AlphaServer 4000 5/400	EV56	1429
AlphaServer_4000	AlphaServer 4000 5/400	EV56	1431
AlphaServer_4000	AlphaServer 4000 5/400	EV56	1441
AlphaServer_4000	AlphaServer 4000 5/400	EV56	1443
AlphaServer_4000	AlphaServer 4000 5/400	EV56	1453
AlphaServer_4000	AlphaServer 4000 5/400	EV56	1455
AlphaServer_4000	AlphaServer 4000 5/466	EV56	1634
AlphaServer_4000	AlphaServer 4000 5/466	EV56	1636
AlphaServer_4000	AlphaServer 4000 5/466	EV56	1654
AlphaServer_4000	AlphaServer 4000 5/466	EV56	1656
AlphaServer_4000	AlphaServer 4000 5/533	EV56	1638
AlphaServer_4000	AlphaServer 4000 5/533	EV56	1640
AlphaServer_4000	AlphaServer 4000 5/533	EV56	1642
AlphaServer_4000	AlphaServer 4000 5/533	EV56	1644
AlphaServer_4000	AlphaServer 4000 5/533	EV56	1658
AlphaServer_4000	AlphaServer 4000 5/533	EV56	1660
AlphaServer_4000	AlphaServer 4000 5/533	EV56	1662
AlphaServer_4000	AlphaServer 4000 5/533	EV56	1664

AlphaServer_4000	AlphaServer 4000 5/600	EV56	1646
AlphaServer_4000	AlphaServer 4000 5/600	EV56	1648
AlphaServer_4000	AlphaServer 4000 5/600	EV56	1666
AlphaServer_4000	AlphaServer 4000 5/600	EV56	1668
AlphaServer_4000	AlphaServer 4000 5/666	EV56	1650
AlphaServer_4000	AlphaServer 4000 5/666	EV56	1652
AlphaServer_4000	AlphaServer 4000 5/666	EV56	1670
AlphaServer_4000	AlphaServer 4000 5/666	EV56	1672

Back to Table of Contents

HP AlphaServer 4100

set session hw_model=	set rom system_name=	set ace cpu_architecture=	set rom dsrdb[0]=
AlphaServer_4100	AlphaServer 4100 5/266	EV5	1313
AlphaServer_4100	AlphaServer 4100 5/266	EV5	1317
AlphaServer_4100	AlphaServer 4100 5/266	EV5	1337
AlphaServer_4100	AlphaServer 4100 5/266	EV5	1341
AlphaServer_4100	AlphaServer 4100 5/266	EV5	1361
AlphaServer_4100	AlphaServer 4100 5/266	EV5	1365
AlphaServer_4100	AlphaServer 4100 5/266	EV5	1385
AlphaServer_4100	AlphaServer 4100 5/266	EV5	1389
AlphaServer_4100	AlphaServer 4100 5/300	EV5	1321
AlphaServer_4100	AlphaServer 4100 5/300	EV5	1325
AlphaServer_4100	AlphaServer 4100 5/300	EV5	1345
AlphaServer_4100	AlphaServer 4100 5/300	EV5	1349
AlphaServer_4100	AlphaServer 4100 5/300	EV5	1369
AlphaServer_4100	AlphaServer 4100 5/300	EV5	1373
AlphaServer_4100	AlphaServer 4100 5/300	EV5	1393
AlphaServer_4100	AlphaServer 4100 5/300	EV5	1397
AlphaServer_4100	AlphaServer 4100 5/400	EV56	1329
AlphaServer_4100	AlphaServer 4100 5/400	EV56	1333
AlphaServer_4100	AlphaServer 4000 5/400	EV56	1353
AlphaServer_4100	AlphaServer 4000 5/400	EV56	1357
AlphaServer_4100	AlphaServer 4000 5/400	EV56	1377
AlphaServer_4100	AlphaServer 4100 5/400	EV56	1381
AlphaServer_4100	AlphaServer 4100 5/400	EV56	1401
AlphaServer_4100	AlphaServer 4100 5/400	EV56	1405
AlphaServer_4100	AlphaServer 4100 5/466	EV56	1594
AlphaServer_4100	AlphaServer 4100 5/466	EV56	1598
AlphaServer_4100	AlphaServer 4100 5/533	EV56	1602
AlphaServer_4100	AlphaServer 4100 5/533	EV56	1606
AlphaServer_4100	AlphaServer 4100 5/533	EV56	1610
AlphaServer_4100	AlphaServer 4100 5/533	EV56	1614
AlphaServer_4100	AlphaServer 4100 5/600	EV56	1618
AlphaServer_4100	AlphaServer 4100 5/600	EV56	1622
AlphaServer_4100	AlphaServer 4100 5/666	EV56	1626
AlphaServer_4100	AlphaServer 4100 5/666	EV56	1630

Back to Table of Contents

HP AlphaServer/AlphaStation DS10 and HP AlphaServer DS10L

set session hw_model=	set rom system_name=	set ace cpu_architecture=	set rom dsrdb[0]=
AlphaServer_DS10	AlphaServer DS10 6/466	EV6	1839
AlphaServer_DS10	AlphaStation DS10 6/466	EV6	1879
AlphaServer_DS10	AlphaStation XP900 6/466	EV6	1879
AlphaServer_DS10L	AlphaServer DS10L 6/466	EV6	1961
AlphaServer_DS10L	AlphaServer DS10L 67/616	EV67	1962
AlphaServer_DS10	AlphaStation DS10 67/616	EV67	1962
AlphaServer_DS10	AlphaServer DS10 67/616	EV67	1970

Back to Table of Contents

HP AlphaServer DS15 and HP AlphaStation DS15

set session hw_model=	set rom system_name=	set ace cpu_architecture=	set rom dsrdb[0]=
AlphaServer_DS15	AlphaServer DS15 68CB/1000	EV68	2047
AlphaServer_DS15	AlphaStation DS15 68CB/1000	EV68	2048
AlphaServer_DS15	AlphaServer TS15 68CB/1000	EV68	2049

Back to Table of Contents

HP AlphaServer DS20 and HP AlphaStation DS20

set session hw_model=	set rom system_name=	set ace cpu_architecture=	set rom dsrdb[0]=
AlphaServer_DS20	AlphaServer DS20 6/500	EV6	1838
AlphaServer_DS20	AlphaServer DS20E 6/500	EV6	1840
AlphaServer_DS20	AlphaServer DS20 6/500	EV6	1920
AlphaServer_DS20	AlphaServer DS20 6/500	EV6	1921
AlphaServer_DS20	AlphaServer DS20E 67/667	EV67	1939
AlphaServer_DS20	AlphaStation DS20E 6/500	EV6	1941
AlphaServer_DS20	AlphaStation DS20E 67/667	EV57	1943
AlphaServer_DS20	AlphaServer DS20E 68A/833	EV68	1964
AlphaServer_DS20	AlphaServer DS20E 68A/833	EV68	1982
AlphaServer_DS20	AlphaServer DS20L 68A/833	EV68	2006

Back to Table of Contents

HP AlphaServer DS25 and HP AlphaStation DS25

set session hw_model=	set rom system_name=	set ace cpu_architecture=	set rom dsrdb[0]=
AlphaServer_DS25	AlphaServer DS25 68CB/1000	EV68	1994
AlphaServer_DS25	AlphaStation DS25 68CB/1000	EV68	1995

Back to Table of Contents

HP AlphaServer ES40 and AlphaStation ES40

set session hw_model=	set rom system_name=	set ace cpu_architecture=	set rom dsrdb[0]=
AlphaServer_ES40	AlphaServer ES40 6/500	EV6	1813
AlphaServer_ES40	AlphaServer ES40 6/500	EV6	1861
AlphaServer_ES40	AlphaServer ES40 6/500	EV6	1869
AlphaServer_ES40	AlphaServer ES40 6/500	EV6	1923
AlphaServer_ES40	AlphaServer ES40 6/500	EV6	1931
AlphaServer_ES40	AlphaServer ES40 6/667	EV6	1817
AlphaServer_ES40	AlphaServer ES40 6/667	EV6	1865
AlphaServer_ES40	AlphaServer ES40 6/667	EV6	1873
AlphaServer_ES40	AlphaServer ES40 6/667	EV6	1927
AlphaServer_ES40	AlphaServer ES40 6/667	EV6	1935
AlphaServer_ES40	AlphaStation ES40 67/667	EV67	1949
AlphaServer_ES40	AlphaStation ES40 67/667	EV67	1957
AlphaServer_ES40	AlphaStation ES40 68/833	EV68	1984
AlphaServer_ES40	AlphaStation ES40 68/833	EV68	1988

Back to Table of Contents

HP AlphaServer ES45

set session hw_model=	set rom system_name=	set ace cpu_architecture=	set rom dsrdb[0]=
AlphaServer_ES45	AlphaServer ES45/3B 68CB/1000	EV68	1971
AlphaServer_ES45	AlphaServer ES45/2 68CB/1000	EV68	1975
AlphaServer_ES45	AlphaServer ES45/2B 68CB/1000	EV68	1975
AlphaServer_ES45	AlphaServer ES45/1B 68CB/1000	EV68	2002
AlphaServer_ES45	AlphaServer ES45/3B 68CB/1250	EV68	2013
AlphaServer_ES45	AlphaServer ES45/2 68CB/1250	EV68	2017
AlphaServer_ES45	AlphaServer ES45/2B 68CB/1250	EV68	2017
AlphaServer_ES45	AlphaServer ES45/1B 68CB/1250	EV68	2021

Back to Table of Contents

HP AlphaServer GS80

set session hw_model=	set rom system_name=	set ace cpu_architecture=	set rom dsrdb[0]=	set rom dsrdb[1]=	set rom dsrdb[4]=
AlphaServer_GS80	AlphaServer GS80 67/728	EV67	1967		
AlphaServer_GS80	AlphaServer GS1280	EV67	2038	50	3050

Back to Table of Contents

HP AlphaServer GS160

set session hw_model=	set rom system_name=	set ace cpu_architecture=	set rom dsrdb[0]=	set rom dsrdb[1]=	set rom dsrdb[4]=
AlphaServer_GS160	AlphaServer GS160 67/728	EV67	1968		
AlphaServer_GS160	AlphaServer GS1280	EV67	2039	50	3050

Back to Table of Contents

HP AlphaServer GS320

set session hw_model=	set rom system_name=	set ace cpu_architecture=	set rom dsrdb[0]=	set rom dsrdb[1]=	set rom dsrdb[4]=
AlphaServer_GS320	AlphaServer GS320 67/728	EV67	1969		
AlphaServer_GS320	AlphaServer GS1280	EV67	2040	50	3050

Back to Table of Contents

Auto Boot

CHARON-AXP systems can be configured to boot the operating system automatically at start up.

auto_action restart

Parameter	auto_action restart
Type	Text string

Value	Determines whether CHARON-AXP boots automatically if the correct boot flags are set (and saved in the HP Alpha console files).
	Example:
	`>>>set bootdef_dev dka0`
	`>>>set auto_action restart`

Back to Table of Contents

Setting System Marketing Model (SMM)

CHARON-AXP allows to set an exact System Marketing Model (SMM) for a given model of HP Alpha, for example:

```
set rom dsrdb[0]=1090
```

Refer to this section to find allowed values of SMM per each HP Alpha model supported by CHARON-AXP.

Back to Table of Contents

Console

Table of Contents

Back to Table of Contents

General Description

CHARON-AXP offers a one- (OPA0) or two-port serial console, depending on the specified HP Alpha model.

ℹ The AlphaServer DS10, AlphaServer DS10L, AlphaServer DS15, AlphaServer 400, AlphaServer 800, AlphaServer 1000 and AlphaServer 1000A have an additional on-board serial line controller providing a serial line TTA0.

The regular console ("COM1") is already preloaded and mapped to "OPA0", so you need to specify just its mapping to the host resources ("OPA0"), for example:

```
load physical_serial_line OPA0 line="/dev/tty0"
```

When using the TTA0 console, the mapping looks like that:

```
load virtual_serial_line TTA0 port=10000
set COM2 line=TTA0
```

The first line specifies a mapping to a unique TCP/IP host port ("10000" in this example), while the second line connects this mapping (TTA0) to the on-board serial line controller having the name "COM2" in CHARON environment.

Refer to this section for details of mapping.

Back to Table of Contents

Console Parameters

CHARON-AXP console line TTA0 (COM2) has the following parameters:

ℹ All the values in the following tables are case insensitive.

communication

Parameter	communication
Type	Text string
Value	• "ascii" - for connection to terminals (default) • "binary" - for binary (packet) protocols, which are used mainly for communicating with PLCs

line

Parameter	line
Type	Identifier
Value	This parameter is used to connect a particular serial line mapping interface to the controller. Example: `set COM2 line=TTA0`

Example

```
set COM2 communication="binary"
```

Back to Table of Contents

Mapping Serial line controllers to system resources

Types of serial line mapping:

Type	**Function**
physical_serial_line	This type of mapping associates some TTY port on host system with an emulated HP Alpha serial line controller virtual "line". The TTY port can be physical hardware port or a logical TTY port.
virtual_serial_line	This type of mapping associates a network connection on the host system with an emulated HP Alpha serial line controller virtual "line"
operator_console	This type of mapping associates the current TTY console with the OPA0 console port (if CHARON-AXP does not run as service)

Example:

```
load physical_serial_line TTA0
```

Back to Table of Contents

physical_serial_line

line

<table>
<tr><td><b>Parameter</b></td><td>line</td></tr>
<tr><td><b>Type</b></td><td>Text string</td></tr>
<tr><td><b>Value</b></td><td>A defined TTY port on host system:
<table>
<tr><th>Syntax</th><th>Description</th></tr>
<tr><td>"/dev/tty&lt;N&gt;"</td><td>Virtual serial lines</td></tr>
<tr><td>"/dev/ttyS&lt;N&gt;"</td><td>Onboard serial lines</td></tr>
<tr><td>"/dev/ttyUSB&lt;N&gt;"</td><td>Modem or USB serial lines adapters</td></tr>
<tr><td>"/dev/tty<XXX&gt;"</td><td>Proprietary (depending on a driver) devices such as DIGI or MOXA cards</td></tr>
</table>
A specific account for running CHARON ("charon") does not allow usage of physical consoles "<code>/dev/tty&lt;N&gt;</code>" as CHARON consoles. If you plan to map CHARON console to "<code>/dev/tty&lt;N&gt;</code>" use only "root" account for CHARON running.</td></tr>
</table>

baud

Parameter	baud
Type	Numeric
Value	Forces the baud rate of the corresponding TTY port to a specified value.The variety of supported values depends on the underlying physical communication resource (TTY port). The most widely used values are: 300, 1200, 9600, 19200, 38400. Example: `set OPA0 baud=38400`

break_on

Parameter	break_on
Type	Text string
Value	Specifies what byte sequences received over the physical serial line will trigger a HALT command. This parameter works only for the console line. Specify the following values: "Crtl-P", "Break" or "none" ("none" disables triggering a HALT condition). Example: `set OPA0 break_on="Crtl-P"` The default value is "Break". ⚠ This parameter can be specified only for COM1 (OPA0) console

stop_on

Parameter	stop_on
Type	Text string
Value	Specifies what byte sequences received over the physical serial line will trigger a STOP condition. The STOP condition causes CHARON-AXP to exit. Specify either “F6” or “none” ("none" disables triggering a STOP condition). Example: `set OPA0 stop_on="F6"` The default value is "none". Setting "F6" triggers the STOP condition upon receipt of the "`<ESC>[17~`" sequence. Terminals usually send these sequences by pressing the **F6** button ⚠ This parameter can be specified only for COM1 (OPA0) console

log

Parameter	log
Type	Text string

Value	A string specifying a file name to store the content of the console sessions or a directory where the log files for each individual session will be stored.
	If an existing directory is specified, CHARON-AXP automatically enables creation of individual log files, one for each session using the same scheme as used for the generation of the rotating log files. If the "log" parameter is omitted, CHARON-AXP does not create a console log.
	Examples:
	`set OPA0 log="log.txt"`
	`set OPA0 log="/Charon/Logs"`
	Only existing directory can be used as a value of the "log" parameter.

Example

Mapping a console line to a host physical serial line:

```
load physical_serial_line OPA0
set OPA0 line="/dev/ttyS1"
```

Back to Table of Contents

virtual_serial_line

host

Parameter	host
Type	Text string
Value	A remote host's IP address or hostname (and optionally a remote TCP/IP port number) for the virtual serial line connection. If omitted, the virtual serial line does not initiate a connection to the remote host and will listen for incoming connection requests. Specify the value in the following form: `set OPA0 host="<host-name>[:<port-no>]"` If the "<port-no>" is not specified, the virtual serial line uses the TCP/IP port number specified by the "port" parameter (see below).

port

Parameter	port
Type	Numeric
Value	The TCP/IP port number for the virtual serial line. A virtual serial line always listens on this port for incoming connection requests. If multiple virtualized machines are running on a server, ensure the port number is unique across the platform.

break_on

Parameter	break_on
Type	Text string
Value	Specifies what byte sequences received over a virtual serial line triggers a HALT command. Specify one of the following values: "Crtl-P", "Break" or "none" to disable triggering a HALT condition. The commands are case insensitive. Example: `set OPA0 break_on="Crtl-P"` The default value is "Break". ⚠ This parameter can be specified only for COM1 (OPA0) console

stop_on

Parameter	stop_on
Type	Text string
Value	Specifies what byte sequences received over the virtual serial line will trigger a STOP condition. The STOP condition causes CHARON-AXP to exit. Specify either “F6” or “none” ("none" disables triggering a STOP condition). The commands are case insensitive. Example: `set OPA0 stop_on="F6"` The default value is "none". Setting "**F6**" triggers the STOP condition upon receipt of the "`<ESC>[17~`" sequence. ⚠ This parameter can be specified only for COM1 (OPA0) console

log

Parameter	log
Type	Text string

Value	A string specifying the filename to store the content of the console sessions or a directory where log files for each individual session will be stored.
	If an existing directory is specified, CHARON-AXP automatically enables the creation of individual log files, one for each session using the same scheme as used for the generation of the rotating log files. If the "log" parameter is omitted, CHARON-AXP does not create any console log.
	Examples:
	`set OPA0 log="log.txt"`
	`set OPA0 log="/Charon/Logs"`
	Only existing directory can be used as a value of the "log" parameter.

Example

Mapping a console line to a host TCP/IP port 10003:

```
load virtual_serial_line OPA0 port=10003
```

Notes on "virtual_serial_line" options

1. Use the combination of "port" and "host" parameters as follows to connect a 3rd party terminal emulator or similar program.

   ```
   load virtual_serial_line TTA0 host="192.168.1.1" port=10000
   ```

   In this example CHARON-AXP connects to port 10000 of a host with TCP/IP address "192.168.1.1" and at the same time it accepts connections on local port 10000.

2. It is possible to specify a port on a remote host (note that CHARON always acts as a server). The syntax is:

   ```
   load virtual_serial_line TTA0 host="192.168.1.1:20000" port=10000
   ```

   In this example CHARON-AXP accepts connection on local port 10000 and connects to remote port 20000 of a host with TCP/IP address "192.168.1.1"

Note: the examples above are mainly used for inter-CHARON communications. They are used to connect CHARON-AXP to an application that communicates to CHARON-AXP as described below.

Example of two CHARON systems connected to each other:

On host "A":

```
load virtual_serial_line TTA0 port=5500 host="B"
```

On host "B":

```
load virtual_serial_line TTA0 port=5500 host="A"
```

On these two hosts, executing CHARON-AXP, the two TTA0 lines connect to each other, thus creating a "serial" cable between the two emulated Alphas. The sequential order in which the instances of CHARON-AXP are started makes no difference.

Back to Table of Contents

operator_console

break_on, stop_on

Parameter	break_on, stop_on
Type	Text string

Value	These two parameters are hardcoded to the following values and cannot be changed:
	`stop_on="F6"`
	`break_on="Ctrl-P,F5"`

Example

```
load operator_console OPA0
```

Back to Table of Contents

"ttyY" notation specifics

Note that the "*`ttyY`*" notation can have different forms depending on the nature of the device used:

Mapping	Type	Commentary
"/dev/tty<N>" where N is from 0 to 11	Linux virtual tty	Those tty devices must be free from the Linux "getty/mgetty" and similar programs (specified in "/etc/inittab") Example: `"/dev/tty1"`
"/dev/ttyS<N>" where N is a number	Onboard serial lines	Example: `"/dev/ttyS1"`

"/dev/tty<XXX>"
where XXX is a complex letter/number notation
	Proprietary (depending on a driver) devices	Example for a first port of a MOXA card:
		`"/dev/ttyR01"`
		Example for a first port of a DIGI card:
		`"/dev/ttyaa"`

"/dev/ttyUSB<N>" where N is a number	Modem or USB serial lines adapters	Example: `"/dev/ttyUSB1"`

Back to Table of Contents

Placement of peripheral devices on PCI bus

Table of Contents

Back to Table of Contents

General Description

Each peripheral device of CHARON-AXP connects to CHARON-AXP emulated PCI bus with the following configuration parameters:

bus

<table>
<tr><td><b>Parameter</b></td><td>bus</td></tr>
<tr><td><b>Type</b></td><td>Text string</td></tr>
<tr><td><b>Value</b></td><td>Value formats:
<table>
<tr><th>Models</th><th>Format</th></tr>
<tr><td>AlphaServer 400-4100, DS, ES</td><td>"pci_&lt;X&gt;"</td></tr>
<tr><td>AlphaServer GS</td><td>"qbb_&lt;X>_pca_<Y>_pci_&lt;Z&gt;"</td></tr>
</table>
<p>When specified, the bus configuration parameter tells the CHARON-AXP software the virtual PCI bus to which virtual HP Alpha system shall connect a certain virtual PCI adapter.</p>
<p>By default the bus configuration parameter is not specified.</p>
<p>If the bus configuration parameter is not specified, CHARON-AXP software connects the virtual PCI adapter to the first available virtual PCI bus.</p>
<p><u>Example (AlphaServer ES40)</u>:</p>
<pre>load KZPBA PKA bus=pci_1</pre>
<p><u>Example (AlphaServer GS80)</u>:</p>
<pre>load KZPBA PKA bus=qbb_1_pca_1_pci_0</pre>
</td></tr>
</table>

device

<table>
<tr><td><b>Parameter</b></td><td>device</td></tr>
<tr><td><b>Type</b></td><td>Numeric</td></tr>
<tr><td><b>Value</b></td><td><p>When specified, the device configuration parameter specifies position of a virtual PCI adapter on virtual PCI bus.</p>
<p>By default the device configuration parameter is not specified.</p>
<p>If the device configuration parameter is not specified, the CHARON software connects the virtual PCI adapter at the first available position of the virtual PCI bus.</p>
<p><u>Example</u>:</p>
<pre>load KZPBA PKA device=2</pre>
</td></tr>
</table>

function

Parameter	function
Type	Numeric
Value	When specified, the function configuration parameter specifies position of a virtual PCI adapter on virtual PCI bus. By default the function configuration parameter is not specified. If the function configuration parameter is not specified, the CHARON software connects the virtual PCI adapter at the first available position of the virtual PCI bus. Example: `load KZPBA PKA function=0`

irq_bus

Parameter	irq_bus
Type	Text string
Value	When specified, the "irq_bus" configuration parameter specifies virtual bus routing interrupt requests from virtual PCI adapter to CHARON-AXP virtual Alpha CPUs. By default the "irq_bus" configuration parameter is not specified. The "irq_bus" configuration parameter must be set to “isa” for AlphaServer 400. For HP Alpha systems other than AlphaServer 400 the "irq_bus" configuration parameter must be left as is (i.e. not specified). Example: `load KZPBA PKA irq_bus=isa`

irq

Parameter	irq
Type	Numeric
Value	When specified, the "irq" configuration parameter assigns interrupt request to the virtual PCI adapter in HP Alpha system. By default the irq configuration parameter is not specified. If the irq configuration parameter is not specified, the CHARON-AXP software uses the correct values depending on the selected PCI position of a virtual PCI adapter. Example: `load KZPBA PKA irq=24`

Note

Typically all or some of those parameters are specified on loading of some PCI controller in the following way:

```
load KZPBA PKA bus=pci_1 device=1 function=0 irq_bus=isa irq=24
```

Back to Table of Contents

Available PCI slots per each HP Alpha model emulated by CHARON-AXP

The tables below specify a map of preloaded devices and available slots for each HP Alpha models emulated by CHARON-AXP.

AlphaServer 400 (3 PCI slots)

In addition to 3 PCI vacant slots there are 2 PCI positions occupied by on-board devices. All 5 PCI positions are listed in the following table in the order in which Alpha SRM console enumerates them.

Slot	pci_<N>	device	function	irq	Description	Preloaded Name
PCI0 (bus=pci_0)						
-	0	6	0	11	NCR 53C810 PCI SCSI Adapter	PKA
-	0	7	0	-	Intel i82378 PCI ISA Bridge (SATURN)	
0	0	11	0	10	<option>	
1	0	12	0	15	<option>	
2	0	13	0	9	<option>	

The IRQ stands for ISA IRQ Number because all interrupts are routed through the Intel i82378 PCI ISA Bridge (SATURN) resident cascade of Intel i8259 interrupt controllers.

So far the CHARON-AXP emulators do not support virtual NCR 53C810 PCI SCSI adapter. Instead, virtual QLOGIC ISP1040B PCI SCSI adapter is used.

ℹ No support for Multi-Function PCI devices in AlphaServer 400.

Example: Loading DE435 into slot 0

```
load DE435/dec21x4x EWA bus=pci_0 device=11 function=0 irq_bus=isa
```

ℹ The "irq_bus=isa" setting is specific to AlphaServer 400 only.

Back to Table of Contents

AlphaServer 800 (4 PCI slots)

In addition to 4 PCI vacant slots there are 3 PCI positions occupied by on-board devices. All 7 PCI positions are listed in the following table in the order in which Alpha SRM console enumerates them.

Slot	pci_<N>	device	function	irq	Description	Preloaded Name
PCI0 (bus=pci_0)						
-	0	5	0	0	QLOGIC ISP1020 PCI SCSI Adapter	PKA
-	0	6	0	0	S3 Trio32/64 Display Adapter	
-	0	7	0	-	Intel i82375 PCI EISA Bridge (MERCURY)	
0	0	11	0	1	<option>	
			1	2	<option>, function 1	
			2	17	<option>, function 2	
			3	18	<option>, function 3	
1	0	12	0	3	<option>	
			1	4	<option>, function 1	
			2	19	<option>, function 2	
			3	20	<option>, function 3	
2	0	13	0	5	<option>	
			1	6	<option>, function 1	
			2	21	<option>, function 2	
			3	22	<option>, function 3	
3	0	14	0	7	<option>	
			1	8	<option>, function 1	
			2	23	<option>, function 2	
			3	24	<option>, function 3	

The IRQ stands for input line of ASIC interrupt controllers. It has nothing to do with "EISA" style interrupts. So far, the CHARON-AXP emulators do not emulate S3 Trio32/64 Display Adapter. So position of the device 6, function 0 on the PCI 0 remains empty.

Example 1: Loading DE500BA into slot 0

```
load DE500BA/dec21x4x EWA bus=pci_0 device=11 function=0
```

Example 2: Loading multiple DE500BA's into slot 3, populating all 4 functions (gives 4 Ethernet ports)

```
load DE500BA/dec21x4x EWA bus=pci_0 device=14 function=0
load DE500BA/dec21x4x EWB bus=pci_0 device=14 function=1
load DE500BA/dec21x4x EWC bus=pci_0 device=14 function=2
load DE500BA/dec21x4x EWD bus=pci_0 device=14 function=3
```

Example 3: Loading mixture of KZPBA and DE500BA into slot 1, populating 2 functions out of 4

```
load KZPBA PKB bus=pci_0 device=12 function=0
load DE500BA/dec21x4x EWA bus=pci_0 device=12 function=1
```

ℹ In the above example device name is PKB as there is a built-in PK-like PCI SCSI Adapter located “closer” to CPU and therefore assigned name PKA.

Back to Table of Contents

AlphaServer 1000 (3 PCI slots)

In addition to 3 PCI vacant slots there are 2 PCI positions occupied by on-board devices. All 5 PCI positions are listed in the following table in the order in which Alpha SRM console enumerates them.

Slot	pci_<N>	device	function	irq	Description	Preloaded Name
PCI0 (bus=pci_0)						
-	0	6	0	12	NCR 53C810 PCI SCSI Adapter	PKA
-	0	7	0	-	Intel i82375 PCI EISA Bridge (MERCURY)	
0	0	11	0	0	<option>	
			1	1	<option>, function 1	
			2	2	<option>, function 2	
			3	3	<option>, function 3	
1	0	12	0	4	<option>	
			1	5	<option>, function 1	
			2	6	<option>, function 2	
			3	7	<option>, function 3	
2	0	13	0	8	<option>	
			1	9	<option>, function 1	
			2	10	<option>, function 2	
			3	11	<option>, function 3	

The IRQ stands for input line of ASIC interrupt controllers. It has nothing to do with "EISA" style interrupts. So far, the CHARON-AXP emulators do not emulate NCR 53C810 PCI SCSI adapter. Instead, emulation of QLOGIC ISP1040B PCI SCSI adapter is used.

Example 1: Loading DE500BA into slot 0

```
load DE500BA/dec21x4x EWA bus=pci_0 device=11 function=0
```

Example 2: Loading multiple DE500BA's into slot 0, populating all 4 functions (gives 4 Ethernet ports)

```
load DE500BA/dec21x4x EWA bus=pci_0 device=11 function=0
load DE500BA/dec21x4x EWB bus=pci_0 device=11 function=1
load DE500BA/dec21x4x EWC bus=pci_0 device=11 function=2
load DE500BA/dec21x4x EWD bus=pci_0 device=11 function=3
```

Example 3: Loading mixture of KZPBA and DE500BA into slot 2, populating 2 functions out of 4

```
load KZPBA PKB bus=pci_0 device=13 function=0
load DE500BA/dec21x4x EWA bus=pci_0 device=13 function=1
```

ℹ In the above example device name is PKB as there is a built-in PK-like PCI SCSI Adapter located “closer” to CPU and therefore assigned name PKA.

Back to Table of Contents

AlphaServer 1000A (7 PCI slots)

In addition to 7 PCI vacant slots there are 3 PCI positions occupied by on-board devices. All 10 PCI positions are listed in the following table in the order in which Alpha SRM console enumerates them.

Slot	pci_<N>	device	function	irq	Description	Preloaded Name
PCI0 (bus=pci_0)						
-	0	6	0	-	Intel i82375 PCI EISA Bridge (MERCURY)	
-	0	7	0	-	DECchip 21050 PCI-to-PCI Bridge)	
0	0	11	0	1	<option>	
			1	2	<option>, function 1	
			2	17	<option>, function 2	
			3	18	<option>, function 3	
1	0	12	0	2	<option>	
			1	3	<option>, function 1	
			2	19	<option>, function 2	
			3	20	<option>, function 3	
2	0	13	0	3	<option>	
			1	4	<option>, function 1	
			2	21	<option>, function 2	
			3	22	<option>, function 3	
PCI1 (bus=pci_1)						
-	1	0	0	0	NCR 53C810 PCI SCSI Adapter	PKA
3	1	1	0	7	<option>	
			1	8	<option>, function 1	
			2	23	<option>, function 2	
			3	24	<option>, function 3	
4	1	2	0	9	<option>	
			1	10	<option>, function 1	
			2	25	<option>, function 2	
			3	26	<option>, function 3	
5	1	3	0	11	<option>	
			1	12	<option>, function 1	
			2	27	<option>, function 2	
			3	28	<option>, function 3	
6	1	4	0	13	<option>	
			1	14	<option>, function 1	
			2	29	<option>, function 2	
			3	30	<option>, function 3	

The IRQ stands for input line of ASIC interrupt controllers. It has nothing to do with "EISA" style interrupts. So far, the CHARON-AXP emulators do not emulate NCR 53C810 PCI SCSI adapter. Instead, emulation of QLOGIC ISP1040B PCI SCSI adapter is used.

Example 1: Loading DE500BA into slot 0

```
load DE500BA/dec21x4x EWA bus=pci_0 device=11 function=0
```

Example 2: Loading multiple DE500BA’s into slot 0, populating all 4 functions (gives 4 Ethernet ports)

```
load DE500BA/dec21x4x EWA bus=pci_0 device=11 function=0
load DE500BA/dec21x4x EWB bus=pci_0 device=11 function=1
load DE500BA/dec21x4x EWC bus=pci_0 device=11 function=2
load DE500BA/dec21x4x EWD bus=pci_0 device=11 function=3
```

Example 3: Loading mixture of KZPBA and DE500BA into slot 3, populating 2 functions out of 4

```
load KZPBA PKB bus=pci_1 device=1 function=0
load DE500BA/dec21x4x EWA bus=pci_1 device=1 function=1
```

ℹ In the above example device name is PKB as there is a built-in PK-like PCI SCSI Adapter located “closer” to CPU and therefore assigned name PKA.

Back to Table of Contents

AlphaServer 1200 (6 PCI slots)

In addition to 6 PCI vacant slots there are 2 PCI positions occupied by on-board devices. All 8 PCI positions are listed in the following table in the order in which Alpha SRM console enumerates them.

Slot	pci_<N>	device	function	irq	Description	Preloaded Name
PCI1 (bus=pci_1)						
-	1	1	0	4	NCR 53C810 PCI SCSI Adapter	PKA
0	1	2	0	8	<option>	
			1	9	<option>, function 1	
			2	10	<option>, function 2	
			3	11	<option>, function 3	
1	1	3	0	12	<option>	
			1	13	<option>, function 1	
			2	14	<option>, function 2	
			3	16	<option>, function 3	
2	1	4	0	16	<option>	
			1	17	<option>, function 1	
			2	18	<option>, function 2	
			3	19	<option>, function 3	
PCI0 (bus=pci_0)						
-	0	1	0	-	Intel i82375 PCI EISA Bridge (MERCURY)	
4	0	2	0	8	<option>	
			1	9	<option>, function 1	
			2	10	<option>, function 2	
			3	11	<option>, function 3	
5	0	3	0	12	<option>	
			1	13	<option>, function 1	
			2	14	<option>, function 2	
			3	15	<option>, function 3	
6	0	4	0	16	<option>	
			1	17	<option>, function 1	
			2	18	<option>, function 2	
			3	19	<option>, function 3	

So far, the CHARON-AXP emulators do not emulate NCR 53C810 PCI SCSI adapter. Instead, emulation of QLOGIC ISP1040B PCI SCSI adapter is used.

Example 1: Loading DE500BA into slot 4

```
load DE500BA/dec21x4x EWA bus=pci_0 device=2 function=0
```

Example 2: Loading multiple DE500BA’s into slot 4, populating all 4 functions (gives 4 Ethernet ports)

```
load DE500BA/dec21x4x EWA bus=pci_0 device=2 function=0
load DE500BA/dec21x4x EWB bus=pci_0 device=2 function=1
load DE500BA/dec21x4x EWC bus=pci_0 device=2 function=2
load DE500BA/dec21x4x EWD bus=pci_0 device=2 function=3
```

Example 3: Loading mixture of KZPBA and DE500BA into slot 1, populating 2 functions out of 4

```
load KZPBA PKB bus=pci_1 device=2 function=0
load DE500BA/dec21x4x EWA bus=pci_1 device=2 function=1
```

ℹ In the above example device name is PKB as there is a built-in PK-like PCI SCSI Adapter located “closer” to CPU and therefore assigned name PKA.

Back to Table of Contents

AlphaServer 2000 (3 PCI slots)

In addition to 3 PCI vacant slots there are 3 PCI positions occupied by on-board devices. All 6 PCI positions are listed in the following table in the order in which Alpha SRM console enumerates them.

Slot	pci_<N>	device	function	irq	Description	Preloaded Name
PCI0 (bus=pci_0)						
-	0	0	0	2	DEC TULIP PCI Ethernet adapter	EWA
-	0	1	0	1	NCR 53C810 PCI SCSI Adapter	PKA
-	0	2	0	-	Intel i82375 PCI EISA Bridge (MERCURY)	
0	0	6	0	0	<option>	
			1	24	<option>, function 1	
			2	26	<option>, function 2	
			3	29	<option>, function 3	
1	0	7	0	4	<option>	
			1	25	<option>, function 1	
			2	27	<option>, function 2	
			3	30	<option>, function 3	
2	0	8	0	5	<option>	
			1	20	<option>, function 1	
			2	28	<option>, function 2	
			3	31	<option>, function 3	

The IRQ stands for input line of T2 resident cascade of Intel i8259 interrupt controllers. It has nothing to do with “EISA” style interrupts.

So far the CHARON-AXP emulators do not support virtual NCR 53C810 PCI SCSI adapter. Instead, virtual QLOGIC ISP1040B PCI SCSI adapter is used.

Example 1: Loading DE500BA into slot 0

```
load DE500BA/dec21x4x EWA bus=pci_0 device=6 function=0
```

Example 2: Loading multiple DE500BA’s into slot 0, populating all 4 functions (gives 4 Ethernet ports)

```
load DE500BA/dec21x4x EWA bus=pci_0 device=6 function=0
load DE500BA/dec21x4x EWB bus=pci_0 device=6 function=1
load DE500BA/dec21x4x EWC bus=pci_0 device=6 function=2
load DE500BA/dec21x4x EWD bus=pci_0 device=6 function=3
```

Example 3: Loading mixture of KZPBA and DE500BA into slot 1, populating 2 functions out of 4

```
load KZPBA PKB bus=pci_0 device=7 function=0
load DE500BA/dec21x4x EWA bus=pci_0 device=7 function=1
```

ℹ In the above example device name is PKB as there is a built-in PK-like PCI SCSI Adapter located “closer” to CPU and therefore assigned name PKA, and device name is EWB as there is a built-in EW-like PCI Ethernet Adapter located “closer” to CPU and therefore assigned name EWA.

Back to Table of Contents

AlphaServer 2100 (3 PCI slots)

In addition to 3 PCI vacant slots there are 3 PCI positions occupied by on-board devices. All 6 PCI positions are listed in the following table in the order in which Alpha SRM console enumerates them.

Slot	pci_<N>	device	function	irq	Description	Preloaded Name
PCI0 (bus=pci_0)						
-	0	0	0	2	DEC TULIP PCI Ethernet adapter	EWA
-	0	1	0	1	NCR 53C810 PCI SCSI Adapter	PKA
-	0	2	0	-	Intel i82375 PCI EISA Bridge (MERCURY)	
0	0	6	0	0	<option>	
			1	24	<option>, function 1	
			2	26	<option>, function 2	
			3	29	<option>, function 3	
1	0	7	0	4	<option>	
			1	25	<option>, function 1	
			2	27	<option>, function 2	
			3	30	<option>, function 3	
2	0	8	0	5	<option>	
			1	20	<option>, function 1	
			2	28	<option>, function 2	
			3	31	<option>, function 3	

The IRQ stands for input line of T2 resident cascade of Intel i8259 interrupt controllers. It has nothing to do with “EISA” style interrupts.

So far the CHARON-AXP emulators do not support virtual NCR 53C810 PCI SCSI adapter. Instead, virtual QLOGIC ISP1040B PCI SCSI adapter is used.

Example 1: Loading DE500BA into slot 0

```
load DE500BA/dec21x4x EWA bus=pci_0 device=6 function=0
```

Example 2: Loading multiple DE500BA’s into slot 0, populating all 4 functions (gives 4 Ethernet ports)

```
load DE500BA/dec21x4x EWA bus=pci_0 device=6 function=0
load DE500BA/dec21x4x EWB bus=pci_0 device=6 function=1
load DE500BA/dec21x4x EWC bus=pci_0 device=6 function=2
load DE500BA/dec21x4x EWD bus=pci_0 device=6 function=3
```

Example 3: Loading mixture of KZPBA and DE500BA into slot 1, populating 2 functions out of 4

```
load KZPBA PKB bus=pci_0 device=7 function=0
load DE500BA/dec21x4x EWA bus=pci_0 device=7 function=1
```

ℹ In the above example device name is PKB as there is a built-in PK-like PCI SCSI Adapter located “closer” to CPU and therefore assigned name PKA, and device name is EWB as there is a built-in EW-like PCI Ethernet Adapter located “closer” to CPU and therefore assigned name EWA.

Back to Table of Contents

AlphaServer 4000 (16 PCI slots)

In addition to 16 PCI vacant slots there are 2 PCI positions occupied by on-board devices. All 18 PCI positions are listed in the following table in the order in which Alpha SRM console enumerates them.

Slot	pci_<N>	device	function	irq	Description	Preloaded Name
PCI1 (bus=pci_1)						
-	1	1	0	4	NCR 53C810 PCI SCSI Adapter	PKA
1	1	2	0	8	<option>	
			1	9	<option>, function 1	
			2	10	<option>, function 2	
			3	11	<option>, function 3	
2	1	3	0	12	<option>	
			1	13	<option>, function 1	
			2	14	<option>, function 2	
			3	15	<option>, function 3	
3	1	4	0	16	<option>	
			1	17	<option>, function 1	
			2	18	<option>, function 2	
			3	19	<option>, function 3	
4	1	5	0	20	<option>	
			1	21	<option>, function 1	
			2	22	<option>, function 2	
			3	23	<option>, function 3	
PCI0 (bus=pci_0)						
-	0	1	0	-	Intel i82375 PCI EISA Bridge (MERCURY)	
5	0	2	0	8	<option>	
			1	9	<option>, function 1	
			2	10	<option>, function 2	
			3	11	<option>, function 3	
6	0	3	0	12	<option>	
			1	13	<option>, function 1	
			2	14	<option>, function 2	
			3	15	<option>, function 3	
7	0	4	0	16	<option>	
			1	17	<option>, function 1	
			2	18	<option>, function 2	
			3	19	<option>, function 3	
8	0	5	0	20	<option>	
			1	21	<option>, function 1	

			2	22	<option>, function 2	
			3	23	<option>, function 3	
PCI3 (bus=pci_3)						
9	3	2	0	8	<option>	
			1	9	<option>, function 1	
			2	10	<option>, function 2	
			3	11	<option>, function 3	
10	3	3	0	12	<option>	
			1	13	<option>, function 1	
			2	14	<option>, function 2	
			3	15	<option>, function 3	
11	3	4	0	16	<option>	
			1	17	<option>, function 1	
			2	18	<option>, function 2	
			3	19	<option>, function 3	
12	3	5	0	20	<option>	
			1	21	<option>, function 1	
			2	22	<option>, function 2	
			3	23	<option>, function 3	
PCI2 (bus=pci_2)						
13	2	2	0	8	<option>	
			1	9	<option>, function 1	
			2	10	<option>, function 2	
			3	11	<option>, function 3	
14	2	3	0	12	<option>	
			1	13	<option>, function 1	
			2	14	<option>, function 2	
			3	15	<option>, function 3	
15	2	4	0	16	<option>	
			1	17	<option>, function 1	
			2	18	<option>, function 2	
			3	19	<option>, function 3	
16	2	5	0	20	<option>	
			1	21	<option>, function 1	
			2	22	<option>, function 2	
			3	23	<option>, function 3	

So far the CHARON-AXP emulators do not support virtual NCR 53C810 PCI SCSI adapter. Instead, virtual QLOGIC ISP1040B PCI SCSI adapter is used.

Example 1: Loading DE500BA into slot 4

```
load DE500BA/dec21x4x EWA bus=pci_1 device=5 function=0
```

Example 2: Loading multiple DE500BA’s into slot 4, populating all 4 functions (gives 4 Ethernet ports)

```
load DE500BA/dec21x4x EWA bus=pci_1 device=5 function=0
load DE500BA/dec21x4x EWB bus=pci_1 device=5 function=1
load DE500BA/dec21x4x EWC bus=pci_1 device=5 function=2
load DE500BA/dec21x4x EWD bus=pci_1 device=5 function=3
```

Example 3: Loading mixture of KZPBA and DE500BA into slot 1, populating 2 functions out of 4

```
load KZPBA PKB bus=pci_1 device=2 function=0
load DE500BA/dec21x4x EWA bus=pci_1 device=2 function=1
```

ℹ In the above example device name is PKB as there is a built-in PK-like PCI SCSI Adapter located “closer” to CPU and therefore assigned name PKA.

Back to Table of Contents

AlphaServer 4100 (8 PCI slots)

In addition to 8 PCI vacant slots there are 2 PCI positions occupied by on-board devices. All 10 PCI positions are listed in the following table in the order in which Alpha SRM console enumerates them.

Slot	pci_<N>	device	function	irq	Description	Preloaded Name
PCI1 (bus=pci_1)						
-	1	1	0	4	NCR 53C810 PCI SCSI Adapter	PKA
1	1	2	0	8	<option>	
			1	9	<option>, function 1	
			2	10	<option>, function 2	
			3	11	<option>, function 3	
2	1	3	0	12	<option>	
			1	13	<option>, function 1	
			2	14	<option>, function 2	
			3	15	<option>, function 3	
3	1	4	0	16	<option>	
			1	17	<option>, function 1	
			2	18	<option>, function 2	
			3	19	<option>, function 3	
4	1	5	0	20	<option>	
			1	21	<option>, function 1	
			2	22	<option>, function 2	
			3	23	<option>, function 3	
PCI0 (bus=pci_0)						
-	0	1	0	-	Intel i82375 PCI EISA Bridge (MERCURY)	
5	0	2	0	8	<option>	
			1	9	<option>, function 1	
			2	10	<option>, function 2	
			3	11	<option>, function 3	
6	0	3	0	12	<option>	
			1	13	<option>, function 1	
			2	14	<option>, function 2	
			3	15	<option>, function 3	
7	0	4	0	16	<option>	
			1	17	<option>, function 1	
			2	18	<option>, function 2	
			3	19	<option>, function 3	
8	0	5	0	20	<option>	
			1	21	<option>, function 1	

	2	22	<option>, function 2	
	3	23	<option>, function 3	

So far the CHARON-AXP emulators do not support virtual NCR 53C810 PCI SCSI adapter. Instead, virtual QLOGIC ISP1040B PCI SCSI adapter is used.

Example 1: Loading DE500BA into slot 4

```
load DE500BA/dec21x4x EWA bus=pci_1 device=5 function=0
```

Example 2: Loading multiple DE500BA’s into slot 4, populating all 4 functions (gives 4 Ethernet ports)

```
load DE500BA/dec21x4x EWA bus=pci_1 device=5 function=0
load DE500BA/dec21x4x EWB bus=pci_1 device=5 function=1
load DE500BA/dec21x4x EWC bus=pci_1 device=5 function=2
load DE500BA/dec21x4x EWD bus=pci_1 device=5 function=3
```

Example 3: Loading mixture of KZPBA and DE500BA into slot 1, populating 2 functions out of 4

```
load KZPBA PKB bus=pci_1 device=2 function=0
load DE500BA/dec21x4x EWA bus=pci_1 device=2 function=1
```

ℹ In the above example device name is PKB as there is a built-in PK-like PCI SCSI Adapter located “closer” to CPU and therefore assigned name PKA.

Back to Table of Contents

AlphaServer DS10 (4 PCI slots)

In addition to 4 PCI vacant slots there are 5 PCI positions occupied by on-board devices. All 9 PCI positions are listed in the following table in the order in which Alpha SRM console enumerates them.

Slot	pci_<N>	device	function	irq	Description	Preloaded Name
PCI1 (bus=pci_1)						
-	0	7	0	-	ALi M1543C PCI ISA bridge	
-	0	9	0	29	DECchip 21143 PCI Ethernet Adapter	EWA
-	0	11	0	30	DECchip 21143 PCI Ethernet Adapter	EWB
-	0	13	0	-	ALi M1543C PCI IDE/ATAPI controller	DQA, DQB
1	0	14	0	35	<option>	
			1	34	<option>, function 1	
			2	33	<option>, function 2	
			3	32	<option>, function 3	
2	0	15	0	39	<option>	
			1	38	<option>, function 1	
			2	37	<option>, function 2	
			3	36	<option>, function 3	
3	0	16	0	43	<option>	
			1	42	<option>, function 1	
			2	41	<option>, function 2	
			3	40	<option>, function 3	
4	0	17	0	47	<option>	
			1	46	<option>, function 1	
			2	45	<option>, function 2	
			3	44	<option>, function 3	
-	0	19	0	11	ALi M1543C PCI USB adapter	

Example 1: Loading DE500BA into slot 1

```
load DE500BA/dec21x4x EWA bus=pci_0 device=14 function=0
```

Example 2: Loading multiple DE500BA’s into slot 1, populating all 4 functions (gives 4 Ethernet ports)

```
load DE500BA/dec21x4x EWA bus=pci_0 device=14 function=0
load DE500BA/dec21x4x EWB bus=pci_0 device=14 function=1
load DE500BA/dec21x4x EWC bus=pci_0 device=14 function=2
load DE500BA/dec21x4x EWD bus=pci_0 device=14 function=3
```

Example 3: Loading mixture of KZPBA and DE500BA into slot 1, populating 2 functions out of 4

```
load KZPBA PKB bus=pci_0 device=14 function=0
load DE500BA/dec21x4x EWC bus=pci_0 device=14 function=1
```

ℹ In the above example device name is EWC as there are two built-in EW-like PCI Ethernet Adapters located “closer” to CPU and therefore assigned names EWA and EWB.

Back to Table of Content

AlphaServer DS10L (1 PCI slot)

In addition to 1 PCI vacant slots there are 5 PCI positions occupied by on-board devices. All 6 PCI positions are listed in the following table in the order in which Alpha SRM console enumerates them.

Slot	pci_<N>	device	function	irq	Description	Preloaded Name
PCI1 (bus=pci_1)						
-	0	7	0	-	ALi M1543C PCI ISA bridge	
-	0	9	0	29	DECchip 21143 PCI Ethernet Adapter	EWA
-	0	11	0	30	DECchip 21143 PCI Ethernet Adapter	EWB
-	0	13	0	-	ALi M1543C PCI IDE/ATAPI controller	DQA, DQB
1	0	17	0	47	<option>	
			1	46	<option>, function 1	
			2	45	<option>, function 2	
			3	44	<option>, function 3	
-	0	19	0	11	ALi M1543C PCI USB adapter	

Example 1: Loading DE500BA into slot 1

```
load DE500BA/dec21x4x EWA bus=pci_0 device=17 function=0
```

Example 2: Loading multiple DE500BA's into slot 1, populating all 4 functions (gives 4 Ethernet ports)

```
load DE500BA/dec21x4x EWA bus=pci_0 device=17 function=0
load DE500BA/dec21x4x EWB bus=pci_0 device=17 function=1
load DE500BA/dec21x4x EWC bus=pci_0 device=17 function=2
load DE500BA/dec21x4x EWD bus=pci_0 device=17 function=3
```

Example 3: Loading mixture of KZPBA and DE500BA into slot 1, populating 2 functions out of 4

```
load KZPBA PKB bus=pci_0 device=17 function=0
load DE500BA/dec21x4x EWC bus=pci_0 device=17 function=1
```

ℹ In the above example device name is EWC as there are two built-in EW-like PCI Ethernet Adapters located "closer" to CPU and therefore assigned names EWA and EWB.

Back to Table of Content

AlphaServer DS15 (4 PCI slots)

In addition to 4 PCI vacant slots there are 7 PCI positions occupied by on-board devices. All 11 PCI positions are listed in the following table in the order in which Alpha SRM console enumerates them.

Slot	pci_<N>	device	function	irq	Description	Preloaded Name
PCI0 (bus=pci_0)						
-	0	7	0	-	ALi M1543C PCI ISA bridge	
-	0	8	0	-	Adaptec AIC-7899 (channel 0)	PKA
			1	-	Adaptec AIC-7899 (channel 1)	PKB
-	0	9	0	-	Intel i82559 PCI Ethernet Adapter	EIA (EWA)
-	0	10	0	-	Intel i82559 PCI Ethernet Adapter	EIB (EWB)
-	0	13	0	-	ALi M1543C PCI IDE/ATAPI controller	DQA, DQB
-	0	19	0	-	ALi M1543C PCI USB adapter	
PCI2 (bus=pci_2)						
1	2	7	0	40	<option>	
			1	41	<option>, function 1	
			2	42	<option>, function 2	
			3	43	<option>, function 3	
2	2	8	0	36	<option>	
			1	37	<option>, function 1	
			2	38	<option>, function 2	
			3	39	<option>, function 3	
3	2	9	0	24	<option>	
			1	25	<option>, function 1	
			2	26	<option>, function 2	
			3	27	<option>, function 3	
4	2	10	0	20	<option>	
			1	21	<option>, function 1	
			2	22	<option>, function 2	
			3	23	<option>, function 3	

The IRQ stands for bit position in DRIR of TITAN chip. It has nothing to do with “ISA” style interrupts which are routed to IRQ 55 (including ALi M1543C PCI IDE/ATAPI controller).

So far the CHARON-AXP emulators do not emulate Adaptec AIC-7899. Instead, emulation of QLOGIC ISP1040B is used.

So far the CHARON-AXP emulators do not emulate Intel i82559. Instead, emulation of DECchip 21143 is used.

So far the CHARON-AXP emulators do not emulate ALi M1543C PCI USB adapter. So position of the device 19, function 0 on the PCI 0 remains empty.

Example 1: Loading DE500BA into slot 1

```
load DE500BA/dec21x4x EWA bus=pci_2 device=7 function=0
```

Example 2: Loading multiple DE500BA’s into slot 2, populating all 4 functions (gives 4 Ethernet ports)

```
load DE500BA/dec21x4x EWA bus=pci_2 device=8 function=0
load DE500BA/dec21x4x EWB bus=pci_2 device=8 function=1
load DE500BA/dec21x4x EWC bus=pci_2 device=8 function=2
load DE500BA/dec21x4x EWD bus=pci_2 device=8 function=3
```

Example 3: Loading mixture of KZPBA and DE500BA into slot 3, populating 2 functions out of 4

```
load KZPBA PKC bus=pci_2 device=9 function=0
load DE500BA/dec21x4x EWA bus=pci_2 device=9 function=1
```

ℹ In the above example device name is PKC as there are two built-in PK-like PCI SCSI Adapters located “closer” to CPU and therefore assigned names PKA and PKB.

Back to Table of Content

AlphaServer DS20 (6 PCI slots)

In addition to 6 PCI vacant slots there are 5 PCI positions occupied by on-board devices. All 11 PCI positions are listed in the following table in the order in which Alpha SRM console enumerates them.

Slot	pci_<N>	device	function	irq	Description	Preloaded Name
PCI1 (bus=pci_1)						
4	1	7	0	47	<option>	
			1	46	<option>, function 1	
			2	45	<option>, function 2	
			3	44	<option>, function 3	
5	1	8	0	43	<option>	
			1	42	<option>, function 1	
			2	41	<option>, function 2	
			3	49	<option>, function 3	
6	1	9	0	39	<option>	
			1	38	<option>, function 1	
			2	37	<option>, function 2	
			3	36	<option>, function 3	
PCI0 (bus=pci_0)						
-	0	5	0	-	ALi M1543C PCI ISA bridge	
-	0	6	0	19	Adaptec AIC-7895 (channel 0)	PKA
			1	18	Adaptec AIC-7895 (channel 1)	PKB
-	0	15	0	-	ALi M1543C PCI IDE/ATAPI controller	DQA, DQB
-	0	19	0	-	ALi M1543C PCI USB adapter	
1	0	7	0	31	<option>	
			1	30	<option>, function 1	
			2	29	<option>, function 2	
			3	28	<option>, function 3	
2	0	8	0	27	<option>	
			1	26	<option>, function 1	
			2	25	<option>, function 2	
			3	24	<option>, function 3	
3	0	9	0	23	<option>	
			1	22	<option>, function 1	
			2	21	<option>, function 2	
			3	20	<option>, function 3	

The IRQ stands for bit position in DRIR of Tsunami/Typhoon Chip. It has nothing to do with “ISA” style interrupts which are routed to IRQ 55 (including ALi M1543C PCI IDE/ATAPI controller).

Unless SCSI option is plugged into PCI slot 4, 5, or 6, the onboard SCSI controllers appear as PKA (pka7.0.0.6.0) and PKB (pkb7.0.0.106.0) respectively.

So far the CHARON-AXP emulators do not support virtual Adaptec AIC-7895 PCI SCSI adapter. Instead, virtual QLOGIC ISP1040B PCI SCSI adapter is used.

So far the CHARON-AXP emulators do not support virtual ALi M1543C PCI USB adapter. So position of the device 19, function 0 on the PCI 0 remains empty

Example 1: Loading DE500BA into slot 4

```
load DE500BA/dec21x4x EWA bus=pci_1 device=7 function=0
```

Example 2: Loading multiple DE500BA's into slot 4, populating all 4 functions (gives 4 Ethernet ports)

```
load DE500BA/dec21x4x EWA bus=pci_1 device=7 function=0
load DE500BA/dec21x4x EWB bus=pci_1 device=7 function=1
load DE500BA/dec21x4x EWC bus=pci_1 device=7 function=2
load DE500BA/dec21x4x EWD bus=pci_1 device=7 function=3
```

Example 3: Loading mixture of KZPBA and DE500BA into slot 1, populating 2 functions out of 4

```
load KZPBA PKC bus=pci_0 device=7 function=0
load DE500BA/dec21x4x EWA bus=pci_0 device=7 function=1
```

ℹ In the above example device name is PKC as there are two built-in PK-like PCI SCSI Adapters located “closer” to CPU and therefore assigned names PKA and PKB.

Back to Table of Content

AlphaServer DS25 (6 PCI slots)

In addition to 6 PCI vacant slots there are 7 PCI positions occupied by on-board devices. All 13 PCI positions are listed in the following table in the order in which Alpha SRM console enumerates them.

Slot	pci_<N>	device	function	irq	Description	Preloaded Name
PCI0 (bus=pci_0)						
-	0	7	0	-	ALi M1543C PCI ISA bridge	
-	0	8	0	-	Intel i82559 PCI Ethernet Adapter	EIA (EWA)
1	0	9	0	24	<option>	
			1	25	<option>, function 1	
			2	26	<option>, function 2	
			3	27	<option>, function 3	
2	0	10	0	12	<option>	
			1	13	<option>, function 1	
			2	14	<option>, function 2	
			3	15	<option>, function 3	
-	0	16	0	-	ALi M1543C PCI IDE/ATAPI controller	DQA, DQB
-	0	19	1	-	ALi M1543C PCI USB adapter	
PCI1 (bus=pci_1)						
3	1	1	0	28	<option>	
			1	29	<option>, function 1	
			2	30	<option>, function 2	
			3	31	<option>, function 3	
4	1	2	0	32	<option>	
			1	33	<option>, function 1	
			2	34	<option>, function 2	
			3	35	<option>, function 3	
PCI2 (bus=pci_2)						
-	2	1	0	-	Adaptec AIC-7899 (channel 0)	PKA
			1	-	Adaptec AIC-7899 (channel 1)	PKB
-	2	5	0	-	BroadCom BCM5703 PCI Ethernet Adapter	EIB (EWB)
PCI3 (bus=pci_3)						
5	3	1	0	36	<option>	
			1	37	<option>, function 1	
			2	38	<option>, function 2	
			3	39	<option>, function 3	
6	3	2	0	40	<option>	
			1	41	<option>, function 1	
			2	42	<option>, function 2	

	3	43	<option>, function 3	

The IRQ stands for bit position in DRIR of TITAN Chip. It has nothing to do with “ISA” style interrupts which are routed to IRQ 55 (including ALi M1543C PCI IDE/ATAPI controller).

So far the CHARON-AXP emulators do not emulate Intel i82559. Instead, emulation of DECchip 21143 is used.

So far the CHARON-AXP emulators do not emulate ALi M1543C PCI USB adapter. So position of the device 19, function 0 on the PCI 0 remains empty.

Unless SCSI option is plugged into PCI slot 1, 2, 3, or 4, the onboard SCSI controllers appear as PKA (pka7.0.0.1.2) and PKB (pkb7.0.0.101.2) respectively.

So far the CHARON-AXP emulators do not emulate Adaptec AIC-7899. Instead, emulation of QLOGIC ISP1040B is used.

So far the CHARON-AXP emulators do not emulate BroadCom BCM5703. Instead, emulation of DECchip 21143 is used.

Example 1: Loading DE500BA into slot 5

```
load DE500BA/dec21x4x EWA bus=pci_3 device=1 function=0
```

Example 2: Loading multiple DE500BA’s into slot 5, populating all 4 functions (gives 4 Ethernet ports)

```
load DE500BA/dec21x4x EWA bus=pci_3 device=1 function=0
load DE500BA/dec21x4x EWB bus=pci_3 device=1 function=1
load DE500BA/dec21x4x EWC bus=pci_3 device=1 function=2
load DE500BA/dec21x4x EWD bus=pci_3 device=1 function=3
```

Example 3: Loading mixture of KZPBA and DE500BA into slot 6, populating 2 functions out of 4

```
load KZPBA PKC bus=pci_3 device=2 function=0
load DE500BA/dec21x4x EWA bus=pci_3 device=2 function=1
```

ℹ In the above example device name is PKC as there are two built-in PK-like PCI SCSI Adapters located “closer” to CPU and therefore assigned names PKA and PKB.

Back to Table of Content

AlphaServer ES40 (10 PCI slots)

In addition to 10 PCI vacant slots there are 3 PCI positions occupied by on-board devices. All 13 PCI positions are listed in the following table in the order in which Alpha SRM console enumerates them.

Slot	pci_<N>	device	function	irq	Description	Preloaded Name
PCI1 (bus=pci_1)						
5	1	1	0	24	<option>	
			1	25	<option>, function 1	
			2	26	<option>, function 2	
			3	27	<option>, function 3	
6	1	2	0	28	<option>	
			1	29	<option>, function 1	
			2	30	<option>, function 2	
			3	31	<option>, function 3	
7	1	3	0	32	<option>	
			1	33	<option>, function 1	
			2	34	<option>, function 2	
			3	35	<option>, function 3	
8	1	4	0	36	<option>	
			1	37	<option>, function 1	
			2	38	<option>, function 2	
			3	39	<option>, function 3	
9	1	5	0	40	<option>	
			1	41	<option>, function 1	
			2	42	<option>, function 2	
			3	43	<option>, function 3	
10	1	6	0	44	<option>	
			1	45	<option>, function 1	
			2	46	<option>, function 2	
			3	47	<option>, function 3	
PCI0 (bus=pci_0)						
1	0	1	0	8	<option>	
			1	9	<option>, function 1	
			2	10	<option>, function 2	
			3	11	<option>, function 3	
2	0	2	0	12	<option>	
			1	13	<option>, function 1	
			2	14	<option>, function 2	
			3	15	<option>, function 3	

3	0	3	1	16	<option>	
			1	17	<option>, function 1	
			2	18	<option>, function 2	
			3	19	<option>, function 3	
4	0	4	0	20	<option>	
			1	21	<option>, function 1	
			2	22	<option>, function 2	
			3	23	<option>, function 3	
-	0	5	0	-	ALi M1543C PCI ISA bridge	
-	0	15	0	-	ALi M1543C PCI ISA bridge	DQA, DQB
-	0	19	0	-	ALi M1543C PCI USB adapter	

The IRQ stands for bit position in DRIR of Tsunami/Typhoon chip. It has nothing to do with “ISA” style interrupts which are routed to IRQ 55 (including ALi M1543C PCI IDE/ATAPI controller).

So far the CHARON-AXP emulators do not support virtual ALi M1543C PCI USB adapter. So position of the device 19, function 0 on the PCI 0 remains empty.

Example 1: Loading DE500BA into slot 5

```
load DE500BA/dec21x4x EWA bus=pci_1 device=1 function=0
```

Example 2: Loading multiple DE500BA’s into slot 5, populating all 4 functions (gives 4 Ethernet ports)

```
load DE500BA/dec21x4x EWA bus=pci_1 device=1 function=0
load DE500BA/dec21x4x EWB bus=pci_1 device=1 function=1
load DE500BA/dec21x4x EWC bus=pci_1 device=1 function=2
load DE500BA/dec21x4x EWD bus=pci_1 device=1 function=3
```

Example 3: Loading mixture of KZPBA and DE500BA into slot 1, populating 2 functions out of 4

```
load KZPBA PKA bus=pci_0 device=1 function=0
load DE500BA/dec21x4x EWA bus=pci_0 device=1 function=1
```

Back to Table of Content

AlphaServer ES45 (10 PCI slots)

In addition to 10 PCI vacant slots there are 3 PCI positions occupied by on-board devices. All 13 PCI positions are listed in the following table in the order in which Alpha SRM console enumerates them.

Slot	pci_<N>	device	function	irq	Description	Preloaded Name
PCI0 (bus=pci_0)						
-	0	7	0	-	ALi M1543C PCI ISA bridge	
1	0	8	0	20	<option>	
			1	21	<option>, function 1	
			2	22	<option>, function 2	
			3	23	<option>, function 3	
2	0	9	0	24	<option>	
			1	25	<option>, function 1	
			2	26	<option>, function 2	
			3	27	<option>, function 3	
3	0	10	0	12	<option>	
			1	13	<option>, function 1	
			2	14	<option>, function 2	
			3	15	<option>, function 3	
4	0	11	0	16	<option>	
			1	17	<option>, function 1	
			2	18	<option>, function 2	
			3	19	<option>, function 3	
-	0	16	0	-	ALi M1543C PCI IDE/ATAPI controller	DQA, DQB
-	0	19	0	-	ALi M1543C PCI USB adapter	
PCI1 (bus=pci_1)						
5	1	1	0	28	<option>	
			1	29	<option>, function 1	
			2	30	<option>, function 2	
			3	31	<option>, function 3	
6	1	2	0	32	<option>	
			1	33	<option>, function 1	
			2	34	<option>, function 2	
			3	35	<option>, function 3	
PCI2 (bus=pci_2)						
7	2	1	0	0	<option>	
			1	1	<option>, function 1	
			2	2	<option>, function 2	
			3	3	<option>, function 3	

8	2	2	0	4	<option>	
			1	5	<option>, function 1	
			2	6	<option>, function 2	
			3	7	<option>, function 3	
PCI3 (bus=pci_3)						
9	3	1	0	36	<option>	
			1	37	<option>, function 1	
			2	38	<option>, function 2	
			3	39	<option>, function 3	
10	3	2	0	40	<option>	
			1	41	<option>, function 1	
			2	42	<option>, function 2	
			3	43	<option>, function 3	

The IRQ stands for bit position in DRIR of TITAN chip. It has nothing to do with “ISA” style interrupts which are routed to IRQ 55 (including ALi M1543C PCI IDE/ATAPI controller).

So far the CHARON-AXP emulators do not support virtual ALi M1543C PCI USB adapter. So position of the device 19, function 0 on the PCI 0 remains empty.

Example 1: Loading DE500BA into slot 5

```
load DE500BA/dec21x4x EWA bus=pci_1 device=1 function=0
```

Example 2: Loading multiple DE500BA’s into slot 5, populating all 4 functions (gives 4 Ethernet ports)

```
load DE500BA/dec21x4x EWA bus=pci_1 device=1 function=0
load DE500BA/dec21x4x EWB bus=pci_1 device=1 function=1
load DE500BA/dec21x4x EWC bus=pci_1 device=1 function=2
load DE500BA/dec21x4x EWD bus=pci_1 device=1 function=3
```

Example 3: Loading mixture of KZPBA and DE500BA into slot 1, populating 2 functions out of 4

```
load KZPBA PKA bus=pci_0 device=8 function=0
load DE500BA/dec21x4x EWA bus=pci_0 device=8 function=1
```

Back to Table of Content

AlphaServer GS80 (8 PCI busses)

Slot	pci_<N>	device	function	irq	Description	Preloaded Name
QBB0.PCA0.PCI0 (bus=qbb_0_pca_0_pci_0)						
0/1	0	1	0	36	QLOGIC ISP1040B PCI SCSI Adapter	PKA
2	0	2	0	40	<option>	
3	0	3	0	44	<option>	
-	0	7	0	-	ALi M1543C PCI ISA bridge	
-	0	15	0	-	ALi M1543C PCI IDE/ATAPI controller	DQA
-	0	19	0	-	ALi M1543C PCI USB adapter	
QBB0.PCA0.PCI1 (bus=qbb_0_pca_0_pci_1)						
4	1	4	0	48	<option>	
5	1	5	0	52	<option>	
6	1	6	0	56	<option>	
7	1	7	0	60	<option>	
QBB0.PCA1.PCI0 (bus=qbb_0_pca_1_pci_0)						
0/1	2	0	0	32	<option>	
2	2	2	0	40	<option>	
3	2	3	0	44	<option>	
QBB0.PCA1.PCI1 (bus=qbb_0_pca_1_pci_1)						
4	3	4	0	48	<option>	
5	3	5	0	52	<option>	
6	3	6	0	56	<option>	
7	3	7	0	60	<option>	
QBB1.PCA0.PCI0 (bus=qbb_1_pca_0_pci_0)						
0/1	8	0	0	32	<option>	
2	8	2	0	40	<option>	
3	8	3	0	44	<option>	
QBB1.PCA0.PCI1 (bus=qbb_1_pca_0_pci_1)						
4	9	4	0	48	<option>	
5	9	5	0	52	<option>	
6	9	6	0	56	<option>	
7	9	7	0	60	<option>	
QBB1.PCA1.PCI0 (bus=qbb_1_pca_1_pci_0)						
0/1	10	0	0	32	<option>	
2	10	2	0	40	<option>	
3	10	3	0	44	<option>	
QBB1.PCA1.PCI1 (bus=qbb_1_pca_1_pci_1)						

4	11	4	0	48	<option>	
5	11	5	0	52	<option>	
6	11	6	0	56	<option>	
7	11	7	0	60	<option>	

PCI 2 and 3 on each QBB are not populated.

So far the CHARON-AXP emulators do not support virtual ALi M1543C PCI USB adapter. So position of the device 19, function 0 on the PCI 0 remains empty.

Total number of PCI devices configured through CFG file may not exceed 20.

Example: Loading DE500BA into slot 2 of QBB0.PCA0

```
load DE500BA/dec21x4x EWA bus=qbb_0_pca_0_pci_0 device=2 function=0
```

Back to Table of Content

AlphaServer GS160 (16 PCI busses)

Slot	pci_<N>	device	function	irq	Description	Preloaded Name
QBB0.PCA0.PCI0 (bus=qbb_0_pca_0_pci_0)						
0/1	0	1	0	36	QLOGIC ISP1040B PCI SCSI Adapter	PKA
2	0	2	0	40	<option>	
3	0	3	0	44	<option>	
-	0	7	0	-	ALi M1543C PCI ISA bridge	
-	0	15	0	-	ALi M1543C PCI IDE/ATAPI controller	DQA
-	0	19	0	-	ALi M1543C PCI USB adapter	
QBB0.PCA0.PCI1 (bus=qbb_0_pca_0_pci_1)						
4	1	4	0	48	<option>	
5	1	5	0	52	<option>	
6	1	6	0	56	<option>	
7	1	7	0	60	<option>	
QBB0.PCA1.PCI0 (bus=qbb_0_pca_1_pci_0)						
0/1	2	0	0	32	<option>	
2	2	2	0	40	<option>	
3	2	3	0	44	<option>	
QBB0.PCA1.PCI1 (bus=qbb_0_pca_1_pci_1)						
4	3	4	0	48	<option>	
5	3	5	0	52	<option>	
6	3	6	0	56	<option>	
7	3	7	0	60	<option>	
QBB1.PCA0.PCI0 (bus=qbb_1_pca_0_pci_0)						
0/1	8	0	0	32	<option>	
2	8	2	0	40	<option>	
3	8	3	0	44	<option>	
QBB1.PCA0.PCI1 (bus=qbb_1_pca_0_pci_1)						
4	9	4	0	48	<option>	
5	9	5	0	52	<option>	
6	9	6	0	56	<option>	
7	9	7	0	60	<option>	
QBB1.PCA1.PCI0 (bus=qbb_1_pca_1_pci_0)						
0/1	10	0	0	32	<option>	
2	10	2	0	40	<option>	
3	10	3	0	44	<option>	
QBB1.PCA1.PCI1 (bus=qbb_1_pca_1_pci_1)						

4	11	4	0	48	<option>	
5	11	5	0	52	<option>	
6	11	6	0	56	<option>	
7	11	7	0	60	<option>	
QBB2.PCA0.PCI0 (bus=qbb_2_pca_0_pci_0)						
0/1	16	0	0	32	<option>	
2	16	2	0	40	<option>	
3	16	3	0	44	<option>	
QBB2.PCA0.PCI1 (bus=qbb_2_pca_0_pci_1)						
4	17	4	0	48	<option>	
5	17	5	0	52	<option>	
6	17	6	0	56	<option>	
7	17	7	0	60	<option>	
QBB2.PCA1.PCI0 (bus=qbb_2_pca_1_pci_0)						
0/1	18	0	0	32	<option>	
2	18	2	0	40	<option>	
3	18	3	0	44	<option>	
QBB2.PCA1.PCI1 (bus=qbb_2_pca_1_pci_1)						
4	19	4	0	48	<option>	
5	19	5	0	52	<option>	
6	19	6	0	56	<option>	
7	19	7	0	60	<option>	
QBB3.PCA0.PCI0 (bus=qbb_3_pca_0_pci_0)						
0/1	24	0	0	32	<option>	
2	24	2	0	40	<option>	
3	24	3	0	44	<option>	
QBB3.PCA0.PCI1 (bus=qbb_3_pca_0_pci_1)						
4	25	4	0	48	<option>	
5	25	5	0	52	<option>	
6	25	6	0	56	<option>	
7	25	7	0	60	<option>	
QBB3.PCA1.PCI0 (bus=qbb_3_pca_1_pci_0)						
0/1	26	0	0	32	<option>	
2	26	2	0	40	<option>	
3	26	3	0	44	<option>	
QBB0.PCA0.PCI1 (bus=qbb_0_pca_0_pci_1)						
4	27	4	0	48	<option>	
5	27	5	0	52	<option>	
6	27	6	0	56	<option>	

7	27	7	0	60	<option>	

PCA 2 and 3 on each QBB are not populated in emulator.

So far the CHARON-AXP emulators do not emulate ALi M1543C PCI USB adapter. So position of the device 19, function 0 on the PCI 0 on QBB 0 remains empty.

Total number of PCI devices configured through CFG file may not exceed 20.

Example: Loading DE500BA into slot 2 of QBB0.PCA0

```
load DE500BA/dec21x4x EWA bus=qbb_0_pca_0_pci_0 device=2 function=0
```

Back to Table of Content

AlphaServer GS320 (32 PCI busses)

Slot	pci_<N>	device	function	irq	Description	Preloaded Name
QBB0.PCA0.PCI0 (bus=qbb_0_pca_0_pci_0)						
0/1	0	1	0	36	QLOGIC ISP1040B PCI SCSI Adapter	PKA
2	0	2	0	40	<option>	
3	0	3	0	44	<option>	
-	0	7	0	-	ALi M1543C PCI ISA bridge	
-	0	15	0	-	ALi M1543C PCI IDE/ATAPI controller	DQA
-	0	19	0	-	ALi M1543C PCI USB adapter	
QBB0.PCA0.PCI1 (bus=qbb_0_pca_0_pci_1)						
4	1	4	0	48	<option>	
5	1	5	0	52	<option>	
6	1	6	0	56	<option>	
7	1	7	0	60	<option>	
QBB0.PCA1.PCI0 (bus=qbb_0_pca_1_pci_0)						
0/1	2	0	0	32	<option>	
2	2	2	0	40	<option>	
3	2	3	0	44	<option>	
QBB0.PCA1.PCI1 (bus=qbb_0_pca_1_pci_1)						
4	3	4	0	48	<option>	
5	3	5	0	52	<option>	
6	3	6	0	56	<option>	
7	3	7	0	60	<option>	
QBB1.PCA0.PCI0 (bus=qbb_1_pca_0_pci_0)						
0/1	8	0	0	32	<option>	
2	8	2	0	40	<option>	
3	8	3	0	44	<option>	
QBB1.PCA0.PCI1 (bus=qbb_1_pca_0_pci_1)						
4	9	4	0	48	<option>	
5	9	5	0	52	<option>	
6	9	6	0	56	<option>	
7	9	7	0	60	<option>	
QBB1.PCA1.PCI0 (bus=qbb_1_pca_1_pci_0)						
0/1	10	0	0	32	<option>	
2	10	2	0	40	<option>	
3	10	3	0	44	<option>	
QBB1.PCA1.PCI1 (bus=qbb_1_pca_1_pci_1)						

4	11	4	0	48	<option>	
5	11	5	0	52	<option>	
6	11	6	0	56	<option>	
7	11	7	0	60	<option>	
QBB2.PCA0.PCI0 (bus=qbb_2_pca_0_pci_0)						
0/1	16	0	0	32	<option>	
2	16	2	0	40	<option>	
3	16	3	0	44	<option>	
QBB2.PCA0.PCI1 (bus=qbb_2_pca_0_pci_1)						
4	17	4	0	48	<option>	
5	17	5	0	52	<option>	
6	17	6	0	56	<option>	
7	17	7	0	60	<option>	
QBB2.PCA1.PCI0 (bus=qbb_2_pca_1_pci_0)						
0/1	18	0	0	32	<option>	
2	18	2	0	40	<option>	
3	18	3	0	44	<option>	
QBB2.PCA1.PCI1 (bus=qbb_2_pca_1_pci_1)						
4	19	4	0	48	<option>	
5	19	5	0	52	<option>	
6	19	6	0	56	<option>	
7	19	7	0	60	<option>	
QBB3.PCA0.PCI0 (bus=qbb_3_pca_0_pci_0)						
0/1	24	0	0	32	<option>	
2	24	2	0	40	<option>	
3	24	3	0	44	<option>	
QBB3.PCA0.PCI1 (bus=qbb_3_pca_0_pci_1)						
4	25	4	0	48	<option>	
5	25	5	0	52	<option>	
6	25	6	0	56	<option>	
7	25	7	0	60	<option>	
QBB3.PCA1.PCI0 (bus=qbb_3_pca_1_pci_0)						
0/1	26	0	0	32	<option>	
2	26	2	0	40	<option>	
3	26	3	0	44	<option>	
QBB3.PCA1.PCI1 (bus=qbb_3_pca_1_pci_1)						
4	27	4	0	48	<option>	
5	27	5	0	52	<option>	
6	27	6	0	56	<option>	

7	27	7	0	60	<option>	
QBB4.PCA0.PCI0 (bus=qbb_4_pca_0_pci_0)						
0/1	32	0	0	32	<option>	
2	32	2	0	40	<option>	
3	32	3	0	44	<option>	
QBB4.PCA0.PCI1 (bus=qbb_4_pca_0_pci_1)						
4	33	4	0	48	<option>	
5	33	5	0	52	<option>	
6	33	6	0	56	<option>	
7	33	7	0	60	<option>	
QBB4.PCA1.PCI0 (bus=qbb_4_pca_1_pci_0)						
0/1	34	0	0	32	<option>	
2	34	2	0	40	<option>	
3	34	3	0	44	<option>	
QBB4.PCA1.PCI1 (bus=qbb_4_pca_1_pci_1)						
4	35	4	0	48	<option>	
5	35	5	0	52	<option>	
6	35	6	0	56	<option>	
7	35	7	0	60	<option>	
QBB5.PCA0.PCI0 (bus=qbb_5_pca_0_pci_0)						
0/1	40	0	0	32	<option>	
2	40	2	0	40	<option>	
3	40	3	0	44	<option>	
QBB5.PCA0.PCI1 (bus=qbb_5_pca_0_pci_1)						
4	41	4	0	48	<option>	
5	41	5	0	52	<option>	
6	41	6	0	56	<option>	
7	41	7	0	60	<option>	
QBB5.PCA1.PCI0 (bus=qbb_5_pca_1_pci_0)						
0/1	42	0	0	32	<option>	
2	42	2	0	40	<option>	
3	42	3	0	44	<option>	
QBB5.PCA1.PCI1 (bus=qbb_5_pca_1_pci_1)						
4	43	4	0	48	<option>	
5	43	5	0	52	<option>	
6	43	6	0	56	<option>	
7	43	7	0	60	<option>	
QBB6.PCA0.PCI0 (bus=qbb_6_pca_0_pci_0)						
0/1	48	0	0	32	<option>	

2	48	2	0	40	<option>	
3	48	3	0	44	<option>	
QBB6.PCA0.PCI1 (bus=qbb_6_pca_0_pci_1)						
4	49	4	0	48	<option>	
5	49	5	0	52	<option>	
6	49	6	0	56	<option>	
7	49	7	0	60	<option>	
QBB6.PCA1.PCI0 (bus=qbb_6_pca_1_pci_0)						
0/1	50	0	0	32	<option>	
2	50	2	0	40	<option>	
3	50	3	0	44	<option>	
QBB6.PCA1.PCI1 (bus=qbb_6_pca_1_pci_1)						
4	51	4	0	48	<option>	
5	51	5	0	52	<option>	
6	51	6	0	56	<option>	
7	51	7	0	60	<option>	
QBB7.PCA0.PCI0 (bus=qbb_7_pca_0_pci_0)						
0/1	56	0	0	32	<option>	
2	56	2	0	40	<option>	
3	56	3	0	44	<option>	
QBB7.PCA0.PCI1 (bus=qbb_7_pca_0_pci_1)						
4	57	4	0	48	<option>	
5	57	5	0	52	<option>	
6	57	6	0	56	<option>	
7	57	7	0	60	<option>	
QBB7.PCA1.PCI0 (bus=qbb_7_pca_1_pci_0)						
0/1	58	0	0	32	<option>	
2	58	2	0	40	<option>	
3	58	3	0	44	<option>	
QBB7.PCA1.PCI1 (bus=qbb_7_pca_1_pci_1)						
4	59	4	0	48	<option>	
5	59	5	0	52	<option>	
6	59	6	0	56	<option>	
7	59	7	0	60	<option>	

PCA 2 and 3 on each QBB are not populated in emulator.

So far the MSC/AXP emulators do not emulate ALi M1543C PCI USB adapter. So position of the device 19, function 0 on the PCI 0 on QBB 0 remains empty.

Total number of PCI devices configured through CFG file may not exceed 20.

Example: Loading DE500BA into slot 2 of QBB0.PCA0

```
load DE500BA/dec21x4x EWA bus=qbb_0_pca_0_pci_0 device=2 function=0
```

Back to Table of Content

Disks and tapes

Contents

KZPBA PCI SCSI adapter

Table of Contents

Back to Table of Contents

General description

KZPBA is a PCI SCSI adapter based on the QLogic ISP1040 Fast Wide SCSI adapter chip for HP Alpha.

In CHARON-AXP environment it supports up to 120 disks and tapes.

ℹ For systems with more than 16 heavily used units it is recommended to configure several virtual KZPBA PCI SCSI adapters and distribute the heavily loaded units evenly between the adapters.

Back to Table of Contents

Loading KZPBA storage adapter

Syntax for loading KZPBA storage adapter:

```
load KZPBA <name>
```

Example:

```
load KZPBA PKA
```

The adapter instance name ("PKA" in the example above) is used then for parametrization, for example:

```
set PKA container[602]="/Mydisks/vms_distributive.vdisk"
```

The numbers in the square brackets represent SCSI ID and LUN of the devices on the virtual KZPBA SCSI bus.

They have the following format: **XXYY**, where:

Parameter	Range	Description
XX	0..15	SCSI ID
YY	00..07	LUN

By default KZPBA adapter uses first available PCI slot. If instead some particular slot is needed, refer to this section for details of specific placement of PCI peripherals on CHARON-AXP PCI bus.

By default each loaded KZPBA SCSI PCI adapter has SCSI ID=7. This setting can be changed with "scsi_id" parameter, for example:

```
set PKA scsi_id=0
```

ℹ CHARON-AXP HP Alpha models may have one or two KZPBA adapters preloaded.

Back to Table of Contents

Configuration parameters

The KZPBA PCI SCSI adapter emulation has the following configuration parameters:

scsi_id

<table>
<tr><th>Parameter</th><td>scsi_id</td></tr>
<tr><th>Type</th><td>Numeric</td></tr>
<tr><th>Value</th><td>Specifies SCSI ID of KZPBA PCI SCSI Adapter in a range 0..7
By default the "scsi_id" configuration parameter is set to 7.
Example:
<code>set PKA scsi_id=0</code></td></tr>
</table>

host, port

<table>
<tr><th>Parameter</th><td>host, port</td></tr>
<tr><th>Type</th><td>Text String</td></tr>
<tr><th>Value</th><td>These parameters are used in SCSI cluster configurations.
<table>
<tr><th>Parameter</th><th>Description</th></tr>
<tr><td>host</td><td>Specifies remote end-point (remote host name and, optionally, TCP/IP port on remote host) of SCSI connection between this KZPBA PCI SCSI adapter and remote KZPBA PCI SCSI adapter on some host.</td></tr>
<tr><td>port</td><td>Specifies local end-point (TCP/IP port on local host) of SCSI connection between this KZPBA PCI SCSI adapter and remote KZPBA PCI SCSI adapter on some host.</td></tr>
</table>
By default the "host" and "port" configuration options are not specified.
Syntax:
<code>port[connection-number]=<local port> host[connection-number]="<host-name{:tcpip-port-no}>"</code>
where: connection_number = remote_scsi_id * 100 + lun_id
Example:
<code>set PKA port[600]=17060 host[600]="localhost:16070"</code></td></tr>
</table>

container

Parameter

container[N]

N is "XXYY" number, where:

- XX - SCSI ID (0..15)
- YY - LUN (00..07)

Type

Text String

Value

Possible values of the parameter are strings in one of the following forms:

- **Physical disk**
  - "/dev/sd<L>" where "L" is letter.

    ⚠ Be careful not to destroy all the information from the disk dedicated to CHARON-AXP by mistak.

    These disks must not be formatted by the host OS.

    Example:

    ```
    set PKA container[0]="/dev/sdb"
    ```

    It is also possible to use not a whole disk, but previously created partitions on it. In this case the syntax is the following: "/dev/sd<L><N>" where N is the number of partition to be used.

    Example:

    ```
    set PKA container[0]="/dev/sdc1"
    ```

- **Multipath disk**
  - "/dev/dm-<N>"
  - "/dev/mapper/mpath<N>"
  - "/dev/mapper/disk<N>"

    ⚠ Be careful not to destroy all the information from the disk dedicated to CHARON-AXP by mistake.

    These disks must not be formatted by the host OS.

    Example:

    ```
    set PKA container[100]="/dev/dm-0"
    ```

- **Direct mapping to some SCSI device, for example, a SCSI disk or tape reader**
  - "/dev/sg<N>"

    Example:

    ```
    set PKA container[200]="/dev/sg0"
    ```

- **CD-ROM device**
  - "/dev/sr<N>"
  - "/dev/cdrom"
  - "/dev/cdrom<N>"

    Example:

    ```
    set PKA container[400]="/dev/sr0"
    ```

- **ISO file for reading distribution CD-ROM image**
  - *[<drive>":\"<path-name>"\"]<file-name>[".iso"]*
    Mapping may also include the full path (recommended), for example: "/my_disks/vms_distributive.iso"

    Example:

    ```
    set PKA container[600]="/my_disks/vms_distributive.iso"
    ```

- **File representing a physical disk of the HP Alpha system (disk image)**
  - *[<drive>":\"<path-name>"\"]<file-name>[".vdisk"]*
    These files can be created from scratch with "mkdskcmd" utility. Data and OS disk backups are transferred from the original system via tapes or network and restored into these container files.

    Mapping may also include the full path, for example: "/my_disks/my_boot_disk.vdisk"

    Example:

    ```
    set PKA container[401]="my_dka401.vdisk"
    ```

- **File representing the tape (tape image)**
  - *[<drive>":\"<path-name>"\"]<file-name>".vtape"*

    These files are created automatically.

    Mapping may also include a full path (recommended), for example: "/my_tapes/backup.vtape"

    Example:

    ```
    set PKA container[500]="my_mka500.vtape"
    ```

This parameter is initially not set, thus creating NO storage elements on the controller.

media_type

Parameter	media_type[N] N is "XXYY" number, where: - XX - SCSI ID (0..15) - YY - LUN (00..07)
Type	Text String
Value	Instructs CHARON-AXP to use the supplied value as the PRODUCT field in the SCSI INQUIRY data returned to a software running on virtual HP Alpha system in response to SCSI INQUIRY command. If not specified, CHARON-AXP attempts to guess the SCSI INQUIRY data based on virtual SCSI device type and underlying container (which is specified in the corresponding container configuration parameter). Initially is not specified. Example: `set PKA media_type[0]="HSZ70"`

removable

Parameter	removable[N] N is "XXYY" number, where: ▪ XX - SCSI ID (0..15) ▪ YY - LUN (00..07)
Type	Boolean
Value	When set to "true", the removable configuration parameter instructs CHARON-AXP to report the corresponding virtual SCSI device as removable. By default the removable configuration parameter is set to "false". Example: `set PKA removable[400]=true` ℹ Note that virtual SCSI tapes and CD-ROM devices are always reported as removable regardless of the "removable" configuration parameter.

geometry

Parameter

geometry[N]

N is "XXYY" number, where:

- XX - SCSI ID (0..15)
- YY - LUN (00..07)

Type

Text String

Value

This formatted string value specifies the explicit geometry of the disk storage element. This parameter is not applicable to tape storage elements.

The string format is <X>”/”<Y>[“/”<Z>] or <X>”,”<Y>[“,”<Z>][“,”<B>] where:

Parameter	Description
X	The number of sectors per track
Y	The number of tracks per cylinder
Z	The number of cylinders on the unit. If omitted, Z is calculated based on X, Y and the total number of sectors on the unit that reflects the size of the disk storage element. This is an optional parameter.
B	The total size of the disk (in blocks) reported to the guest OS. If omitted it is calculated automatically. This is an optional parameter.

If this parameter is not set, CHARON-AXP will configure the geometry based on the most probable disk type.

Initially not set.

> It is possiblle to specify each parameter independently of another one. The following syntax is used for that:
>
> ```
> set PKA geometry[300]="*,*,*,16777210"
> ```

The syntax described above is applicable only to disk storage elements. If the container is a tape image, the following format is used instead:

Syntax:

```
"<image-size>[, <early-warning-zone-size>]"
```

where:

Parameter	Description
image-size	The tape size in MB
early-warning-zone-size	The size (in KB) of the space left on the tape when a warning to the OS is issued. If omitted, 64K is assumed.

Example:

```
set PKA geometry[603] = "255/255"
```

use_io_file_buffering

Parameter	use_io_file_buffering[N] N is "XXYY" number, where: ▪ XX - SCSI ID (0..15) ▪ YY - LUN (00..07)
Type	Boolean
Value	When set to "true", instructs CHARON-AXP to enable host operating system I/O cache on reading/writing operations. When enabled, host operating system I/O cache may significantly improve I/O performance of the virtual system. At the same time maintaining I/O cache requires additional host resources (CPU and memory) which may negatively affect overall performance of the virtual system. Initially is set to "false". Example: `set PKA use_io_file_buffering[603]=true`

When a tape or disk image connected to an emulated KZPBA controller is dismounted by OpenVMS, it is disconnected from CHARON-AXP and can be manipulated. It can be replaced with a different disk image if it keeps the same name. This capability may be useful when designing back-up and restore procedures. When copying CHARON-AXP disk images while CHARON-AXP is running, please take care to minimize the risk of overloading a heavily loaded CHARON-AXP host system. For example, using a sequential series of simple ftp binary copies is less resource intensive and thus less disruptive than multiple, simultaneous copies.

Empty disk images are created with the "mkdskcmd" utility. Tape images ("*.vtape") will be created automatically if they don't exist (no utility needed).

CHARON-AXP is able to boot from disk images of any OpenVMS/Alpha and Tru64 version.

> The virtual KZPBA storage controller examines the file extension `(vdisk or vtape)` to distinguish between a disk image and a tape image.
>
> Configured physical devices or tape/disk images that do not exist on the host system will, in general, cause OpenVMS/Alpha to report the unit offline. In some cases this will result in a VMS BUG CHECK. In this case, an error message will be written to the log file.

Back to Table of Contents

KGPSA-CA PCI Fibre Channel adapter

Table of Contents

Back to Table of Contents

General description

CHARON-AXP supports emulation of DEC-KGPSA-CA PCI Fibre Channel adapter.

Every instance of KGPSA-CA works in one of the three following modes:

- Fabric virtualization mode (creating virtual fabric in combination with virtual FC-3 Storage Controller). This is default mode.
- CHARON PCI Pass Through mode (using a specific CHARON PCI Pass Through driver)

Back to Table of Contents

Loading KGPSA storage adapter

Syntax for loading KGPSA-CA storage adapter:

```
load KGPSA <name>
```

Example:

```
load KGPSA FGA
```

The adapter instance name ("FGA" in the example above) is used then for parametrization, for example:

```
set FGA container[100]="/my_disks/vms_distributive.vdisk"
```

Numbers in the square brackets represent KGPSA-CA units. They can be in the range 0..32766, but no more than 255 units can be configured on a single controller.

By default KGPSA-CA adapter uses first available PCI slot. If instead some particular slot is needed, refer to this section for details of specific placement of PCI peripherals on CHARON-AXP PCI bus.

Back to Table of Contents

Configuration parameters

The KGPSA-CA PCI FC adapter emulation has the following configuration parameters:

host_bus_location

Parameter	host_bus_location
Type	Text String

Value	⚠ Pass through mode only!
	Establish connection between virtual DEC-KGPSA-CA PCI FC adapter and physical EMULEX LightPulse PCI/PCI-X/PCIe FC adapter (pass through mode)
	Syntax:
	`load KGPSA <controller name> host_bus_location="/dev/kgpsa<X>"`
	Example:
	`load KGPSA FGA host_bus_location="/dev/kgpsa0"`

wwid

Parameter	wwid[N] N is 0..32766 (no more than 255 units)
Type	Text String

Value	Sets WWID for emulated KGPSA adapter unit.
	Syntax:
	`set <controller name> wwid[unit-number]="XXXX-XXXX-XXXX-XXXX-XXXX-XXXX-XXXX-XXXX"`
	Example:
	`set FGA wwid[2]="6008-05F3-0005-2950-BF8E-0B86-A0C7-0001"`

container

<table>
<tr>
<td><b>Parameter</b></td>
<td>

container[N]

N is 0..32766 (no more than 255 units)

</td>
</tr>
<tr>
<td><b>Type</b></td>
<td>

Text String

</td>
</tr>
<tr>
<td><b>Value</b></td>
<td>

Possible values of the parameter are strings in one of the following forms:

- **Physical disk**
  - "/dev/sd<L>", where "L" is letter

    ⚠ Be careful not to destroy all the information from the disk dedicated to CHARON-AXP by mistake.

    These disks must not be formatted by the host OS.

    Example:

    ```
    set FGA container[0]="/dev/sdb"
    ```

    It is also possible to use not a whole disk, but previously created partitions on it. In this case the syntax is the following: "/dev/sd<L><N>" where N is the number of partition to be used.

    Example:

    ```
    set FGA container[0]="/dev/sdc3"
    ```

- **Multipath disk**
  - "/dev/dm-<N>"
  - "/dev/mapper/mpath<N>"
  - "/dev/mapper/disk<N>"

    ⚠ Be careful not to destroy all the information from the disk dedicated to CHARON-AXP by mistake.

    These disks must not be formatted by the host OS.

    Example:

    ```
    set FGA container[200]="/dev/dm-0"
    ```

- **File representing a physical disk of the HP Alpha system (disk image)**
  - [<drive>":\"<path-name>"\"]<file-name>[".vdisk"]
    These files can be created from scratch with "mkdskcmd" utility. Data and OS disk backups are transferred from the original system via tapes or network and restored into these container files.

    Mapping may also include the full path (recommended), for example: "/my_disks/my_boot_disk.vdisk"

    Example:

    ```
    set FGA container[401]="my_dka401.vdisk"
    ```

This parameter is initially not set, thus creating NO storage elements on the controller.

</td>
</tr>
</table>

media_type

Parameter	media_type[N] N is 0..32766 (no more than 255 units)
Type	Text String
Value	Instructs CHARON-AXP to use the supplied value as the PRODUCT field in the FC INQUIRY data returned to a software running on virtual HP Alpha system in response to FC INQUIRY command. If not specified, CHARON-AXP attempts to guess the FC INQUIRY data based on virtual FC device type and underlying container (which is specified in the corresponding container configuration parameter). Initially is not specified. Example: `set FGA media_type[0]="HSZ70"`

removable

Parameter	removable[N] N is 0..32766 (no more than 255 units)
Type	Boolean
Value	When set to "true", the removable configuration parameter instructs CHARON-AXP to report the corresponding virtual FC device as removable. By default the removable configuration parameter is set to "false". Example: `set FGA removable[400]=true`

geometry

Parameter

geometry[N]

N is 0..32766 (no more than 255 units)

Type

Text String

Value

This formatted string value specifies the explicit geometry of the disk storage element.

The string format is <X>”/”<Y>[“/”<Z>] or <X>”,”<Y>[“,”<Z>][“,”<B>] where:

Parameter	Description
X	The number of sectors per track
Y	The number of tracks per cylinder
Z	The number of cylinders on the unit. If omitted, Z is calculated based on X, Y and the total number of sectors on the unit that reflects the size of the disk storage element. This is an optional parameter.
B	The total size of the disk (in blocks) reported to the guest OS. If omitted it is calculated automatically. This is an optional parameter.

If this parameter is not set, CHARON-AXP will configure the geometry based on the most probable disk type.

Initially not set.

Example:

```
set FGA geometry[201] = "255/255"
```

It is possiblle to specify each parameter independently of another one. The following syntax is used for that:

```
set FGA geometry[300]="*,*,*,16777210"
```

use_io_file_buffering

Parameter

use_io_file_buffering[N]

N is 0..32766 (no more than 255 units)

Type

Boolean

Value

When set to "true", instructs CHARON-AXP to enable host operating system I/O cache on reading/writing operations.

When enabled, host operating system I/O cache may significantly improve I/O performance of the virtual system. At the same time maintaining I/O cache requires additional host resources (CPU and memory) which may negatively affect overall performance of the virtual system.

Initially is set to "false".

Example:

```
set FGA use_io_file_buffering[300]=true
```

When a disk image connected to an emulated KGPSA-CA controller is dismounted by OpenVMS, it is disconnected from CHARON-AXP and can be manipulated. It can be replaced with a different disk image if it keeps the same name. This capability may be useful when designing back-up and restore procedures. When copying CHARON-AXP disk images while CHARON-AXP is running, please take care to minimize the risk of overloading a heavily loaded CHARON-AXP host system. For example, using a sequential series of simple ftp binary copies is less resource intensive and thus less disruptive than multiple, simultaneous copies.

Empty disk images are created with the "mkdskcmd" utility.

CHARON-AXP is able to boot from disk images of any OpenVMS/Alpha and Tru64 version.

Back to Table of Contents

Mapping to host resources

Fabric virtualization mode

In this mode KGPSA-CA PCI FC adapter can be directly mapped to physical disks (both local and iSCSI) and disk images as shown in the following example:

```
set FGA container[0]="/my_disks/my_dka401.vdisk"
set FGA container[100]="/dev/sdb"
set FGA container[200]="/dev/sdc2"
set FGA container[300]="/dev/dm-0"
```

See the "Configuration parameters" section for details.

Back to Table of Contents

Pass Through mode

The CHARON PCI Pass Through mode allows connection between virtual DEC-KGPSA-CA PCI FC adapter and physical EMULEX LightPulse PCI/PCI-X/PCIe FC adapter plugged into host’s PCI/PCI-X/PCIe slot.

Syntax:

```
load <controller name> host_bus_location="/dev/kgpsa<N>"
```

Example:

```
load KGPSA FGA host_bus_location="/dev/kgpsa0"
```

The following is a list of EMULEX LightPulse PCI/PCI-X/PCIe FC adapters supported by CHARON-AXP PCI Pass Through driver and suitable for emulation of KGPSA-CA PCI FC adapter in CHARON PCI Pass Through mode:

Supported	Not Supported	Not tested
LP8000 LP9000 LP9002 LP9802 LP10000 LP10000DC LP10000-S LPX1000 LP11002 LPe11002 (FC2242SR, A8003A) LPe1105 LPe12002 (AJ762B)	LPe1150 (FC2142SR, A8002A)	LPe11000

Back to Table of Contents

Pass Through mode establishing sequence

To establish "pass through" mode do the following:

1. Install the EMULEX LightPulse PCI/PCI-X/PCIe FC adapter (see below for a list of supported models) to some spare PCI/PCI-X/PCIe slot of the host system
2. Build PPT driver for EMULEX LightPulse PCI/PCI-X/PCIe FC
3. Install PPT driver for EMULEX LightPulse PCI/PCI-X/PCIe FC
4. Add PPT driver for EMULEX LightPulse PCI/PCI-X/PCIe FC to Linux startup
5. Map KGPSA-CA adapter(-s) to EMULEX LightPulse PCI/PCI-X/PCIe FC adapter instance(-s) in CHARON-AXP configuration file

> If kernel of the host system has been upgraded or reinstalled all the steps of the PPT KGPSA driver installation must be repeated

Back to Table of Contents

Building PPT driver for EMULEX LightPulse PCI/PCI-X/PCIe FC

To build PPT driver for EMULEX LightPulse PCI/PCI-X/PCIe FC do the following:

Step	Description

1	Make sure that the required building tools and include files are installed. If they are absent install them:
	`# yum groupinstall “Development Tools”`
	`# yum install kernel-headers kernel-develop`
	The kernel version must match the version of the installed kernel-headers (i.e. this packages must have same versions. It can be verified via "`rpm -q -a | grep kernel-`")
	Check that the "kernel" and the "kernel-headers" have the same version, and ensure that system is booted from this kernel version (not from some older one and etc) with "`uname -a`" command.

2	Open xterm and change the default directory to "`/opt/charon/drivers/kgpsa`": `# cd /opt/charon/drivers/kgpsa`
3	Issue "make clean; make" commands to build kernel object: `# make clean; make` ⚠ It is prohibited to use a module built on a certain version of kernel on another one.
4	Check that there are no compilation errors and the file "`ppt_kgpsa.ko`" has been built

Back to Table of Contents

Installation of PPT driver for EMULEX LightPulse PCI/PCI-X/PCIe FC

To install PPT driver for EMULEX LightPulse PCI/PCI-X/PCIe FC do the following:

Step	Description
1	Unload standard "lpfc" driver; to do that issue the following command: `# rmmod lpfc`
2	Load "`ppt_kgpsa.ko`" driver; to do that issue the following command: `# insmod ppt_kgpsa.ko`
3	Issue "dmesg" command and check that no error appeared during the driver loading, also check that the driver has found all KGPSA devices: `# dmesg`

Back to Table of Contents

Adding PPT driver for EMULEX LightPulse PCI/PCI-X/PCIe FC to Linux startup

To add PPT driver for EMULEX LightPulse PCI/PCI-X/PCIe FC to Linux startup do the following:

Step	Description
1	Disable auto-loading of Linux standard "lpfc" driver on boot. To do that add "lpfc" to the black list file "`/etc/modprobe.d/blacklist.conf`"
2	Copy the KGPSA-CA kernel module to the location of Linux kernel modules, for example: `# cp /opt/charon/drivers/kgpsa/ppt_kgpsa.ko /lib/modules/3.10.9-200.fc20.x86_64/kernel/drivers/scsi/` ⚠ The particular path may be different, depending on the kernel version and Linux distribution.

3	Enable auto load of the module:
	RedHat Linux:
	`# echo modprobe ppt_kgpsa >> /etc/rc.modules`
	Fedora Core Linux:
	`# echo ppt_kgpsa > /etc/modules-load.d/ppt_kgpsa.conf`

4	Regenerate new "initramfs" image with "mkinitrd": `# mkinitrd -f /boot/initramfs-3.10.9-200.fc20.x86_64.img 3.10.9-200.fc20.x86_64` ⚠ The particular path may be different, depending on the kernel version and Linux distribution.

Back to Table of Contents

Acer Labs 1543C IDE/ATAPI CD-ROM adapter

Table of Contents

Back to Table of Contents

General description

CHARON-AXP supports emulation of an integrated virtual Acer Labs 1543C IDE/ATAPI controller.

Back to Table of Contents

Loading Acer Labs 1543C IDE/ATAPI adapter

By default the integrated virtual Acer Labs 1543C IDE/ATAPI controller is preloaded with a name "ide".

Example:

```
set ide container="/dev/sg0"
```

Back to Table of Contents

Configuration parameters

The Acer Labs 1543C IDE/ATAPI adapter emulation has only one configuration parameter:

container

Parameter	container
Type	Text String
Value	• "/dev/sg<N>" Specifies a physical device correspondent to ATAPI or SATA CD/DVD-ROM drive attached to the host system. By default it is left unspecified. ℹ Read this article on how to find the physical device name for mapping.

Example

```
set ide container="/dev/sg0"
```

When running HP OpenVMS/Alpha Operating System on top of CHARON-AXP virtualization layer the specified CD/DVD-ROM drive is available as DQA0: device.

CHARON-AXP is able to boot any OpenVMS/Alpha and Tru64 version from Acer Labs 1543C IDE/ATAPI CD-ROM.

> Virtual Acer Labs 1543C IDE/ATAPI can me mapped only to physical CD-ROM drives. If a CD-ROM container or an ISO file should be used, it is required to utilize KZPBA-CA controller as it offers full support of both physical and virtual mappings to system resources.

Back to Table of Contents

Finding the target "/dev/sg" device

Table of Contents

Back to Table of Contents

General description

This section describes how to find proper "/dev/sg" device for CHARON mapping

Back to Table of Contents

Procedures of finding the target "/dev/sg" device

First method

In xterm console issue:

```
# cat /proc/scsi/sg/device_hdr; cat /proc/scsi/sg/devices
```

The output will look something like:

```
host    chan    id      lun     type    opens   qdepth  bus    online
4       0       0       0       5       1       1       0      1
5       0       0       0       0       1       1       0      1
```

The fifth field ("type") is the device type.

Value	Device
0	Disk
1	Tape
5	CD-ROM

The "N" in the "/dev/sgN" is the line number in this table (starting from 0) corresponded to the devices CHARON-AXP will use.

Thus "/dev/sg0" will be CD-ROM mapping in this example.

Back to Table of Contents

Second method

On a freshly booted system, issue the following command:

```
# dmesg | grep sg
```

The output will look like that:

```
[ 1.503622] sr 4:0:0:0: Attached scsi generic sg0 type 5

[ 1.780897] sd 5:0:0:0: Attached scsi generic sg1 type 0
```

⚠ This table lists all the devices, not only the real SCSI ones (SATA/IDE for example). CHARON supports only real SCSI devices.

Back to Table of Contents

Networking

Table of Contents

Back to Table of Contents

General description

CHARON-AXP supports emulation of the following network adapters:

- DE435
- DE450
- DE500AA
- DE500BA
- DE602
- DE602AA

Each of them is a PCI Ethernet adapter based on the DEC21040 (DE435, DE450, DE500AA and DE500BA) and the Intel i8255x (DE602 and DE602AA) PCI Ethernet adapter chips for the HP Alpha.

CHARON-AXP maps the virtual adapter to a dedicated Ethernet adapter in the Linux host system.

All the emulated controllers are loaded and configured in the same way.

⚠ The Ethernet adapter in the Linux host system must support dynamic changes of its MAC address (i.e. no reboot of the host system is required to change the MAC address), which is the case with nearly all modern Ethernet adapters.

ℹ By default the PCI Ethernet adapters use first available PCI slot. If instead some particular slot is needed, refer to this section for details of specific placement of PCI peripherals on CHARON-AXP PCI bus.

Configuration steps

To configure CHARON-AXP networking, follow these 3 steps:

1. **Load network adapter (if required)**

   Use the "load" command as shown below.

   Example:

For DEC21040 adapters	For Intel i8255x adapters
`load DE500BA/dec21x4x NIC`	`load DE602/i8255x NIC`

   ℹ By default each loaded virtual network adapter uses first available PCI slot. If instead some particular slot is neded, refer to this section for details of specific placement of PCI peripherals on CHARON-AXP PCI bus.

2. **Load "packet_port" or "tap_port"**

   Load "packet_port" or "tap_port" to connect network adapter to the host hardware network card (or to a virtual network interface).

   Example:

   ```
   load packet_port/chnetwrk NDIS interface = "eth1"
   ```

3. **Connect the loaded "packet_port" ("tap_port") to the loaded virtual network adapter**

   Connect the network adapter to the "packet_port" ("tap_port") by setting the interface name.

   Example:

   ```
   set NIC interface = NDIS
   ```

Back to Table of Contents

Configuration parameters

Each virtual network controller has the following parameters that are specified with the "set" command:

interface

Parameter	interface
Type	Text String
Value	Name of the corresponding instance of the "packet_port" or "tap_port" component

station_address

Parameter	station_address
Type	Text String

Value	The "station_address" provides the ability to configure the adapter’s permanent address. By default the adapter’s permanent address is read from the host system’s NIC.
	Format:
	`XX-XX-XX-XX-XX-XX`
	or
	`XX:XX:XX:XX:XX:XX`
	Example:
	`set EWA station_address="AF:01:AC:78:1B:CC"`

rx_fifo_size

Parameter	rx_fifo_size
Type	Numeric
Value	"rx_fifo_size" sets the receive FIFO size. The value is specified in Kb and, by default, is pre-calculated from the connected port’s size of the receive queue. Typically, you do not need to change the "rx_fifo_size" parameter. It is available for extended tuning and debugging purposes.

adapter_mode

<table>
<tr><td><b>Parameter</b></td><td>adapter_mode</td></tr>
<tr><td><b>Type</b></td><td>Text String</td></tr>
<tr><td><b>Value</b></td><td>
<p>Defines the link speed and the duplex settings of the virtual network adapter (except for DE602/DE602AA - see below).</p>
<p>The values are:</p>
<table>
<tr><th>Parameter</th><th>Description</th></tr>
<tr><td>"Auto"</td><td>Auto-negotiate (default)</td></tr>
<tr><td>"10BaseT-HD"</td><td>10Mbps half duplex</td></tr>
<tr><td>"10BaseT-FD"</td><td>10Mbps full duplex</td></tr>
<tr><td>"100BaseT-HD"</td><td>100Mbps half duplex</td></tr>
<tr><td>"100BaseT-FD"</td><td>100Mbps full duplex</td></tr>
</table>
<p>Defines the link speed and the duplex settings of the virtual network adapter (except for DE602/DE602AA - see below).</p>
<p>The values are:</p>
<p><u>Example</u>:</p>
<pre>set EWA adapter_mode="100BaseT-HD"</pre>
</td></tr>
</table>

Example

```
load packet_port/chnetwrk EWA0 interface = "eth1"
set EWA interface = EWA0
set EWA station_address="0C:FE:35:AA:67:3B"
```

Back to Table of Contents

DE602 and DE602AA network adapters link speed and duplex settings

Regardless of the "adapter_mode" setting in CHARON-AXP configuration file (see above), DE602 and DE602AA network adapters remains in "Auto-negotiation" mode, since the EIDRIVER of OpenVMS checks for EIx0_MODE environment variable when configuring the network card.

So mode propagation is implemented in CHARON-AXP via SRM console EIx0_MODE environment variable ("x" is A, B, C... depending on CHARON-AXP configuration), for example:

```
>>>help set

usage: set <variable-name> <value>
set <variable-name> ""

set eia0_mode { Twisted | Full | Fast | FastFD | Auto* }

>>>
```

⚠ The EIx0_MODE variable name is case insensitive, while its values are case sensitive! This is feature of OpenVMS EIDRIVER.

The values are:

Parameter	Description
"Auto"	Auto-negotiate (default)
"Twisted"	10Mbps half duplex
"Full"	10Mbps full duplex
"Fast"	100Mbps half duplex
"FastFD"	100Mbps full duplex

Example:

```
>>>set eia0_mode FastFD
```

Back to Table of Contents

Packet Port

The CHARON-specific "packet_port" interface establishes a connection between an Ethernet adapter in the Linux host system and a network adapter in the virtual HP Alpha system.

For every virtual adapter instance loaded, one dedicated host Ethernet physical adapter is required.

To create instances of the "packet_port", use the “load” command in the configuration file as follows:

```
load packet_port/chnetwrk <instance-name>
```

Example:

```
load packet_port/chnetwrk pp1
```

"packet_port" uses several configuration parameters to control its behavior.

interface

Parameter	interface
Type	Text string

Value	This parameter identifies an Ethernet adapter of the host system dedicated to CHARON-AXP.
	Syntax:
	`set <name> interface="<adapter>"`
	Example:
	`set pp1 interface="eth1"`

port_enable_mac_addr_change

Parameter	port_enable_mac_addr_change
Type	Boolean
Value	If "true" is specified (default value), CHARON-AXP sets the appropriate Ethernet address automatically. If "false" is specified, set the Ethernet address manually. Example: `set pp1 port_enable_mac_addr_change=false`

port_retry_on_tx

Parameter	port_retry_on_tx
Type	Numeric
Value	The "port_retry_on_tx" parameter controls the number of times a port will attempt to transmit a packet before giving up. By default, the value is 3. Increasing this value may introduce problems in carrier loss logic, because not all NIC drivers support a carrier status query. Typically, you do not need to increase the value. Example: `set pp1 port_retry_on_tx=8`

port_pending_rx_number

Parameter	port_pending_rx_number
Type	Numeric
Value	The "port_pending_rx_number" parameter sets the number of pending receive buffers. The default value is 63. The maximum value allowed is 195. You may want to increase the "port_pending_rx_number" when you have very busy networking and experience problems like losing connections not related to the carrier loss. Typically, you do not need to change this parameter. Example: `set pp1 port_pending_rx_number=128`

port_pending_tx_number

Parameter	port_pending_tx_number
Type	Numeric
Value	The "port_pending_tx_number" parameter sets the number of buffers the port uses to transmit. The default value is 62. You may want to increase the "port_pending_tx_number" value if the log file indicates dropped TX packets due to TX queue overflow. Typically, you do not need to change this parameter. Example: `set pp1 port_pending_tx_number=128`

Example

```
load DE500BA/dec21x4x EWA
load packet_port/chnetwrk pp1 interface="eth1"
set EWA interface=pp1
```

CHARON-AXP supports VLAN adapters. If for some reasons you are going to use them, proceed with their installation and configuration according to the network adapter's vendor's User's Guide and then use the resulting VLAN interface the same way as the regular network interface.

Back to Table of Contents

Sample configuration files

Contents

HP AlphaServer 800 configuration file

```
#
# Copyright (C) 1999-2014 STROMASYS
# All rights reserved.
#
# The software contained on this media is proprietary to and embodies
# the confidential technology of STROMASYS. Possession, use, duplication,
# or dissemination of the software and media is authorized only pursuant
# to a valid written license from STROMASYS.
#
#============================================================================
#
# Sample configuration file for AlphaServer 800 machines.
#
#----------------------------------------------------------------------------

set session hw_model = AlphaServer_800

#============================================================================
#
# Choose a name for the instance, if needed, to differentiate it among other
# instances running on the same host.
#
#----------------------------------------------------------------------------

#set session configuration_name = AlphaServer_800

#============================================================================
#
# Use the following commands to disable the rotating LOG files and enable
# a single LOG file. Select either append or overwrite (for each time the
# instance starts) and specify desired log path and file name.
#
#----------------------------------------------------------------------------

#set session log_method = append
#set session log_method = overwrite
#set session log = AlphaServer_800.log

#============================================================================
#
# Overrides system assigned process's CPU affinity. The session changes
# the process's CPU affinity to the one specified.
#
#----------------------------------------------------------------------------

#set session affinity="0, 1, 2, 3"

#============================================================================
#
# The 'n_of_io_cpus' option overrides number of host CPU cores reserved for
# I/O processing. If omitted the session reserves 33% of available host CPU
# cores for I/O processing. Note that total amount of available host CPU
# cores is determined based on process's CPU affinity.
#
#----------------------------------------------------------------------------

#set session n_of_io_cpus=1
#set session n_of_io_cpus=2
#set session n_of_io_cpus=...

#============================================================================
#
# AlphaServer 800 5/333
#
#----------------------------------------------------------------------------
```

```
set ace cpu_architecture = EV56
set rom dsrdb[0] = 1310 system_name = "AlphaServer 800 5/333"

#============================================================================
#
# AlphaServer 800 5/400
#
#----------------------------------------------------------------------------

#set ace cpu_architecture = EV56
#set rom dsrdb[0] = 1584 system_name = "AlphaServer 800 5/400"

#============================================================================
#
# AlphaServer 800 5/500
#
#----------------------------------------------------------------------------

#set ace cpu_architecture = EV56
#set rom dsrdb[0] = 1585 system_name = "AlphaServer 800 5/500"

#============================================================================
#
# Override default System Serial Number.
#
#----------------------------------------------------------------------------

#set rom system_serial_number = SN01234567

#============================================================================
#
# Specify size of RAM from 256MB up to 8192MB (8GB) in 256MB extents.
#
#----------------------------------------------------------------------------

#set ram size=256
#set ram size=512
#set ram size=1024
#set ram size=4096
#set ram size=8192

#============================================================================
#
# Uncomment to allow the virtual SRM console environment be preserved across
# emulator restarts.
#
#----------------------------------------------------------------------------

#set rom container="AlphaServer_800.bin"

#============================================================================
#
# Uncomment to allow saving CMOS NVRAM content, so that to preserve
# Time & Date information.
#
#----------------------------------------------------------------------------

#set toy container="AlphaServer_800.dat"

#============================================================================
#
# Select the connection method for the console serial line OPA0.
#
#----------------------------------------------------------------------------

#load physical_serial_line OPA0 line="/dev/ttyN"
#load virtual_serial_line OPA0 port=10003
load operator_console OPA0

#----------------------------------------------------------------------------
#
# Uncomment to allow 'F6' to terminate the running emulator.
```

```
#
#----------------------------------------------------------------------------

#set OPA0 stop_on = F6

#============================================================================
#
# Select connection for the serial line TTA0.
#
#----------------------------------------------------------------------------

#load physical_serial_line TTA0 line="/dev/ttyN"
#load virtual_serial_line TTA0 port=10000

#============================================================================
#
# If TTA0 is loaded, attach it to secondary serial line controller COM2.
#
#----------------------------------------------------------------------------

#set COM2 line=TTA0

#============================================================================
#
# Improve granularity of emulated AXP timer.
#
#----------------------------------------------------------------------------

#set eisa clock_period=1000

#============================================================================
#
# Load optional DE500BA PCI Ethernet Adapter (EWA).
#
#----------------------------------------------------------------------------

#load DE500BA/dec21x4x EWA interface=EWA0
#load packet_port/chnetwrk EWA0 interface="eth0"

#============================================================================
#
# The AlphaServer 800 contains built-in PCI SCSI adapter called PKA within
# the configuration file.
#
#----------------------------------------------------------------------------
#
# Uncomment to connect the emulator's DKA0 to the disk image.
#
#----------------------------------------------------------------------------

#set PKA container[0] = "<file-name>.vdisk"

#============================================================================
#
# Uncomment to connect the emulator's DKA100 to a host disk drive.
#
#----------------------------------------------------------------------------

#set PKA container[100] = "/dev/sd<L>"

#============================================================================
#
# Uncomment to connect the emulator's GKA200 to an unknown SCSI device.
#
#----------------------------------------------------------------------------

#set PKA container[200] = "/dev/sg<N>"

#============================================================================
#
# Uncomment to connect the emulator's DKA300 to the host's CD/DVD-ROM drive.
#
```

```
# Device name may be different depending on particular version of host
# operating system. Choose one which suits best.
#
#----------------------------------------------------------------------------

#set PKA container[300] = "/dev/cdrom"
#set PKA container[300] = "/dev/cdrom1"
#set PKA container[300] = "/dev/cdrom<N>"
#set PKA container[300] = "/dev/sr0"
#set PKA container[300] = "/dev/sr<N>"

#============================================================================
#
# Uncomment to connect the emulator's DKA400 to an .ISO file (CD/DVD image).
#
#----------------------------------------------------------------------------

#set PKA container[400] = "<file-name>.iso"

#============================================================================
#
# Uncomment to connect the emulator's MKA500 to the host's SCSI tape drive.
#
#----------------------------------------------------------------------------

#set PKA container[500] = "/dev/sg<N>"

#============================================================================
#
# Uncomment to connect the emulator's MKA600 to a .VTAPE file (tape image).
#
#----------------------------------------------------------------------------

#set PKA container[600] = "<file-name>.vtape"

#============================================================================
#
# Uncomment to enable emulation of secondary DEC-KZPBA SCSI controller (PKB).
#
#----------------------------------------------------------------------------

#load KZPBA PKB scsi_id = 7

#============================================================================
#
# Uncomment to connect the emulator's DKB0 to the disk image.
#
#----------------------------------------------------------------------------

#set PKB container[0] = "<file-name>.vdisk"

#============================================================================
#
# Uncomment to connect the emulator's DKB100 to a host disk drive.
#
#----------------------------------------------------------------------------

#set PKB container[100] = "/dev/sd<L>"

#============================================================================
#
# Uncomment to connect the emulator's GKB200 to an unknown SCSI device.
#
#----------------------------------------------------------------------------

#set PKB container[200] = "/dev/sg<N>"

#============================================================================
#
# Uncomment to connect the emulator's DKB300 to the host's CD/DVD-ROM drive.
#
# Device name may be different depending on particular version of host
# operating system. Choose one which suits best.
#
#----------------------------------------------------------------------------

#set PKB container[300] = "/dev/cdrom"
#set PKB container[300] = "/dev/cdrom1"
#set PKB container[300] = "/dev/cdrom<N>"
#set PKB container[300] = "/dev/sr0"
#set PKB container[300] = "/dev/sr<N>"

#============================================================================
#
# Uncomment to connect the emulator's DKB400 to an .ISO file (CD/DVD image).
#
#----------------------------------------------------------------------------

#set PKB container[400] = "<file-name>.iso"

#============================================================================
#
# Uncomment to connect the emulator's MKB500 to the host's SCSI tape drive.
#
#----------------------------------------------------------------------------

#set PKB container[500] = "/dev/sg<N>"

#============================================================================
#
# Uncomment to connect the emulator's MKB600 to a .VTAPE file (tape image).
#
#----------------------------------------------------------------------------

#set PKB container[600] = "<file-name>.vtape"

#============================================================================
#
# Uncomment to enable emulation of DEC-KGPSA-CA PCI FC Adapter.
#
#----------------------------------------------------------------------------

#load KGPSA FGA

#============================================================================
#
# Uncomment to connect the emulator's $1$DGA0 to the disk image.
#
#----------------------------------------------------------------------------

#set FGA container[0] = "<file-name>.vdisk"

#============================================================================
#
# Uncomment to connect the emulator's $1$DGA100 to a host disk drive.
#
#----------------------------------------------------------------------------

#set FGA container[100] = "/dev/sd<L>"

#============================================================================
#
# Uncomment to enable emulation of secondary DEC-KGPSA-CA PCI FC Adapter.
#
#----------------------------------------------------------------------------

#load KGPSA FGB

#============================================================================
#
# Uncomment to enable PCI Pass Through access to physical EMULEX LP FC HBA,
# use two adapters to provide multipath with failover.
#
#----------------------------------------------------------------------------
```

```
#set FGA host_bus_location = "/dev/kgpsaX"
#set FGB host_bus_location = "/dev/kgpsaY"

# this is the end of the configuration file #################################
```

Back to Top

HP AlphaServer 4000 configuration file

```
#
# Copyright (C) 1999-2014 STROMASYS
# All rights reserved.
#
# The software contained on this media is proprietary to and embodies
# the confidential technology of STROMASYS. Possession, use, duplication,
# or dissemination of the software and media is authorized only pursuant
# to a valid written license from STROMASYS.
#
#============================================================================
#
# Sample configuration file for AlphaServer 4000 machines.
#
#----------------------------------------------------------------------------

set session hw_model = AlphaServer_4000

#============================================================================
#
# Choose a name for the instance, if needed, to differentiate it among other
# instances running on the same host.
#
#----------------------------------------------------------------------------

#set session configuration_name = AlphaServer_4000

#============================================================================
#
# Use the following commands to disable the rotating LOG files and enable
# a single LOG file. Select either append or overwrite (for each time the
# instance starts) and specify desired log path and file name.
#
#----------------------------------------------------------------------------

#set session log_method = append
#set session log_method = overwrite
#set session log = AlphaServer_4000.log

#============================================================================
#
# Overrides system assigned process's CPU affinity. The session changes
# the process's CPU affinity to the one specified.
#
#----------------------------------------------------------------------------

#set session affinity="0, 1, 2, 3"

#============================================================================
#
# The 'n_of_io_cpus' option overrides number of host CPU cores reserved for
# I/O processing. If omitted the session reserves 33% of available host CPU
# cores for I/O processing. Note that total amount of available host CPU
# cores is determined based on process's CPU affinity.
#
#----------------------------------------------------------------------------

#set session n_of_io_cpus=1
#set session n_of_io_cpus=2
#set session n_of_io_cpus=...

#============================================================================
#
# AlphaServer 4000 5/300
#
#----------------------------------------------------------------------------
```

```
#set ace cpu_architecture = EV5
#set rom dsrdb[0] = 1450 system_name = "AlphaServer 4000 5/300"

#=============================================================================
#
# AlphaServer 4000 5/400
#
#-----------------------------------------------------------------------------

set ace cpu_architecture = EV56
set rom dsrdb[0] = 1454 system_name = "AlphaServer 4000 5/400"

#=============================================================================
#
# The 'n_of_cpus' option reduces number of emulated Alpha CPUs in the
# configuration.
#
#-----------------------------------------------------------------------------

#set session n_of_cpus=1

#=============================================================================
#
# Override default System Serial Number.
#
#-----------------------------------------------------------------------------

#set rom system_serial_number = SN01234567

#=============================================================================
#
# Specify size of RAM from 256MB up to 32768MB (32GB) in 256MB extents.
#
#-----------------------------------------------------------------------------

#set ram size=256
#set ram size=512
#set ram size=1024
#set ram size=4096
#set ram size=32768

#=============================================================================
#
# Uncomment to allow the SRM console environment be preserved across
# emulator restarts.
#
#-----------------------------------------------------------------------------

#set rom container="AlphaServer_4000.bin"

#=============================================================================
#
# Uncomment to allow saving CMOS NVRAM content, so that to preserve
# Time & Date information.
#
#-----------------------------------------------------------------------------

#set toy container="AlphaServer_4000.dat"

#=============================================================================
#
# Select the connection method for the console serial line OPA0.
#
#-----------------------------------------------------------------------------

#load physical_serial_line OPA0 line="/dev/ttyN"
#load virtual_serial_line OPA0 port=10003
load operator_console OPA0

#-----------------------------------------------------------------------------
#
# Uncomment to allow 'F6' to terminate the running emulator.
```

```
#
#----------------------------------------------------------------------------

#set OPA0 stop_on = F6

#============================================================================
#
# Improve granularity of emulated AXP timer.
#
#----------------------------------------------------------------------------

#set eisa clock_period=1000

#============================================================================
#
# Load optional DE500BA PCI Ethernet Adapter (EWA).
#
#----------------------------------------------------------------------------

#load DE500BA/dec21x4x EWA interface=EWA0
#load packet_port/chnetwrk EWA0 interface="eth0"

#============================================================================
#
# Load another optional DE500BA PCI Ethernet Adapter (EWB).
#
#----------------------------------------------------------------------------

#load DE500BA/dec21x4x EWB interface=EWB0
#load packet_port/chnetwrk EWB0 interface="eth1"

#============================================================================
#
# Load another optional DE500BA PCI Ethernet Adapter (EWC).
#
#----------------------------------------------------------------------------

#load DE500BA/dec21x4x EWC interface=EWC0
#load packet_port/chnetwrk EWC0 interface="eth2"

#============================================================================
#
# The AlphaServer 4000 contains built-in PCI SCSI adapter, called PKA within
# the configuration file.
#
#----------------------------------------------------------------------------
#
# Uncomment to connect the emulator's DKA0 to the disk image.
#
#----------------------------------------------------------------------------

#set PKA container[0] = "<file-name>.vdisk"

#============================================================================
#
# Uncomment to connect the emulator's DKA100 to a host disk drive.
#
#----------------------------------------------------------------------------

#set PKA container[100] = "/dev/sd<L>"

#============================================================================
#
# Uncomment to connect the emulator's GKA200 to an unknown SCSI device.
#
#----------------------------------------------------------------------------

#set PKA container[200] = "/dev/sg<N>"

#============================================================================
#
# Uncomment to connect the emulator's DKA300 to the host's CD/DVD-ROM drive.
```

```
#
# Device name may be different depending on particular version of host
# operating system. Choose one which suits best.
#
#-----------------------------------------------------------------------------

#set PKA container[300] = "/dev/cdrom"
#set PKA container[300] = "/dev/cdrom1"
#set PKA container[300] = "/dev/cdrom<N>"
#set PKA container[300] = "/dev/sr0"
#set PKA container[300] = "/dev/sr<N>"

#=============================================================================
#
# Uncomment to connect the emulator's DKA400 to an .ISO file (CD/DVD image).
#
#-----------------------------------------------------------------------------

#set PKA container[400] = "<file-name>.iso"

#=============================================================================
#
# Uncomment to connect the emulator's MKA500 to the host's SCSI tape drive.
#
#-----------------------------------------------------------------------------

#set PKA container[500] = "/dev/sg<N>"

#=============================================================================
#
# Uncomment to connect the emulator's MKA600 to a .VTAPE file (tape image).
#
#-----------------------------------------------------------------------------

#set PKA container[600] = "<file-name>.vtape"

#=============================================================================
#
# Uncomment to enable emulation of secondary DEC-KZPBA SCSI controller (PKB).
#
#-----------------------------------------------------------------------------

#load KZPBA PKB scsi_id = 7

#=============================================================================
#
# Uncomment to connect the emulator's DKB0 to the disk image.
#
#-----------------------------------------------------------------------------

#set PKB container[0] = "<file-name>.vdisk"

#=============================================================================
#
# Uncomment to connect the emulator's DKB100 to a host disk drive.
#
#-----------------------------------------------------------------------------

#set PKB container[100] = "/dev/sd<L>"

#=============================================================================
#
# Uncomment to connect the emulator's GKB200 to an unknown SCSI device.
#
#-----------------------------------------------------------------------------

#set PKB container[200] = "/dev/sg<N>"

#=============================================================================
#
# Uncomment to connect the emulator's DKB300 to the host's CD/DVD-ROM drive.
#
```

```
# Device name may be different depending on particular version of host
# operating system. Choose one which suits best.
#
#-----------------------------------------------------------------------------

#set PKB container[300] = "/dev/cdrom"
#set PKB container[300] = "/dev/cdrom1"
#set PKB container[300] = "/dev/cdrom<N>"
#set PKB container[300] = "/dev/sr0"
#set PKB container[300] = "/dev/sr<N>"

#=============================================================================
#
# Uncomment to connect the emulator's DKB400 to an .ISO file (CD/DVD image).
#
#-----------------------------------------------------------------------------

#set PKB container[400] = "<file-name>.iso"

#=============================================================================
#
# Uncomment to connect the emulator's MKB500 to the host's SCSI tape drive.
#
#-----------------------------------------------------------------------------

#set PKB container[500] = "/dev/sg<N>"

#=============================================================================
#
# Uncomment to connect the emulator's MKB600 to a .VTAPE file (tape image).
#
#-----------------------------------------------------------------------------

#set PKB container[600] = "<file-name>.vtape"

#=============================================================================
#
# Uncomment to enable emulation of DEC-KGPSA-CA PCI FC Adapter.
#
#-----------------------------------------------------------------------------

#load KGPSA FGA

#=============================================================================
#
# Uncomment to connect the emulator's $1$DGA0 to the disk image.
#
#-----------------------------------------------------------------------------

#set FGA container[0] = "<file-name>.vdisk"

#=============================================================================
#
# Uncomment to connect the emulator's $1$DGA100 to a host disk drive.
#
#-----------------------------------------------------------------------------

#set FGA container[100] = "/dev/sd<L>"

#=============================================================================
#
# Uncomment to enable emulation of secondary DEC-KGPSA-CA PCI FC Adapter.
#
#-----------------------------------------------------------------------------

#load KGPSA FGB

#=============================================================================
#
# Uncomment to enable PCI Pass Through access to physical EMULEX LP FC HBA,
# use two adapters to provide multipath with failover.
#
#----------------------------------------------------------------------------

#set FGA host_bus_location = "/dev/kgpsaX"
#set FGB host_bus_location = "/dev/kgpsaY"

# this is the end of the configuration file #################################
```

Back to Top

HP AlphaServer DS20 configuration file

```
#
# Copyright (C) 1999-2014 STROMASYS
# All rights reserved.
#
# The software contained on this media is proprietary to and embodies
# the confidential technology of STROMASYS. Possession, use, duplication,
# or dissemination of the software and media is authorized only pursuant
# to a valid written license from STROMASYS.
#
#============================================================================
#
# Sample configuration file for AlphaServer DS20 machines.
#
#----------------------------------------------------------------------------

set session hw_model = AlphaServer_DS20

#============================================================================
#
# Choose a name for the instance, if needed, to differentiate it among other
# instances running on the same host.
#
#----------------------------------------------------------------------------

#set session configuration_name = AlphaServer_DS20

#============================================================================
#
# Use the following commands to disable the rotating LOG files and enable
# a single LOG file. Select either append or overwrite (for each time the
# instance starts) and specify desired log path and file name.
#
#----------------------------------------------------------------------------

#set session log_method = append
#set session log_method = overwrite
#set session log = AlphaServer_DS20.log

#============================================================================
#
# Overrides system assigned process's CPU affinity. The session changes
# the process's CPU affinity to the one specified.
#
#----------------------------------------------------------------------------

#set session affinity="0, 1, 2, 3"

#============================================================================
#
# The 'n_of_io_cpus' option overrides number of host CPU cores reserved for
# I/O processing. If omitted the session reserves 33% of available host CPU
# cores for I/O processing. Note that total amount of available host CPU
# cores is determined based on process's CPU affinity.
#
#----------------------------------------------------------------------------

#set session n_of_io_cpus=1
#set session n_of_io_cpus=2
#set session n_of_io_cpus=...

#============================================================================
#
# AlphaServer DS20 6/500
#
#----------------------------------------------------------------------------
```

```
#set ace cpu_architecture = EV6
#set rom dsrdb[0] = 1920 system_name = "AlphaServer DS20 6/500"

#============================================================================
#
# AlphaServer DS20E 67/667
#
#----------------------------------------------------------------------------

set ace cpu_architecture = EV67
set rom dsrdb[0] = 1940 system_name = "AlphaServer DS20E 67/667"

#============================================================================
#
# The 'n_of_cpus' option reduces number of emulated Alpha CPUs in the
# configuration.
#
#----------------------------------------------------------------------------

#set session n_of_cpus=1

#============================================================================
#
# Override default System Serial Number.
#
#----------------------------------------------------------------------------

#set rom system_serial_number = SN01234567

#============================================================================
#
# Specify size of RAM from 256MB up to 32768MB (32GB) in 256MB extents.
#
#----------------------------------------------------------------------------

#set ram size=256
#set ram size=512
#set ram size=1024
#set ram size=4096
#set ram size=32768

#============================================================================
#
# Uncomment to allow the virtual SRM console environment be preserved across
# emulator restarts.
#
#----------------------------------------------------------------------------

#set rom container="AlphaServer_DS20.bin"

#============================================================================
#
# Uncomment to allow saving CMOS NVRAM content, so that to preserve
# Time & Date information.
#
#----------------------------------------------------------------------------

#set toy container="AlphaServer_DS20.dat"

#============================================================================
#
# Select the connection method for the console serial line OPA0.
#
#----------------------------------------------------------------------------

#load physical_serial_line OPA0 line="/dev/ttyN"
#load virtual_serial_line OPA0 port=10003
load operator_console OPA0

#----------------------------------------------------------------------------
#
# Uncomment to allow 'F6' to terminate the running emulator.
```

```
#
#----------------------------------------------------------------------------

#set OPA0 stop_on = F6

#============================================================================
#
# Improve granularity of emulated AXP timer.
#
#----------------------------------------------------------------------------

#set isa clock_period=1000

#============================================================================
#
# Uncomment to connect the emulator's DQA0 to host's ATAPI CD/DVD-ROM drive.
#
#----------------------------------------------------------------------------

#set ide container = "/dev/sg<N>"

#============================================================================
#
# Load optional DE500BA PCI Ethernet Adapter (EWA).
#
#----------------------------------------------------------------------------

#load DE500BA/dec21x4x EWA interface=EWA0
#load packet_port/chnetwrk EWA0 interface="eth0"

#============================================================================
#
# Load another optional DE500BA PCI Ethernet Adapter (EWB).
#
#----------------------------------------------------------------------------

#load DE500BA/dec21x4x EWB interface=EWB0
#load packet_port/chnetwrk EWB0 interface="eth1"

#============================================================================
#
# Load another optional DE500BA PCI Ethernet Adapter (EWC).
#
#----------------------------------------------------------------------------

#load DE500BA/dec21x4x EWC interface=EWC0
#load packet_port/chnetwrk EWC0 interface="eth2"

#============================================================================
#
# The AlphaServer DS20 contains two built-in PCI SCSI adapters called PKA and
# PKB within the configuration file.
#
#----------------------------------------------------------------------------
#
# Uncomment to connect the emulator's DKA0 to the disk image.
#
#----------------------------------------------------------------------------

#set PKA container[0] = "<file-name>.vdisk"

#============================================================================
#
# Uncomment to connect the emulator's DKA100 to a host disk drive.
#
#----------------------------------------------------------------------------

#set PKA container[100] = "/dev/sd<L>"

#============================================================================
#
# Uncomment to connect the emulator's GKA200 to an unknown SCSI device.
```

```
#
#-----------------------------------------------------------------------------

#set PKA container[200] = "/dev/sg<N>"

#=============================================================================
#
# Uncomment to connect the emulator's DKA300 to the host's CD/DVD-ROM drive.
#
# Device name may be different depending on particular version of host
# operating system. Choose one which suits best.
#
#-----------------------------------------------------------------------------

#set PKA container[300] = "/dev/cdrom"
#set PKA container[300] = "/dev/cdrom1"
#set PKA container[300] = "/dev/cdrom<N>"
#set PKA container[300] = "/dev/sr0"
#set PKA container[300] = "/dev/sr<N>"

#=============================================================================
#
# Uncomment to connect the emulator's DKA400 to an .ISO file (CD/DVD image).
#
#-----------------------------------------------------------------------------

#set PKA container[400] = "<file-name>.iso"

#=============================================================================
#
# Uncomment to connect the emulator's MKA500 to the host's SCSI tape drive.
#
#-----------------------------------------------------------------------------

#set PKA container[500] = "/dev/sg<N>"

#=============================================================================
#
# Uncomment to connect the emulator's MKA600 to a .VTAPE file (tape image).
#
#-----------------------------------------------------------------------------

#set PKA container[600] = "<file-name>.vtape"

#=============================================================================
#
# Uncomment to connect the emulator's DKB0 to the disk image.
#
#-----------------------------------------------------------------------------

#set PKB container[0] = "<file-name>.vdisk"

#=============================================================================
#
# Uncomment to connect the emulator's DKB100 to a host disk drive.
#
#-----------------------------------------------------------------------------

#set PKB container[100] = "/dev/sd<L>"

#=============================================================================
#
# Uncomment to connect the emulator's GKB200 to an unknown SCSI device.
#
#-----------------------------------------------------------------------------

#set PKB container[200] = "/dev/sg<N>"

#=============================================================================
#
# Uncomment to connect the emulator's DKB300 to the host's CD/DVD-ROM drive.
#
```

```
# Device name may be different depending on particular version of host
# operating system. Choose one which suits best.
#
#----------------------------------------------------------------------------

#set PKB container[300] = "/dev/cdrom"
#set PKB container[300] = "/dev/cdrom1"
#set PKB container[300] = "/dev/cdrom<N>"
#set PKB container[300] = "/dev/sr0"
#set PKB container[300] = "/dev/sr<N>"

#============================================================================
#
# Uncomment to connect the emulator's DKB400 to an .ISO file (CD/DVD image).
#
#----------------------------------------------------------------------------

#set PKB container[400] = "<file-name>.iso"

#============================================================================
#
# Uncomment to connect the emulator's MKB500 to the host's SCSI tape drive.
#
#----------------------------------------------------------------------------

#set PKB container[500] = "/dev/sg<N>"

#============================================================================
#
# Uncomment to connect the emulator's MKB600 to a .VTAPE file (tape image).
#
#----------------------------------------------------------------------------

#set PKB container[600] = "<file-name>.vtape"

#============================================================================
#
# Uncomment to enable emulation of DEC-KGPSA-CA PCI FC Adapter.
#
#----------------------------------------------------------------------------

#load KGPSA FGA

#============================================================================
#
# Uncomment to connect the emulator's $1$DGA0 to the disk image.
#
#----------------------------------------------------------------------------

#set FGA container[0] = "<file-name>.vdisk"

#============================================================================
#
# Uncomment to connect the emulator's $1$DGA100 to a host disk drive.
#
#----------------------------------------------------------------------------

#set FGA container[100] = "/dev/sd<L>"

#============================================================================
#
# Uncomment to enable emulation of secondary DEC-KGPSA-CA PCI FC Adapter.
#
#----------------------------------------------------------------------------

#load KGPSA FGB

#============================================================================
#
# Uncomment to enable PCI Pass Through access to physical EMULEX LP FC HBA,
# use two adapters to provide multipath with failover.
#
#----------------------------------------------------------------------------

#set FGA host_bus_location = "/dev/kgpsaX"
#set FGB host_bus_location = "/dev/kgpsaY"

# this is the end of the configuration file #################################
```

Back to Top

HP AlphaServer ES40 configuration file

```
#
# Copyright (C) 1999-2014 STROMASYS
# All rights reserved.
#
# The software contained on this media is proprietary to and embodies
# the confidential technology of STROMASYS. Possession, use, duplication,
# or dissemination of the software and media is authorized only pursuant
# to a valid written license from STROMASYS.
#
#============================================================================
#
# Sample configuration file for AlphaServer ES40 machines.
#
#----------------------------------------------------------------------------

set session hw_model = AlphaServer_ES40

#============================================================================
#
# Choose a name for the instance, if needed, to differentiate it among other
# instances running on the same host.
#
#----------------------------------------------------------------------------

#set session configuration_name = AlphaServer_ES40

#============================================================================
#
# Use the following commands to disable the rotating LOG files and enable
# a single LOG file. Select either append or overwrite (for each time the
# instance starts) and specify desired log path and file name.
#
#----------------------------------------------------------------------------

#set session log_method = append
#set session log_method = overwrite
#set session log = AlphaServer_ES40.log

#============================================================================
#
# Overrides system assigned process's CPU affinity. The session changes
# the process's CPU affinity to the one specified.
#
#----------------------------------------------------------------------------

#set session affinity="0, 1, 2, 3"

#============================================================================
#
# The 'n_of_io_cpus' option overrides number of host CPU cores reserved for
# I/O processing. If omitted the session reserves 33% of available host CPU
# cores for I/O processing. Note that total amount of available host CPU
# cores is determined based on process's CPU affinity.
#
#----------------------------------------------------------------------------

#set session n_of_io_cpus=1
#set session n_of_io_cpus=2
#set session n_of_io_cpus=...

#============================================================================
#
# AlphaServer ES40 6/500
#
#----------------------------------------------------------------------------
```

```
#set ace cpu_architecture = EV6
#set rom dsrdb[0] = 1816 system_name = "AlphaServer ES40 6/500"
#set rom version[1] = 1.98-4 version[2] = 1.92-5

#============================================================================
#
# AlphaServer ES40 6/667
#
#----------------------------------------------------------------------------

set ace cpu_architecture = EV67
set rom dsrdb[0] = 1820 system_name = "AlphaServer ES40 6/667"

#============================================================================
#
# The 'n_of_cpus' option reduces number of emulated Alpha CPUs in the
# configuration.
#
#----------------------------------------------------------------------------

#set session n_of_cpus=1
#set session n_of_cpus=2
#set session n_of_cpus=3

#============================================================================
#
# Override default System Serial Number.
#
#----------------------------------------------------------------------------

#set rom system_serial_number = SN01234567

#============================================================================
#
# Specify size of RAM from 256MB up to 32768MB (32GB) in 256MB extents.
#
#----------------------------------------------------------------------------

#set ram size=256
#set ram size=512
#set ram size=1024
#set ram size=4096
#set ram size=32768

#============================================================================
#
# Uncomment to allow the SRM console environment be preserved across
# emulator restarts.
#
#----------------------------------------------------------------------------

#set rom container="clipper.bin"

#============================================================================
#
# Uncomment to allow saving CMOS NVRAM content, so that to preserve
# Time & Date information.
#
#----------------------------------------------------------------------------

#set toy container="clipper.dat"

#============================================================================
#
# Select the connection method for the console serial line OPA0.
#
#----------------------------------------------------------------------------

#load physical_serial_line OPA0 line="/dev/ttyN"
#load virtual_serial_line OPA0 port=10003
load operator_console OPA0
```

```
#----------------------------------------------------------------------------
#
# Uncomment to allow 'F6' to terminate the running emulator.
#
#----------------------------------------------------------------------------

#set OPA0 stop_on = F6

#============================================================================
#
# Improve granularity of emulated AXP timer.
#
#----------------------------------------------------------------------------

#set isa clock_period=1000

#============================================================================
#
# Uncomment to connect the emulator's DQA0 to host's ATAPI CD/DVD-ROM drive.
#
#----------------------------------------------------------------------------

#set ide container = "/dev/sg<N>"

#============================================================================
#
# Load optional DE500BA PCI Ethernet Adapter (EWA).
#
#----------------------------------------------------------------------------

#load DE500BA/dec21x4x EWA interface=EWA0
#load packet_port/chnetwrk EWA0 interface="eth0"

#============================================================================
#
# Load another optional DE500BA PCI Ethernet Adapter (EWB).
#
#----------------------------------------------------------------------------

#load DE500BA/dec21x4x EWB interface=EWB0
#load packet_port/chnetwrk EWB0 interface="eth1"

#============================================================================
#
# Load another optional DE500BA PCI Ethernet Adapter (EWC).
#
#----------------------------------------------------------------------------

#load DE500BA/dec21x4x EWC interface=EWC0
#load packet_port/chnetwrk EWC0 interface="eth2"

#============================================================================
#
# Uncomment to enable emulation of DEC-KZPBA SCSI controller.
#
#----------------------------------------------------------------------------

#load KZPBA PKA scsi_id = 7

#============================================================================
#
# Uncomment to connect the emulator's DKA0 to the disk image.
#
#----------------------------------------------------------------------------

#set PKA container[0] = "<file-name>.vdisk"

#============================================================================
#
# Uncomment to connect the emulator's DKA100 to a host disk drive.
#
#----------------------------------------------------------------------------
```

```
#set PKA container[100] = "/dev/sd<L>"

#============================================================================
#
# Uncomment to connect the emulator's GKA200 to an unknown SCSI device.
#
#----------------------------------------------------------------------------

#set PKA container[200] = "/dev/sg<N>"

#============================================================================
#
# Uncomment to connect the emulator's DKA300 to the host's CD/DVD-ROM drive.
#
# Device name may be different depending on particular version of host
# operating system. Choose one which suits best.
#
#----------------------------------------------------------------------------

#set PKA container[300] = "/dev/cdrom"
#set PKA container[300] = "/dev/cdrom1"
#set PKA container[300] = "/dev/cdrom<N>"
#set PKA container[300] = "/dev/sr0"
#set PKA container[300] = "/dev/sr<N>"

#============================================================================
#
# Uncomment to connect the emulator's DKA400 to an .ISO file (CD/DVD image).
#
#----------------------------------------------------------------------------

#set PKA container[400] = "<file-name>.iso"

#============================================================================
#
# Uncomment to connect the emulator's MKA500 to the host's SCSI tape drive.
#
#----------------------------------------------------------------------------

#set PKA container[500] = "/dev/sg<N>"

#============================================================================
#
# Uncomment to connect the emulator's MKA600 to a .VTAPE file (tape image).
#
#----------------------------------------------------------------------------

#set PKA container[600] = "<file-name>.vtape"

#============================================================================
#
# Uncomment to enable emulation of secondary DEC-KZPBA SCSI controller (PKB).
#
#----------------------------------------------------------------------------

#load KZPBA PKB scsi_id = 7

#============================================================================
#
# Uncomment to connect the emulator's DKB0 to the disk image.
#
#----------------------------------------------------------------------------

#set PKB container[0] = "<file-name>.vdisk"

#============================================================================
#
# Uncomment to connect the emulator's DKB100 to a host disk drive.
#
#----------------------------------------------------------------------------

#set PKB container[100] = "/dev/sd<L>"
```

```
#============================================================================
#
# Uncomment to connect the emulator's GKB200 to an unknown SCSI device.
#
#----------------------------------------------------------------------------

#set PKB container[200] = "/dev/sg<N>"

#============================================================================
#
# Uncomment to connect the emulator's DKB300 to the host's CD/DVD-ROM drive.
#
# Device name may be different depending on particular version of host
# operating system. Choose one which suits best.
#
#----------------------------------------------------------------------------

#set PKB container[300] = "/dev/cdrom"
#set PKB container[300] = "/dev/cdrom1"
#set PKB container[300] = "/dev/cdrom<N>"
#set PKB container[300] = "/dev/sr0"
#set PKB container[300] = "/dev/sr<N>"

#============================================================================
#
# Uncomment to connect the emulator's DKB400 to an .ISO file (CD/DVD image).
#
#----------------------------------------------------------------------------

#set PKB container[400] = "<file-name>.iso"

#============================================================================
#
# Uncomment to connect the emulator's MKB500 to the host's SCSI tape drive.
#
#----------------------------------------------------------------------------

#set PKB container[500] = "/dev/sg<N>"

#============================================================================
#
# Uncomment to connect the emulator's MKB600 to a .VTAPE file (tape image).
#
#----------------------------------------------------------------------------

#set PKB container[600] = "<file-name>.vtape"

#============================================================================
#
# Uncomment to enable emulation of DEC-KGPSA-CA PCI FC Adapter.
#
#----------------------------------------------------------------------------

#load KGPSA FGA

#============================================================================
#
# Uncomment to connect the emulator's $1$DGA0 to the disk image.
#
#----------------------------------------------------------------------------

#set FGA container[0] = "<file-name>.vdisk"

#============================================================================
#
# Uncomment to connect the emulator's $1$DGA100 to a host disk drive.
#
#----------------------------------------------------------------------------

#set FGA container[100] = "/dev/sd<L>"

#============================================================================
#
# Uncomment to enable emulation of secondary DEC-KGPSA-CA PCI FC Adapter.
#
#----------------------------------------------------------------------------

#load KGPSA FGB

#============================================================================
#
# Uncomment to enable PCI Pass Through access to physical EMULEX LP FC HBA,
# use two adapters to provide multipath with failover.
#
#----------------------------------------------------------------------------

#set FGA host_bus_location = "/dev/kgpsaX"
#set FGB host_bus_location = "/dev/kgpsaY"

# this is the end of the configuration file #################################
```

Back to Top

HP AlphaServer GS80 configuration file

```
#
# Copyright (C) 1999-2014 STROMASYS
# All rights reserved.
#
# The software contained on this media is proprietary to and embodies
# the confidential technology of STROMASYS. Possession, use, duplication,
# or dissemination of the software and media is authorized only pursuant
# to a valid written license from STROMASYS.
#
#============================================================================
#
# Sample configuration file for AlphaServer GS80 machines.
#
#----------------------------------------------------------------------------

set session hw_model = AlphaServer_GS80

#============================================================================
#
# Choose a name for the instance, if needed, to differentiate it among other
# instances running on the same host.
#
#----------------------------------------------------------------------------

#set session configuration_name = AlphaServer_GS80

#============================================================================
#
# Use the following commands to disable the rotating LOG files and enable
# a single LOG file. Select either append or overwrite (for each time the
# instance starts) and specify desired log path and file name.
#
#----------------------------------------------------------------------------

#set session log_method = append
#set session log_method = overwrite
#set session log = AlphaServer_GS80.log

#============================================================================
#
# Overrides system assigned process's CPU affinity. The session changes
# the process's CPU affinity to the one specified.
#
#----------------------------------------------------------------------------

#set session affinity="0, 1, 2, 3"

#============================================================================
#
# The 'n_of_io_cpus' option overrides number of host CPU cores reserved for
# I/O processing. If omitted the session reserves 33% of available host CPU
# cores for I/O processing. Note that total amount of available host CPU
# cores is determined based on process's CPU affinity.
#
#----------------------------------------------------------------------------

#set session n_of_io_cpus=1
#set session n_of_io_cpus=2
#set session n_of_io_cpus=...

#============================================================================
#
# AlphaServer GS80 67/728
#
#----------------------------------------------------------------------------
```

```
set ace cpu_architecture = EV67
set rom dsrdb[0] = 1967 system_name = "AlphaServer GS80 67/728"

#============================================================================
#
# "Turn" it into 8 CPU capable AlphaServer GS1280. Make sure to has even
# number of CPUs (2, 4, 6, ... 8). This is to reflect that fact that on real
# MARVELL platform CPUs are plugged in pairs (dual-cpu boards).
#
#----------------------------------------------------------------------------

#set ace cpu_architecture = EV67
#set rom system_name = "AlphaServer GS1280"
#set rom dsrdb[0] = 2038 dsrdb[4] = 3050 dsrdb[11] = 1300 dsrdb[12] = 1300

#============================================================================
#
# The 'n_of_cpus' option reduces number of emulated Alpha CPUs in the
# configuration.
#
#----------------------------------------------------------------------------

#set session n_of_cpus=1
#set session n_of_cpus=2
#set session n_of_cpus=...
#set session n_of_cpus=7

#============================================================================
#
# Override default System Serial Number.
#
#----------------------------------------------------------------------------

#set rom system_serial_number = SN01234567

#============================================================================
#
# Specify size of RAM from 256MB up to 65536MB (64GB) in 256MB extents.
#
#----------------------------------------------------------------------------

#set ram size=256
#set ram size=512
#set ram size=1024
#set ram size=4096
#set ram size=65536

#============================================================================
#
# Uncomment to allow the SRM console environment be preserved across
# emulator restarts.
#
#----------------------------------------------------------------------------

#set rom container="AlphaServer_GS80.bin"

#============================================================================
#
# Uncomment to allow saving CMOS NVRAM content, so that to preserve
# Time & Date information.
#
#----------------------------------------------------------------------------

#set toy container="AlphaServer_GS80.dat"

#============================================================================
#
# Select the connection method for the console serial line OPA0.
#
#----------------------------------------------------------------------------
```

```
#load physical_serial_line OPA0 line="/dev/ttyN"
#load virtual_serial_line OPA0 port=10003
load operator_console OPA0

#----------------------------------------------------------------------------
#
# Uncomment to allow 'F6' to terminate the running emulator.
#
#----------------------------------------------------------------------------

#set OPA0 stop_on = F6

#============================================================================
#
# Improve granularity of emulated AXP timer.
#
#----------------------------------------------------------------------------

#set isa clock_period=1000

#============================================================================
#
# Uncomment to connect the emulator's DQA0 to host's ATAPI CD/DVD-ROM drive.
#
#----------------------------------------------------------------------------

#set ide container = "/dev/sg<N>"

#============================================================================
#
# Load optional DE500BA PCI Ethernet Adapter (EWA).
#
#----------------------------------------------------------------------------

#load DE500BA/dec21x4x EWA interface=EWA0
#load packet_port/chnetwrk EWA0 interface="eth0"

#============================================================================
#
# Load another optional DE500BA PCI Ethernet Adapter (EWB).
#
#----------------------------------------------------------------------------

#load DE500BA/dec21x4x EWB interface=EWB0
#load packet_port/chnetwrk EWB0 interface="eth1"

#============================================================================
#
# Load another optional DE500BA PCI Ethernet Adapter (EWC).
#
#----------------------------------------------------------------------------

#load DE500BA/dec21x4x EWC interface=EWC0
#load packet_port/chnetwrk EWC0 interface="eth2"

#============================================================================
#
# The AlphaServer GS80 contains built-in PCI SCSI adapter, called PKA within
# the configuration file.
#
#----------------------------------------------------------------------------
#
# Uncomment to connect the emulator's DKA0 to the disk image.
#
#----------------------------------------------------------------------------

#set PKA container[0] = "<file-name>.vdisk"

#============================================================================
#
```

```
# Uncomment to connect the emulator's DKA100 to a host disk drive.
#
#----------------------------------------------------------------------------

#set PKA container[100] = "/dev/sd<L>"

#============================================================================
#
# Uncomment to connect the emulator's GKA200 to an unknown SCSI device.
#
#----------------------------------------------------------------------------

#set PKA container[200] = "/dev/sg<N>"

#============================================================================
#
# Uncomment to connect the emulator's DKA300 to the host's CD/DVD-ROM drive.
#
# Device name may be different depending on particular version of host
# operating system. Choose one which suits best.
#
#----------------------------------------------------------------------------

#set PKA container[300] = "/dev/cdrom"
#set PKA container[300] = "/dev/cdrom1"
#set PKA container[300] = "/dev/cdrom<N>"
#set PKA container[300] = "/dev/sr0"
#set PKA container[300] = "/dev/sr<N>"

#============================================================================
#
# Uncomment to connect the emulator's DKA400 to an .ISO file (CD/DVD image).
#
#----------------------------------------------------------------------------

#set PKA container[400] = "<file-name>.iso"

#============================================================================
#
# Uncomment to connect the emulator's MKA500 to the host's SCSI tape drive.
#
#----------------------------------------------------------------------------

#set PKA container[500] = "/dev/sg<N>"

#============================================================================
#
# Uncomment to connect the emulator's MKA600 to a .VTAPE file (tape image).
#
#----------------------------------------------------------------------------

#set PKA container[600] = "<file-name>.vtape"

#============================================================================
#
# Uncomment to enable emulation of secondary DEC-KZPBA SCSI controller (PKB).
#
#----------------------------------------------------------------------------

#load KZPBA PKB scsi_id = 7

#============================================================================
#
# Uncomment to connect the emulator's DKB0 to the disk image.
#
#----------------------------------------------------------------------------

#set PKB container[0] = "<file-name>.vdisk"

#============================================================================
#
# Uncomment to connect the emulator's DKB100 to a host disk drive.
```

```
#
#-----------------------------------------------------------------------------

#set PKB container[100] = "/dev/sd<L>"

#=============================================================================
#
# Uncomment to connect the emulator's GKB200 to an unknown SCSI device.
#
#-----------------------------------------------------------------------------

#set PKB container[200] = "/dev/sg<N>"

#=============================================================================
#
# Uncomment to connect the emulator's DKB300 to the host's CD/DVD-ROM drive.
#
# Device name may be different depending on particular version of host
# operating system. Choose one which suits best.
#
#-----------------------------------------------------------------------------

#set PKB container[300] = "/dev/cdrom"
#set PKB container[300] = "/dev/cdrom1"
#set PKB container[300] = "/dev/cdrom<N>"
#set PKB container[300] = "/dev/sr0"
#set PKB container[300] = "/dev/sr<N>"

#=============================================================================
#
# Uncomment to connect the emulator's DKB400 to an .ISO file (CD/DVD image).
#
#-----------------------------------------------------------------------------

#set PKB container[400] = "<file-name>.iso"

#=============================================================================
#
# Uncomment to connect the emulator's MKB500 to the host's SCSI tape drive.
#
#-----------------------------------------------------------------------------

#set PKB container[500] = "/dev/sg<N>"

#=============================================================================
#
# Uncomment to connect the emulator's MKB600 to a .VTAPE file (tape image).
#
#-----------------------------------------------------------------------------

#set PKB container[600] = "<file-name>.vtape"

#=============================================================================
#
# Uncomment to enable emulation of DEC-KGPSA-CA PCI FC Adapter.
#
#-----------------------------------------------------------------------------

#load KGPSA FGA

#=============================================================================
#
# Uncomment to connect the emulator's $1$DGA0 to the disk image.
#
#-----------------------------------------------------------------------------

#set FGA container[0] = "<file-name>.vdisk"

#=============================================================================
#
```

```
# Uncomment to connect the emulator's $1$DGA100 to a host disk drive.
#
#----------------------------------------------------------------------------

#set FGA container[100] = "/dev/sd<L>"

#============================================================================
#
# Uncomment to enable emulation of secondary DEC-KGPSA-CA PCI FC Adapter.
#
#----------------------------------------------------------------------------

#load KGPSA FGB

#============================================================================
#
# Uncomment to enable PCI Pass Through access to physical EMULEX LP FC HBA,
# use two adapters to provide multipath with failover.
#
#----------------------------------------------------------------------------

#set FGA host_bus_location = "/dev/kgpsaX"
#set FGB host_bus_location = "/dev/kgpsaY"

# this is the end of the configuration file #################################
```

Back to Top

CHARON-AXP for Linux deinstallation

Deinstallation procedure

To uninstall a particular CHARON-AXP product:

1. Stop all running CHARON-AXP instances, remove all CHARON-AXP services.

2. Login as "root" user.

3. Issue the following command:

   ```
   # yum remove <CHARON-AXP product name>
   ```

   Example:

   ```
   # yum remove charon-axp-es40-4.6-16803
   ```

If it is required to uninstall all products completely (including all compartment components and drivers), use the following command:

```
# yum remove aksusbd `rpm -q -a | grep charon`
```

Back to Table of Contents