



# CHARON-AXP V4.8 for Linux Users Guide



# Contents

---

Introduction	3
Conventions	6
CHARON-AXP for Linux installation	7
Running CHARON-AXP for Linux	20
CHARON-AXP for Linux configuration	25
Migration to CHARON-AXP for Linux	38
CHARON-AXP for Linux virtual network	46
CHARON-AXP for Linux licensing	50
CHARON-AXP for Linux utilities	60
mkdskcmd	61
mtd	64
hasp_srm_view	65
hasp_update	66
ncu	67
CHARON Guest Utilities for OpenVMS	75
CHARON-AXP for Linux configuration details	79
General Settings	80
Core Devices	88
Console	105
Placement of peripheral devices on PCI bus	115
Disks and tapes	153
KZPBA PCI SCSI adapter	154
KGPSA-CA PCI Fibre Channel adapter	162
Acer Labs 1543C IDE/ATAPI CD-ROM adapter	171
PCI I/O Bypass controller	173
Finding the target "/dev/sg" device	179
Networking	181
Sample configuration files	187
HP AlphaServer 800 configuration file	188
HP AlphaServer 4000 configuration file	193
HP AlphaServer DS20 configuration file	198
HP AlphaServer ES40 configuration file	203
HP AlphaServer GS80 configuration file	209
CHARON-AXP for Linux deinstallation	215
Appendixes	216
glibc.i686 installation without Internet connection	217
How to implement time synchronisation between CHARON-AXP Host OS and Guest OS	221

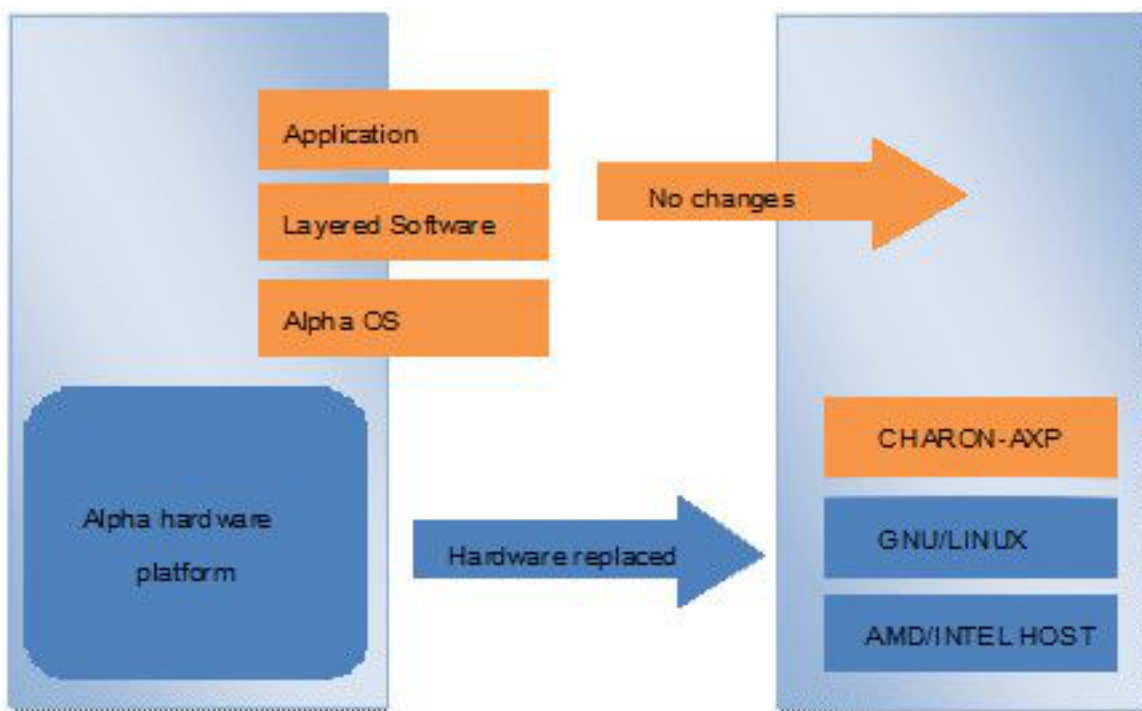
# Introduction

## Table of Contents

- General Description
- The principles of HP Alpha Hardware Virtualization
  - Virtualized hardware
  - Host platform

## 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
- AlphaServer ES40
- AlphaServer ES45

- AlphaServer GS80
- AlphaServer GS160
- AlphaServer GS320

[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.

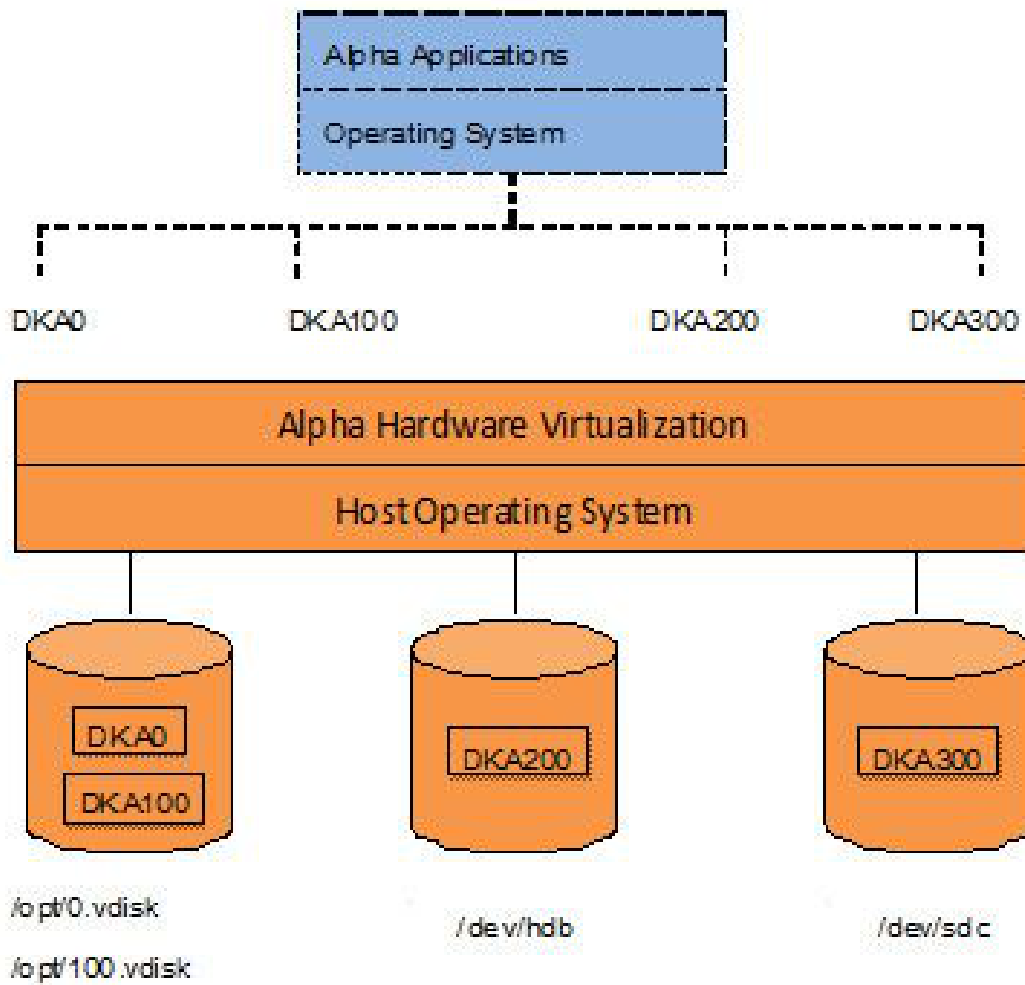
The following table shows which hardware boards CHARON virtualizes:

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).
<b>User input</b>	Bold monospace type in interactive examples indicates typed user input.
<b>&lt;path&gt;</b>	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.
<i>disk0</i>	Italic monospace type, in interactive examples, indicates typed context dependent user input.

## The following definitions apply

Term	Description
Host	The system on which the emulator runs, also called the Charon server
Guest	The operating system running on a Charon instance, for example, Tru64 UNIX, OpenVMS, Solaris or MPE

# CHARON-AXP for Linux installation

## Table of contents

- Introduction
- Hardware Requirements
  - Number of CPU cores
  - CPU type and speed
  - Operative memory
  - Disk storage
  - Ethernet adapters
- Software Requirements
- Host system preparation
- Before installation
- Distribution preparation
- Installation
- CHARON-AXP home directory
- Specific user account creation
- License installation
  - Regular HASP USB dongle
  - Network HASP USB dongle
  - Software license
- License validity verification
  - Troubleshooting
- Network configuration
  - Configuration with NCU utility
  - Manual Configuration
    - Choosing a network interface
    - Designation of network interface to CHARON
    - Switching off the offload parameters
- Final steps
- Upgrade to new version

## Introduction

The CHARON-AXP product is 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 RPM files if they are not needed.

CHARON installation consists of the following steps:

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

[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 cumulative.

The following table lists the minimum and recommended number of CPUs required for each virtual HP Alpha instance (note that each CHARON 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	Minimum number of host CPU cores	Recommended number of host CPU cores
HP AlphaServer 400 - HP AlphaServer 4100	2	2
HP AlphaServer DS10 and DS15	2	2
HP AlphaServer DS20	4	4
HP AlphaServer ES40	6	8
HP AlphaServer GS80	10	16
HP AlphaServer GS160	18	32
HP AlphaServer GS320	34	48

**Hyper-threading must be switched off completely.** Disable hyper-threading 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 Hyper-threaded Core Sharing mode to *None*.

## CPU type and speed

Since CHARON-VAX utilizes LAHF instruction in VAX 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 and Intel introduced it with the Pentium 4 G1 stepping in December 2005.

Concerning CPU speed, the general recommendation is that higher the CPU frequency is, better the emulated HP Alpha performances will be. The minimum 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 executed on one host.
- is calculated according to the following formula:

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

## Disk storage

The total amount of disk space required for CHARON-AXP can be calculated as a sum of all the disk/tape image sizes plus 50 MB for the CHARON software plus the space required for the host operating system. Temporary disk storage is often needed when setting up a new virtual machine (for source disks backups storage, software installation kits, etc...).

When virtual disks/tapes are used to represent physical disk drives / magnetic tapes, the disk/tape image files have the same size as their hardware equivalent, regardless of their degree of utilization.

## Ethernet adapters

CHARON-AXP networking requires 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, management interface, etc.

It is also possible to use [virtual network interfaces](#), but for performance considerations, it is recommended to use physical ones only.

For VMware based CHARON hosts, it is mandatory to use the "E1000" virtual network adapters. "E1000E" adapters are not supported.

[Back to Table of Contents](#)



## Software Requirements

- Red Hat Enterprise Linux 7.x, 64bit
- Red Hat Enterprise Linux 6.5 - 6.8, 64bit
- Linux Centos 7.x, 64bit
- VMware ESXi 5.5 and 6.x up to 6.5 (requires a supported Linux operating system on top of a ESXi virtual machine)

For CentOS, a connection to the internet is required to install the `glibc.i686` package which is not included in the Standard distribution DVD. If there's no connection available, please use the "Everything" distribution DVD.

[Back to Table of Contents](#)

## Host system preparation

The automatic installation of updates must be disabled. Updates to the CHARON host must be done only in specific service maintenance periods established by the system administrator. Before applying new updates one must shutdown the operating system running on CHARON and stop all the running CHARON instances and services.

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

- *On the server side (where the network license will reside):* open port 1947 for both TCP and UDP
- *On the client side,* if broadcast search for remote licenses is to be used, UDP traffic from port 1947 of the license server to ports 30000-65535 of the client must be permitted
- *Both on server and client sides:* set 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 the superuser ("`root`") on the host system. Because Sentinel HASP runtime relies on 32-bit compatibility libraries to run on Linux, the 32-bit compatibility libraries must be installed before continuing. If the emulator host has access to a package repository, either local or remote, use the following command:

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

**i** Sometimes it is not possible to use an online repository for the installation of 32-bit glibc package. In this case the procedure described in the appendixes has to be used: [glibc.i686 installation without Internet connection](#)

2. Create a directory for the CHARON-AXP distribution, copy the TAR.GZ files there and change to this directory as shown in the following example:

```
# mkdir /charon_dist
# cp /tmp/charon-axp-4.8-18302.el71.tar.gz /charon_dist
# cd /charon_dist
```

### WARNING

- If you plan to install CHARON-VAX on the same server, both products, CHARON-AXP and CHARON-VAX, will have to be the same build number.
- If you upgrade from a previous version of CHARON-AXP, please stop all running CHARON virtual machines and uninstall CHARON products before proceeding with the installation steps described below.

## Distribution preparation

- Extract the contents of the distribution TAR.GZ files to the current directory:

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

where:

	Description
VER	Version of CHARON-AXP product, for example 4.8
BN	Build Number of CHARON-AXP product, for example 18302
ZZ	CHARON-AXP target operating system identifier. For CentOS 7.x 'ZZ' value is 'e171', for Red Hat Enterprise Linux 7.x the value is 'e171' and for Red Hat Enterprise Linux 6.5 to 6.8 the value is 'e165'.

### Example:

```
# tar -xvzf charon-axp-4.8-18302.e171.tar.gz
```

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

- Switch to that directory:

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

### Example:

```
# cd charon-axp-4.8-18302.e171
```

- The distribution directory contains the following RPM files:

File name	Description
charon-axp-VER-BN.ZZ.x86_64.rpm	CHARON-AXP
aksusbd-2.5-1.i386.rpm	HASP Run-time
charon-license-VER-BN.ZZ.x86_64.rpm	CHARON Libraries
charon-utils-VER-BN.ZZ.x86_64.rpm	CHARON Utilities

### Example:

```
# ls
aksusbd-2.5-1.i386.rpm
charon-axp-4.8-18302.e171.x86_64.rpm
charon-license-4.8-18302.e171.x86_64.rpm
charon-utils-4.8-18302.e171.x86_64.rpm
```

## Installation

Issue the following command to install all the RPM files present in the directory:

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

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

### Example:

```
Dependencies Resolved

=====

Package Arch Version Repository Size
=====

Installing:
aksusbd i386 2.5-1 /aksusbd-2.5-1.i386 2.9 M
charon-axp x86_64 4.8-18302 /charon-axp-4.8-18302.el71.x86_64 260 M
charon-license
x86_64 4.8-18302 /charon-license-4.8-18302.el71.x86_64 2.9 M
charon-utils
x86_64 4.8-18302 /charon-utils-4.8-18302.el71.x86_64 1.8 M

Transaction Summary
=====

Install 4 Packages

Total size: 267 M
Installed size: 267 M
Is this ok [y/d/N]:y
```

Check the installation process has completed successfully.

### Example:

```
Downloading packages:
Running transaction check
Running transaction test
Transaction test succeeded
Running transaction (shutdown inhibited)
Installing : aksusbd-2.5-1.i386 1/4
Starting aksusbd (via systemctl): [ OK ]
Installing : charon-utils-4.8-18302.x86_64 2/4
Installing : charon-license-4.8-18302.x86_64 3/4
Installing : charon-axp-4.8-18302.x86_64 4/4
Verifying : aksusbd-2.5-1.i386 1/4
Verifying : charon-license-4.8-18302.x86_64 2/4
Verifying : charon-axp-4.8-18302.x86_64 3/4
Verifying : charon-utils-4.8-18302.x86_64 4/4

Installed:
aksusbd.i386 0:2.5-1 charon-axp.x86_64 0:4.8-18302
charon-license.x86_64 0:4.8-18302 charon-utils.x86_64 0:4.8-18302

Complete!
```

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

```
# . /etc/profile.d/charon.sh
```

Note that the "charon-utils" package has the following dependencies:

- ethtool
- bridge-utils
- net-tools
- iproute
- NetworkManager

During "charon-utils" 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 the executable files
/cfg	Contains the configuration files templates
/doc	Contains the documentation
/log	Contains the log files
/disks	Contains the disk containers
/drivers	Contains the CHARON drivers

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

[Back to Table of Contents](#)

## Specific user account creation

Create a specific user account named "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 into these 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
```

Re-login to apply changes.

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

[Back to Table of Contents](#)

## License installation

### Regular HASP USB dongle

If the CHARON license is located on a regular USB dongle, just connect it to the host USB port.

If the CHARON host is accessed remotely, please note that regular HASP licenses cannot be displayed and used to start a CHARON virtual machine. As a workaround it is possible to install CHARON as a daemon (service). This procedure will be described later.

### Network HASP USB dongle

If the 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 and provide it to other hosts on the local network at the same time) or to install it on a local network "license server" for remote access from this particular host.

If a remote license server is to be used:

- Copy the aksusbd-2.5-1.i386.rpm and charon-license-4.8-<build>.<OS identifier>.x86\_64.rpm files (see above) to the server, for example to "/tmp".
- Login as "root" on the server.
- Switch to that directory.
- Install the copied files using "yum".

**Example:**

```
# cd /tmp
# yum install aksusbd* charon-license-*
```

- Connect the network HASP dongle to one of the server USB ports.

The network HASP (red dongles) licenses have no restrictions with respect to remote access.

## Software license

If the CHARON license is a software license (SL), it is installed on the host using the following procedure:

1. Run the `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 file to any folder on the CHARON host, invoke the system default web browser and enter the URL <http://localhost:1947> to display the "**Sentinel Admin Control Center**" (**ACC**) web interface. This interface allows you to view and manage the CHARON licenses.
4. In the **ACC** perform the following steps: select **Update/Attach** from the menu on the left pane then use the **Browse** button to select the received file and click on the **Apply File** button to install the license.
5. Ensure that the software license is now visible in the "**Sentinel Keys**" section of the **ACC**.

It is also possible to use the "`hasp_update`" utility for applying ".v2c" files.

The Software Licenses (SL) are always network licenses, they have no restrictions with respect to being displayed or accessed via a remote connection.

A "Provisional" (demo) license does not require collecting a fingerprint. For its installation start at step 3 in the sequence above

[Back to Table of Contents](#)

## License validity verification

To check the CHARON license validity, invoke the `hasp_srm_view` utility to make sure that the CHARON license is visible and is correct:

- Text of the license is displayed correctly by the `hasp_srm_view` utility, no error messages are shown.
- The 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: dlao.msc.masq

License Manager IP address: 192.168.1.129


HASP Net key detected

The Physical KeyId: 1422726238
CHARON Sentinel HASP License key section
Reading 4032 bytes

The License Number: 000.msc.sanity.tests
The License KeyId: 1422726238
The Master KeyId: 827774524
Release date: 12-SEP-2016
Release time: 15:15:15
Update number: 1
End User name: MSC
Purchasing Customer name: STROMASYS
...
```

If multiple licenses are available, it is possible to check them using the "-all" parameter with the `hasp_srm_view` utility in the following way:

```
# hasp_srm_view -all
```

 It is also possible to display the license content for one specific key using the "-key" parameter and specifying the Key Id (see "`# hasp_srm_view -h`" for more)

**Reminder:** If the CHARON host is accessed over a remote connection, please note that regular HASP licenses cannot be displayed and used in this case. As a workaround it is possible to install CHARON as a daemon (service). This procedure will be described later.

## Troubleshooting

If the CHARON license content cannot be displayed by the `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 the URL `http://localhost:1947` to display the "**Sentinel Admin Control Center**" (ACC) web interface.
2. Click on "**Sentinel Keys**" link to open the corresponding 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), that the 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 to request a new license update.
Several License keys / SLs are displayed	Remove all of them except the one provided by STROMASYS for the just installed version of CHARON.

Removing licenses can be done by physical disconnection of the corresponding USB HASP keys from the CHARON host and physical disconnection of the network HASP keys from all hosts on the local network (or by disabling remote access to network licenses from the CHARON host - see detailed explanations below).

It is also possible to disable access to network licenses if only a local license is to be used: Click on the "**Configuration**" link to open the "**Configuration for Sentinel Manager**" page.

Uncheck "**Allow Access to Remote Licenses**" and "**Broadcast Search for Remote Licenses**" check boxes from the "**Access to Remote License Managers**" tab then press the "**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 they have to be specified in the configuration file.

**Example:**

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

It is also possible to have one "main" and one "backup" license in case the main license becomes unavailable:

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

CHARON-AXP checks its licenses from time to time starting with the main license. If it becomes unavailable, it attempts to access the backup license.

[Back to Table of Contents](#)

## Network configuration

In most cases CHARON will use a network. In this case CHARON requires one or more dedicated network interfaces with any other protocols including TCP/IP removed at the host level.

Two ways of network configuration are possible:

- With the help of the "ncu" utility
- Manually

The first way is recommended. Use the manual approach only in absence of the "ncu" utility or if it impossible to use it.

### Configuration with NCU utility

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

```
# ncu
CHARON Network Configuration Utility, STROMASYS (c) 2016 Version 1.6

Interfaces Dedicated to State
-----
eth0      host          connected to host
eth1      host          disconnected from host

=====VLAN=====
=====

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

The utility lists the 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 the host operating system.



"ncu" offers several options:

- Dedicate interface to CHARON (option "1")
- Release interface to host (option "2")
- Create a bridge between a chosen physical network interface and the Linux virtual network and create a number of virtual network interfaces (option "3")
- Remove the Linux virtual network and all the created virtual network interfaces (option "4")
- Add VLAN (option "5")
- Remove VLAN (option "6")
- Print status (option "7") - use it to display status of network interfaces and the menu shown above
- Exit (option "8")

In the example above we see 2 network interfaces, "eth0" and "eth1", that are dedicated to the host and the host uses only the interface "eth0".

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

Enter "1" then "eth1":

```
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 - Add VLAN
6 - Remove VLAN
7 - Print status
8 - Exit

:> 7
```

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

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

=====VLAN=====
=====

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

:>
```

Enter "8" to return to the console prompt.

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

## Manual Configuration

### Choosing a 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 an 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-ethN" (where N is the number of the interface to be used for CHARON, in this case it is "1") and make sure that all the IP-setup related parameters are removed. The file must look like this:

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

## Switching off the offload parameters

Determine what additional parameters are currently set to "on" on the host network adapter to be used by CHARON using the following command:

```
# 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
```

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 executed at system startup as the offload parameters must be switched off following 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 the name of the file is "offload\_off\_eth1.txt". To execute it on system startup, execute the following command (example):

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

[Back to Table of Contents](#)

## Final steps

- Reboot the host system
- Login as user "charon"
- Verify the offload parameters are effective

[Back to Table of Contents](#)

## Upgrade to new version

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

1. Ensure your license allows you to upgrade to that version. 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 a dedicated folder.
5. Login as "root" user.
6. Remove the old CHARON-AXP version as described in the "[CHARON-AXP for Linux deinstallation](#)" chapter.
7. Proceed with the instructions on the new kit installation as described in the "[Installation](#)" section.
8. Install the license for the new CHARON-AXP as described in the "[License installation](#)" section.
9. Start all the CHARON-AXP services stopped at step #3.

[Back to Table of Contents](#)

# Running CHARON-AXP for Linux

## Table of Contents

- CHARON-AXP symbolic links
- Running CHARON-AXP emulators
  - Running from the console
  - Running as system service (daemon)
    - Installation and start of CHARON-AXP service
    - Stopping CHARON-AXP service
    - Removing CHARON-AXP service

[Back to Table of Contents](#)

## CHARON-AXP symbolic links

Each model of CHARON-AXP has a symbolic link defined to point to the corresponding CHARON executable (see the table below). If the PATH is correctly defined, you can start a virtual machine by specifying the link followed by the configuration file. This is described further.

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.

For 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 [Exception: clustering] - all these objects will be explained later in this

document) and check the hardware devices (the CDROM drive for example) are used by only one instance at a time (not shared).

**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 the console

Copy the selected configuration template from the "/opt/charon/cfg/" directory to some local file and set the correct privileges for the file to be edited.

**Example:**

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

Run the virtual machine using this configuration file:

```
$ es40 my_es40.cfg
```

Below is an example of a normal HP Alpha test sequence, followed by the 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 4.8.18302
(C) 2009-2016 STROMASYS SA.
All rights reserved.


P00>>>
```

The next stage can be either installation of a new HP Alpha/VMS system using a distribution provided by HP or data transfer from some existing HP Alpha system. These possibilities will be discussed in details in the 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 where CHARON-AXP was started, open them with an editing tool and analyze their content. In most cases those files contain very helpful information on what may possibly went wrong.

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

Configuration	How to exit
No change to the template configuration file	Enter <b>"power off"</b> in the CHARON console
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>	Press the <b>"F6"</b> key

 Please note that before stopping CHARON-AXP, a clean shutdown of the operating system running on the virtual machine has to be performed.

## Running as system service (daemon)

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

Follow the description below to install and execute CHARON-AXP as a daemon:

### Installation and start of CHARON-AXP service

1. Copy the sample script `/opt/charon/bin/charon.service` (`/opt/charon/bin/charon` for Red Hat Linux 6.x) to the `/usr/lib/systemd/system/` directory (or to your home directory for Red Hat Enterprise Linux 6.x).


#### Example:

Red Hat Linux 6.x	<pre>\$ cp /opt/charon/bin/charon /my_services/es40_service \$ chmod 755 /my_services/es40_service</pre>
Red Hat Linux 7.x and CentOS 7.x	<pre>\$ cp /opt/charon/bin/charon.service /usr/lib/systemd/system/es40.service \$ chmod 755 /usr/lib/systemd/system/es40.service</pre>

2. Edit the renamed file to replace sample values of the following parameters.

#### Example:

Red Hat Linux 6.x	<pre>exec="/opt/charon/bin/es40" prog="my_es40" config="/my_services/es40-service.cfg"</pre>
Red Hat Linux 7.x and CentOS 7.x	<pre>ExecStart=/opt/charon/bin/es40 -d /my_services/es40-service.cfg WorkingDirectory=/my_services</pre>

 "my\_es40" is a service name in the example above

3. Create and edit the configuration file (`/my_services/es40-service.cfg` in the examples above) the way it was described before and make sure the following pre-requisites are met:

- OPA0 must be configured as a virtual port or physical console, not as an operator console.

#### Example:

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

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

#### 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 and the OPA0 console port number is unique across the CHARON server.

Once the configuration file is ready, execute the following commands (based on the examples above) to install and start CHARON-AXP as a daemon:

Red Hat Linux 6.x	<pre># ln -sf /my_services/es40_service /etc/init.d/es40_service # chkconfig es40_service on # service es40_service start</pre>
Red Hat Linux 7.x and CentOS 7.x	<pre># systemctl enable es40.service # systemctl start es40.service</pre>

#### Red Hat Linux 7.x and CentOS 7.x

If you update the `/usr/lib/systemd/system/<my virtual machine>.service` file, the following command must be executed in order to take changes into account:

```
# systemctl daemon-reload
```

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

This problem can be avoided by specifying the "license\_key\_lookup\_retry" parameter in the following way:

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

where:

- N = Number of retries looking for the 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 a CHARON-AXP daemon, use the following commands.

#### Example:

Red Hat Linux 6.x	<pre># service es40_service stop</pre>
Red Hat Linux 7.x and CentOS 7.x	<pre># systemctl stop es40</pre>

 Please note that before stopping a CHARON-AXP service, a clean shutdown of the operating system running on the virtual machine has to be performed.


[Back to Table of Contents](#)

## Removing CHARON-AXP service

To remove a CHARON-AXP daemon use the following commands.

### Example:

Red Hat Linux 6.x	<pre># chkconfig es40_service off # chkconfig --del es40_service # rm -f /etc/init.d/es40_service</pre>
Red Hat Linux 7.x and CentOS 7.x	<pre># systemctl disable es40.service # rm -f /usr/lib/systemd/system/es40.service</pre>

 Please note that before removing a CHARON-AXP service, a clean shutdown of the operating system running on the virtual machine has to be performed and the service has to be stopped.

 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
- HP Alpha model specification
- Configuration name
- Log file parameters
  - Rotating log (default)
  - Single log
- CPU affinity
- Number of host CPUs dedicated to CHARON I/O
- Setting a specific HP Alpha model
- Reducing number of emulated CPUs
- Setting system serial number
- TOY and ROM containers
- Emulated memory (RAM) size
- Console
  - Mapping to system resources
  - Exit on pressing F6 key
- Improve granularity of emulated timer
- ATAPI CD/DVD-ROM configuration
- Networking
- Disk/tape subsystem
  - KZPBA PCI SCSI disk/tape controller
  - KGPSA-CA PCI FC disk controller
    - KGPSA-CA mapping to the host resources
    - KGPSA-CA pass through mode
- Auto boot

## 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.

### 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 modifications 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:** the lines preceded 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.

### 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.

**Example:**

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

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

```

20161120:113009:INFO :0:000003A5:hexane.cxx(5312): session is loading built-in configuration
"AlphaServer_ES40"...
20161120:113009:INFO :0:000003A6:hexane.cxx(5336): session has finished loading built-in configuration
"AlphaServer_ES40"...
20161120:113009:INFO :0:000003AA:hexane.cxx(5425): session is loading configuration file "es40.cfg"...
20161120:113009:INFO :0:000003AB:hexane.cxx(5455): session has finished loading configuration file "es40.cfg".
20161120:113009:INFO :0:0000032B:hexane.cxx(2547): Start request received.
20161120:113010:INFO :0:000003AC:hexane.cxx(1287): session's process affinity is 000000000000000F, system
affinity is 000000000000000F.
20161120:113010:INFO :0:000003D1:hexane.cxx(1541): session's I/O domain affinity is 0000000000000001, CPU
domain affinity is 000000000000000E
20161120:113010:INFO :0:000003BA:ll_sentine( 725): Looking for a license key ...
20161120:113010:INFO :0:000003DC:ll_sentine( 820): ... found license key 1422726238.
20161120:113010:INFO :0:0000024D:hexane.cxx(2688): STROMASYS SA, (C) 2009-2016
20161120:113010:INFO :0:00000350:hexane.cxx(2734): CHARON-AXP (AlphaServer ES40), V 4.8 B 18302, Nov 11 2016 /
000.msc.sanity.tests / 1422726238
20161120:113010:INFO :0:00000336:hexane.cxx(2761): 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
20161120:113010:INFO :0:0000009D:hexane.cxx(2838): License info:
CHARON product code: "CHAXP-460xx-WI".
Licensed to: "MSC".

20161120:113011:INFO :0:00000097:hexane.cxx(2847): OS Environment: Linux 3.19.8-100.fc20.x86_64 #1 SMP Tue May
12 17:08:50 UTC 2015 x86_64.
20161120:113011:INFO :0:00000098:hexane.cxx(2852): 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
20161120:113011:INFO :0:00000099:hexane.cxx(2857): Host Memory: 4096Mb
20161120:113011:WARN :1:00000354:hexane.cxx(3005): 4 host CPUs detected but 8 recommended, performance might be
limited.
20161120:113011:WARN :1:00000353:hexane.cxx(3023): The host system is below recommended specifications.
20161120:113013:ERROR:2:00000100:lnxpackpor( 753): (95) Operation not supported: EWA0: Failed to query for max
frame size. Assume 1518.
20161120:113013:ERROR:2:00000101:lnxpackpor( 754): EWA0: Failed to query for link speed. Assume 10Mbps.
20161120:113013:INFO :0:0000034B:scsi_disk.( 566): PKA_0 is being set ONLINE
container = "/home/charon/Charon/test/performancecomparison-axp.vdisk"
20161120:113013:INFO :0:0000032C:hexane.cxx(2589): "AlphaServer_ES40" started.
20161120:113013:INFO :0:00000348:scsi_disk.( 554): PKA_0 is being set OFFLINE
20161120:113013:INFO :0:0000034B:scsi_disk.( 566): PKA_0 is being set ONLINE
container = "/home/charon/Charon/test/performancecomparison-axp.vdisk"
20161120:113014:INFO :0:00000348:scsi_disk.( 554): PKA_0 is being set OFFLINE
20161120:113014:INFO :0:0000034B:scsi_disk.( 566): PKA_0 is being set ONLINE
container = "/home/charon/Charon/test/performancecomparison-axp.vdisk"
20161120:113014:INFO :0:00000348:scsi_disk.( 554): PKA_0 is being set OFFLINE
20161120:113014:INFO :0:0000034B:scsi_disk.( 566): PKA_0 is being set ONLINE
container = "/home/charon/Charon/test/performancecomparison-axp.vdisk"
20161120:113028:INFO :0:000003D7:hexane.cxx(4928): All virtual CPUs of "AlphaServer_ES40" have been stopped by
now.
20161120:113028:INFO :0:0000032D:hexane.cxx(2633): "AlphaServer ES40" stop request received.
20161120:113028:INFO :0:0000014C:lnxpackpor( 416): EWA0: Stopping network interface ... please wait.
20161120:113028:INFO :0:00000348:ataunit.cx(1738): ide0 is being set OFFLINE
20161120:113028:INFO :0:00000348:ataunit.cx(1738): ide1 is being set OFFLINE
20161120:113029:INFO :0:0000032E:hexane.cxx(2651): 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" mechanism. This means that a new default log file is created each time CHARON starts and can switch to another log file if the size of the log file exceeds 64Kb (this behavior can be changed with the "set session log\_file\_size" and "set session log\_rotation\_period" parameters; see more details in the "General Settings" chapter of this guide).

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.

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

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

## Single log

Alternatively it is possible to use 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 appended each time CHARON-AXP starts:

```
set session log_method = append
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.

**Example:**

```
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).

The full description of the parameters, with 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 the CHARON host contains not enough CPUs to emulate full range of the CPUs provided by a certain HP Alpha model, it is possible to direct CHARON-AXP to reduce the number of emulated Alpha CPUs in the configuration:

```
set session n_of_cpus=1
```

[Back to Table of Contents](#)

## Setting system serial number

This parameter is used to set 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 the TOY, 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.

**Example:**

```
#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 lists all the parameters per model:

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 DS10L	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.

#### Example:

```
#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:

Option	Description								
physical_serial_line	<p>Mapping to host serial line, both physical and virtual. Use the following mapping for different types of host serial lines:</p> <table border="1"> <thead> <tr> <th>Mapping</th> <th>Description</th> </tr> </thead> <tbody> <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> </tbody> </table> <p>The specific account for running CHARON ("charon") does not allow the usage of physical consoles, "/dev/tty&lt;N&gt;", as CHARON consoles. If you plan to map the CHARON console to "/dev/tty&lt;N&gt;", use only the "root" account to run CHARON.</p>	Mapping	Description	/dev/tty<N>	Virtual serial lines	/dev/ttyS<N>	Onboard serial lines	/dev/ttyUSB<N>	Modem or usb serial lines adapters
Mapping	Description								
/dev/tty<N>	Virtual serial lines								
/dev/ttyS<N>	Onboard serial lines								
/dev/ttyUSB<N>	Modem or usb serial lines adapters								
virtual_serial_line	<p>Mapping to an IP port of the CHARON-VAX host. Using this mapping it is possible to connect to the CHARON-VAX console and disconnect from it at any time.</p>								
operator_console	<p>Mapping to the current TTY console</p>								

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 key

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:

```
set OPA0 stop_on = F6
```

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

[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, DE602 and DE602AA 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"
```

```
load DE602/i8255x EIA interface=EIA0
load packet_port/chnetwrk EIA0 interface="eth0"
```

In the examples above the first line loads DE500BA/DE602 virtual adapter with a name "EWA"/"EIA" (note that "/i8255x" syntax must be used only in case of DE602 and DE602AA adapters); 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"/"EIA0" 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"/"EIA" is connected to the "packet\_port" object "EWA0"/"EIA0"

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

```
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 "mkdiskcmd" 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.  📘 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.  <div style="border: 1px solid yellow; padding: 5px; margin: 5px 0;">Since "/dev/sd&lt;L&gt;" addressing is not persistent, it is strongly recommended to use "/dev/disk/by-id/wwn-*" syntax instead to refer the disk by its WWID - especially in the environments utilizing FC and SAN storages (see below).</div>
"/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/disk/by-*"	Mapping to physical disk. <ul style="list-style-type: none"> <li>• by-id (addressing by the disk ID, for example "/dev/disk/by-id/ata-ST1000DM003-9YN162_S1D01QJ4")</li> <li>• by-label (addressing by the disk label, for example "/dev/disk/by-label/MyStorage")</li> <li>• by-uuid (addressing by the disk UUID, for example "/dev/disk/by-uuid/0e808a2f-cdd3-4944-a245-f729ffd73882")</li> </ul> ⚠ 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 explained in <a href="#">this section</a> .
"/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 KZPBA controller. They have the following structure:

[XXYY], where

Parameter	Range	Description
XX	0...15	<p>Stands for SCSI ID of each connected unit.</p> <p><b>i</b> Note that KZPBA itself has some ID associated with it. By default it is 7, but it can be changed in the following way:</p> <pre>load KZPBA PKA scsi_id = 0</pre> <p>In this example an instance "PKA" of KZPBA controller is assigned with SCSI ID 0.</p>
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 "mkdiskcmd" 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.   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.  <div style="border: 1px solid #f0e68c; padding: 10px; margin: 10px 0;">Since "/dev/sd&lt;L&gt;" addressing is not persistent, so it is strongly recommended to use "/dev/disk/by-id/wwn-*" syntax instead to refer the disk by its WWID - especially in the environments utilizing FC and SAN storages (see below).</div>
"/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/disk/by-*"	Mapping to physical disk. <ul style="list-style-type: none"><li>• by-id (addressing by the disk ID, for example "/dev/disk/by-id/ata-ST1000DM003-9YN162_S1D01QJ4")</li><li>• by-label (addressing by the disk label, for example "/dev/disk/by-label/MyStorage")</li><li>• by-uuid (addressing by the disk UUID, for example "/dev/disk/by-uuid/0e808a2f-cdd3-4944-a245-f729ffd73882")</li></ul>  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 an 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
- Collecting information about the source HP Alpha system
- Creation of CHARON-AXP configuration file
- Making disk images
- Installation of HP Alpha operating system
- Making remote backups
- Restore backups to CHARON-AXP disks
- Alternative ways of data transfer

## 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 a "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
<code>#!/sbin/hwmgr view devices</code>	Get detailed information about the host hardware configuration
<code>#!/sbin/hwmgr show scsi</code>	Get specific information about the host SCSI controllers and attached disks
<code>#!/sbin/hwmgr view hierarchy</code>	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 execllet = 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 12-SEP-2016 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 information, 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 the 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 larger disks to get extra storage space.

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

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

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

```
$ mkdiskcmd -d rz24 -o /my_disks/rz24.vdisk
Please wait...
100% done
Success.
$ mkdiskcmd -d rz23 -o /my_disks/rz23.vdisk
Please wait...
100% done
Success.
$ mkdiskcmd -d rz22 -o /my_disks/rz22.vdisk
Please wait...
100% done
Success.
$ mkdiskcmd -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 backups over the network. 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 large 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 4.8.18200
(C) 2009-2016 STROMASYS SA.
All rights reserved.

P00>>>boot dqa0

```

Install HP Alpha/VMS including DECnet on "dka400". The 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 the CHARON-AXP virtual machine and make sure that the source HP Alpha system is reachable 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 the SYS\$MANAGER:SYSHUTDOWN.COM file may be helpful. The goal is to close as many files as possible. The system disk will have several files opened (pagefile, swapfile, etc.), this is a normal situation.

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

```

When the backup procedure will be completed, the disk "DKA300" of the CHARON-AXP virtual machine 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 in 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 the 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 to 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 the CHARON-AXP host via a PCI card
  - Map the tape drive in the CHARON-AXP configuration file
    - a. Restore the source HP Alpha system backups from tape to disk images via OpenVMS running on CHARON-AXP.
    - b. Boot from standalone backups and restore the content to CHARON-AXP virtual disks.
  - Dump the source HP Alpha system backups to tape images with the "mtd" utility and:
    - a. Boot from the freshly installed OpenVMS system and restore the tape images to CHARON-AXP virtual disks.
    - b. Boot from standalone backups and restore the 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 a boot server) then perform backups 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
- Using "ncu" utility to establish CHARON virtual network
- Manual configuration of CHARON virtual network
  - Host preparation
  - Virtual interface creation
  - Bridge creation
  - Starting bridge
- Usage of the virtual interface in CHARON-AXP configuration

## General description

It is strongly recommended to use only physical network adapters for CHARON-AXP networking to gain maximum performances. In situations where the host has only one network adapter, you can use 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 the "ncu" utility
- Manually

[Back to Table of Contents](#)

## Using "ncu" utility to establish CHARON virtual network

Login as root and start the "ncu" utility:

```
# ncu
CHARON Network Configuration Utility, STROMASYS (c) 2016 Version 1.6

Interfaces Dedicated to State
-----
eth0 host connected to host
eth1 host disconnected from host
lo host unmanaged from host
=====
bridge name bridge id STP enabled interfaces
===== VLAN =====
=====
select action:
1 - Dedicate to CHARON
2 - Release to host
3 - Create Bridge with TAPs
4 - Remove Bridge
5 - Add VLAN
6 - Remove VLAN
7 - Print status
8 - 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 - Add VLAN
6 - Remove VLAN
7 - Print status
8 - Exit

:> 7
```

Now enter "7" to see the created virtual interfaces:

```
Interfaces Dedicated to State
-----
eth0 host connected to host
eth1 bridge disconnected from bridge
lo host unmanaged from host
tap0 CHARON connected to host
tap1 bridge connected to bridge
=====
bridge name      bridge id          STP enabled   interfaces
br0_eth1         8000.22314588acac  no           eth1
tap0
tap1
=====  VLAN  =====
=====

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

:> 8
```

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 "8" to quit the "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
```

The 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/iptables -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

The 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
- Parameters defined by CHARON-AXP license
- CHARON-AXP licensing models
  - Regular Sentinel HASP keys
  - Network Sentinel HASP keys
  - Software licenses
- Multiple licenses configuration and backup license
- License installation
  - Installation of Regular and Network license keys
  - Replacement of currently installed Sentinel run-time
  - Installation and update of CHARON-AXP Software License or HL/HASP dongle License
- License management
  - Sentinel Admin Control Center
    - General Description
    - Disable remote keys access
    - Accessing Sentinel Admin Control Center from remote hosts
  - License management utilities
- Removing CHARON-AXP software licenses
- License Deinstallation
- Special "backup" license keys

## General description

The CHARON-AXP products are protected by licenses, issued by STROMASYS for each customer individually. The CHARON-AXP license defines a set of HP Alpha emulators allowed to run.

The license is implemented in the form of a hardware dongle (a Sentinel HASP key) or a software license bound to the hardware. 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 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 the 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, the 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
<ul style="list-style-type: none"> <li>Physical key ID</li> <li>License Number</li> <li>End user name</li> <li>Master key ID</li> <li>License release date and time</li> <li>Update Number</li> <li>Purchasing Company name. In most cases the company to which the key was issued originally</li> </ul>	<ul style="list-style-type: none"> <li>Commercial product name</li> <li>Commercial product code</li> <li>Commercial product version and range of build numbers suitable for running</li> <li>Range of CHARON-AXP virtual models available for running</li> <li>Type of host CPU required</li> <li>Host operating system required</li> <li>Number of virtual CPUs enabled for virtual SMP systems</li> <li>Minimum number of host CPU cores required</li> <li>Minimum host memory required</li> <li>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.</li> <li>Maximum number of CHARON-AXP instances that can be run concurrently</li> <li>Whether or not CHAPI (CHARON-AXP API) can be used with this product</li> <li>Product and Field Test expiration dates (if any)</li> <li>Product and Field Test executions counter (if any)</li> <li>Maximum number of hosts that may run CHARON-AXP concurrently (in the case of a networking license)</li> <li>Level of support (if any), end date of any support contract, the "First Line" Service Provider</li> <li>Frequency of CHARON-AXP license checking during CHARON-AXP execution</li> </ul>	<ul style="list-style-type: none"> <li>Parameter that reduces the maximum speed of the program</li> <li>Parameter that prohibits use of Advanced CPU Emulation. If not present the Advanced CPU Emulation is enabled</li> </ul>

[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. In the current CHARON-AXP/VAX versions, a network license controls the maximum overall number of active instances, which can be distributed across client host systems according to the preference of the customer.

## 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 the installation of the Sentinel run-time environment.

Software licenses are best suited for stable environments, because their correct function depends on certain characteristics of the host system. Changing any of these characteristics will invalidate the license.

- If the CHARON host runs on real hardware, the software licenses are by default **tightly bound to the hardware** for which they were issued. If major hardware characteristics of the system are changed, the license will be disabled.
- If the CHARON host runs in a **virtual environment** (e.g. VMware), the software licenses are normally bound to the virtual machine ID and a set of additional characteristics of the virtual machine. If any of these parameters are changed, the license will be disabled.

For a more detailed description of the restrictions, please refer to the [Software Licensing restrictions](#) article or contact your Stromasys representative.

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

[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 by default and is able to display all available licenses with the "-all" parameter. It is also possible to check some specific license by its number using the "-key" parameter.

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 (except when the backup license is specified in its configuration file).

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	<p>A number (decimal Sentinel key ID) that specifies regular (N=0) and backup (N=1) license keys to be used by CHARON-AXP.</p> <p><b>Example:</b></p> <pre>set session license_key_id[0]=1877752571 set session license_key_id[1]=354850588</pre> <p>It is also possible to specify both regular and backup key in one line.</p> <p><b>Example:</b></p> <pre>set session license_key_id[0]=1877752571 license_key_id[1]=354850588</pre> <p>Depending on the presence of the regular and/or backup license key IDs in the configuration file, CHARON-AXP behaves differently:</p> <ol style="list-style-type: none"> <li><b>No keys are specified</b> CHARON-AXP behaves as usual (performs unqualified search for any suitable key). If no keys are found, CHARON-AXP exits.</li> <li><b>Both keys are specified</b> 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.</li> <li><b>Only regular key is specified</b> 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.</li> <li><b>Only backup key is specified</b> CHARON-AXP behaves as usual (performs unqualified search for any suitable key). If no keys are found, CHARON-AXP exits.</li> </ol>

[Back to Table of Contents](#)

## License installation

### Installation of Regular and Network license keys

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

- 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.
- 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-2.5-1.i386.rpm charon-license-4.8-18302.e171.x86_64.rpm
```

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

- On the server side (where the network license will reside):* open port 1947 for both TCP and UDP
- On the client side,* if broadcast search for remote licenses is to be used, UDP traffic from port 1947 of the license server to ports 30000-65535 of the client must be permitted.
- 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-license-<...>.rpm" containing the run-time customization) with the command

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

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

```
# rpm --nodeps -ihv aksusbd<...>.rpm charon-license-<...>.rpm
```

[Back to Table of Contents](#)

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

CHARON-AXP Software Licenses (SL) can be installed / updated according to the procedure described below.

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



### Sentinel Admin Control Center

Configuration for Sentinel License Manager on KIRILL

Options: Sentinel Keys, Products, Features, Sessions, Update/Attach, Access Log

Basic Settings | Users | **Access to Remote License Managers** | Access from Remote Clients | Detachable Licenses | Network

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 the "Submit" button to apply settings
- Select the "Network" tab.
- Set the "Network visibility" to "None":



### Sentinel Admin Control Center

Configuration for Sentinel License Manager on KIRILL

Options: Sentinel Keys, Products, Features, Sessions, Update/Attach, Access Log, **Configuration**, Diagnostics

Basic Settings | Users | Access to Remote License Managers | Access from Remote Clients | Detachable Licenses | **Network**

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

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

or collect the ".c2v" file in case a Software License is already installed or the 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.

The content of the installed software license is not shown by the Sentinel HASP Admin Control Center. To see it please run the "hasp\_srm\_view" utility from the 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 the server side (where the network license will reside):* open port 1947 for both TCP and UDP
- *On the client side,* if broadcast search for remote licenses is to be used, UDP traffic from port 1947 of the license server to ports 30000-65535 of the client must be permitted.
- *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. They 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 open the <http://localhost:1947> page.

The Sentinel Admin Control Center is not able to display CHARON-AXP licenses - to view key contents, use the "[hasp\\_srm\\_view](#)" utility.

To access the Sentinel Admin Control Center start any web browser, open the <http://localhost:1947> page. The web interface of the Sentinel Admin Control Center will appear.

The screenshot below gives an example:



## Sentinel Admin Control Center

Options

Sentinel Keys

Products

Features

Sessions

Update/Attach

Access Log

Configuration

Diagnostics

Help

About

### Sentinel Keys Available

#	Location	Vendor	Key ID	Key Type	Configuration	Version	Sessions	Actions
1	<a href="#">XEON4WAYW7</a>	68704	961833018	HASP HL NetTime 50	-	3.25	-	<input type="button" value="Browse"/> <input type="button" value="Net Features"/>
2	Local	68704	354850588	HASP HL NetTime 50	-	3.25	-	<input type="button" value="Products"/> <input type="button" value="Features"/> <input type="button" value="Sessions"/> <input type="button" value="Blink on"/>
3	Local	68704	1351199824	HASP HL Time	-	3.25	-	<input type="button" value="Products"/> <input type="button" value="Features"/> <input type="button" value="Sessions"/> <input type="button" value="Blink on"/>
4	<a href="#">rh64</a>	DEMOMA - evaluation	464243137687019632	HASP SL AdminMode Rehostable	-	2.31	1	<input type="button" value="Browse"/> <input type="button" value="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. A network key ("HASP-HL NetTime") on the host "XEON4WAYW7"
2. A network key installed locally
3. An HASP-HL installed locally
4. A network-wide software license on the host "RH64"

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

[Back to Table of Contents](#)

#### Disable remote keys access

A helpful feature of the 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, uncheck the "Allow Access to Remote Licenses" checkbox and press the "Submit" button to apply this setting:





- Options
- Sentinel Keys
- Products
- Features
- Sessions
- Update/Attach
- Access Log
- Configuration
- Diagnostics
- Help
- About

## Configuration for Sentinel License Manager

Basic Settings	Users	Access to Remote License Managers	Access from Remote Clients	Detachable Licenses
Allow Access to Remote Licenses <input type="checkbox"/> You may experience a delay of a few minutes before your changes will take effect.				
Broadcast Search for Remote Licenses <input checked="" type="checkbox"/>				
Aggressive Search for Remote Licenses <input type="checkbox"/>				
Specify Search Parameters <div style="border: 1px solid gray; height: 100px; width: 100%;"></div>				
<input type="button" value="Submit"/> <input type="button" value="Cancel"/> <input type="button" value="Set Defaults"/>				

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



- Options
- Sentinel Keys
- Products
- Features
- Sessions
- Update/Attach
- Access Log
- Configuration
- Diagnostics
- Help
- About

## Configuration for Sentinel License Manager

Basic Settings	Users	Access to Remote License Managers	Access from Remote Clients	Detachable Licenses
Allow Access from Remote Clients <input type="checkbox"/> You may experience a delay of a few minutes before your changes will take effect.				
Access Restrictions <div style="border: 1px solid gray; padding: 5px;">             allow=all           </div>				
<input type="button" value="Show Recent Client Access"/> The entries are evaluated in the order in which they are specified. As soon as a match is found, evaluation stops. <b>allow=all</b> is implicitly added to end of list				
<input type="button" value="Submit"/> <input type="button" value="Cancel"/> <input type="button" value="Set Defaults"/>				

[Back to Table of Contents](#)

## Accessing Sentinel Admin Control Center from remote hosts

By default, the Sentinel Admin Control Center forbids accessing its web interface from remote machines. To allow access, configure the 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 the Sentinel Admin Control Center run-time:

```
# systemctl restart aksusbd
```

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 the Linux documentation for details on how to configure the firewall. It is also possible to use SSH port forwarding with the following command (replace "CHARON\_MACHINE" by the real CHARON-VAX host name):

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

This will expose the Sentinel Admin Control Center on port 8080 to any computer, and it 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 the license(-s) content, to collect the key(s) status information and host fingerprint (C2V) files.

Applying updates (".v2c" files) is typically done using the Sentinel Admin Control Center (see above) but alternatively it is also possible to use the specific "hasp\_update" utility.

Please refer to the [Utilities](#) section of this Guide for more details.

[Back to Table of Contents](#)

## Removing CHARON-AXP software licenses

The following procedure must be applied to remove software license:

1. Using your web browser, open the <http://localhost:1947> page 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 Admin Control Center

#	Location	Vendor	Key ID	Key Type	Configuration	Version	Sessions	Actions
1	Local	68704	387285561437702475	HASP SL AdminMode Inactive (Cloned)		-	2.33	<a href="#">Certificates</a>

4. Note the name of the corresponding certificate and path to the certificates base in the "Certificates" section.
5. Remove the target certificate file from the specified directory, in most cases: "/var/hasplm/installed/68704/".
6. Reboot the CHARON host.
7. Start the "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-license-<...>.rpm" containing the run-time customization) using the following command:

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

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.
- A 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

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.
mtd	Used to create CHARON tape images from physical tapes and to write tape images back to physical tapes.
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).
CHARON Guest Utilities for OpenVMS	Used to manage virtual tapes and CHARON performance.

All these utilities (except for CHARON Guest Utilities for OpenVMS) are invoked from the Linux console command line.

# mkdiskcmd

## Table of Contents

- Description
- Creating disk images
- Transferring disk images

## Description

The "mkdiskcmd" 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.

## Creating disk images

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

```
$ mkdiskcmd --list
```

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

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

```
$ mkdiskcmd --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:

```

mkdisk for CHARON utility v. 1.14
Copyright: STROMASYS, 2015

Usage:
mkdiskcmd [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

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

Examples:
mkdiskcmd -h
mkdiskcmd -l
mkdiskcmd -a /opt/charon/utils/mkdisk.vtable -o /etc/rk07.vdisk -d rk07
mkdiskcmd -o /etc/custom.vdisk -z 512 -c 16384
mkdiskcmd -t /etc/rz22.vdisk rz25 -a /opt/charon/utils/mkdisk.vtable
mkdiskcmd -t /etc/rz22.vdisk rz25 -a /opt/charon/utils/mkdisk.vtable -z 512 -c 262134

```

The "--avtable" parameter is used to work with an alternative disk specification database (or to point to the standard database ("mkdisk.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 "mkdiskcmd" 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. The virtual machine 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:

```
$ mkdiskcmd --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 "mkdiskcmd --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:**

```
$ mkdiskcmd --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:

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

**Example:**

```
$ mkdiskcmd -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

## Table of Contents

- Description
- Tape container to physical tape transfer

## Description

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>
```

Parameters:

```
MTD - CHARON Magnetic Tape Dump & Restore utility, Version 2.3 (Build 18302)
Copyright (C) 2009-2016 STROMASYS SA. All rights reserved.

Usage: mtd [options] <tape-drive-name> <file-name> - dump tape content to file
mtd <file-name> <tape-drive-name> - restore dump to tape
mtd <file-name> <file-name> - convert formats
mtd <file-name> - examine tape dump and check integrity

<tape-drive-name> - tape drive
<file-name> - name of tape container file (.mtd or .vtape)

Options:
-l <file-name> - log file name (.log)
-n - do not rewind tape
-r <number> - number of attempts to retry failing tape reads
-i - ignore failing tape reads (implies -r 0)
-p - disable progress reporting
-v - enable verbose trace of data transfer (implies -p)
-s - output tape image in SMA format
-g - gather statistics and print upon completion
```

Example:

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

## Tape container to physical tape transfer

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/tapel.vtape /dev/st5
```

[Back to Table of Contents](#)



# hasp\_srm\_view

## Table of Contents

- Description
- Remote collection of status information

## Description

The "hasp\_srm\_view" utility displays the CHARON licenses content.

Run the utility with one of the following parameters to see the license(-s) details:

- "-l" (or without parameters) - CHARON default license details
- "-all" - all available CHARON licenses details
- "-key <key number>" - specific CHARON license (defined by its "key number") details

The "hasp\_srm\_view" utility provides the following functionality:

- Display the licenses details. It is possible to view all available license or some specific one.
- Collecting license status information
- Collecting host fingerprint information

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

```
# hasp_srm_view -help

CHARON Sentinel HASP utility
Copyright: STROMASYS, 2016

Options:
-? or -h or -help - to see help screen

-l - to see CHARON license details (for default key)
-all - to see CHARON license details (for all available keys)
-key <key number> - to see CHARON license details (for specific key)

-c2v <C2V file> - to collect the key status information (C2V file)
-c2v <C2V file> -key <key number> - to collect C2V file for specific local key
-fgp <C2V file> - to collect the host fingerprint information (C2V file)
```

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](#)

# hasp\_update

## Table of Contents

- Description
- Usage

## Description

The "hasp\_update" is a Sentinel standard utility for license management included in the CHARON 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
```

## Usage

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](#)

## Table of Contents

- Description
- Dedication of a host physical interface to CHARON
- Release of a host physical interface back to host
- Creation of a virtual network
- Removal of a virtual network
- Adding VLAN interface
- Removing VLAN interface

## Description

The "ncu" ("Network Control Utility") is used to dedicate a host interface to CHARON, 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.

The NetworkManager service must be running in order to have all the functionalities enabled with "ncu". If the service is not enabled please configure the network using the manual operations described in the section "Manual configuration of CHARON networking" of the Installation chapter of this Guide.

## 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) 2016 Version 1.6

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
===== VLAN =====
=====

select action:

1 - Dedicate to CHARON
2 - Release to host
3 - Create Bridge with TAPs
4 - Remove Bridge
5 - Add VLAN
6 - Remove VLAN
7 - Print status
8 - Exit

:> 8
```

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. Add VLAN interface
6. Remove VLAN interface
7. Print status - use it to display status of network interfaces and the menu shown above
8. 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. Enter "1", 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 - Add VLAN
6 - Remove VLAN
7 - Print status
8 - Exit

:> 7
```

Now the interface "eth1" is dedicated to CHARON:

```
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
=====
===== VLAN =====
=====

select action:

1 - Dedicate to CHARON
2 - Release to host
3 - Create Bridge with TAPs
4 - Remove Bridge
5 - Add VLAN
6 - Remove VLAN
7 - Print status
8 - Exit
```

Enter "8" to return to console prompt.

Now "eth1" can be used by CHARON.

[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) 2016 Version 1.6

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
===== VLAN =====
=====

select action:

1 - Dedicate to CHARON
2 - Release to host
3 - Create Bridge with TAPs
4 - Remove Bridge
5 - Add VLAN
6 - Remove VLAN
7 - Print status
8 - 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 - Add VLAN
6 - Remove VLAN
7 - Print status
8 - Exit

:> 8
```

Enter "8" 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) 2016 Version 1.6

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
===== VLAN =====
=====

select action:

1 - Dedicate to CHARON
2 - Release to host
3 - Create Bridge with TAPs
4 - Remove Bridge
5 - Add VLAN
6 - Remove VLAN
7 - Print status
8 - 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 - Add VLAN
6 - Remove VLAN
7 - Print status
8 - Exit

:> 7
```

Now enter "7" 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            CHARON            connected to host
tap1            bridge           connected to bridge
=====
bridge name      bridge id          STP enabled      interfaces
br0_eth1        8000.768e1ea091d9 no                 eth1
tap0
tap1
===== VLAN =====
=====

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

:> 8

```

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 "8" 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) 2016 Version 1.6

Interfaces    Dedicated to    State
-----
eth0          host           connected to host
eth1          bridge         connected to bridge
lo            host           unmanaged from host
tap0          CHARON         connected to host
tap1          bridge         connected to bridge
=====
bridge name    bridge id      STP enabled    interfaces
br0_eth1      8000.768e1ea091d9  no            eth1
tap0
tap1
===== VLAN =====
=====

select action:

1 - Dedicate to CHARON
2 - Release to host
3 - Create Bridge with TAPs
4 - Remove Bridge
5 - Add VLAN
6 - Remove VLAN
7 - Print status
8 - 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 - Add VLAN
6 - Remove VLAN
7 - Print status
8 - Exit

:> 8
```

Enter "8" to quit "ncu" utility.

[Back to Table of Contents](#)



## Adding VLAN interface

Login a root. Start "ncu" utility:

```
# ncu
CHARON Network Configuration Utility, STROMASYS (c) 2016 Version 1.6

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
===== VLAN =====
=====

select action:

1 - Dedicate to CHARON
2 - Release to host
3 - Create Bridge with TAPs
4 - Remove Bridge
5 - Add VLAN
6 - Remove VLAN
7 - Print status
8 - Exit

:> 5
```

Enter "5" then enter:

1. The physical interface name to be used for creating VLAN
2. The ID of the VLAN device
3. IP address of the VLAN device. Skip this step if no IP is required
4. Network mask of the VLAN device. Enter for no network mask.

```
Specify the phys interface used for VLAN:eth1
Specify the id of VLAN device (<4095):111
Specify the ip address of VLAN device or empty string for no ip address: 192.168.1.100
Specify the netmask address of VLAN device or empty string for no netmask:
225.225.225.0

select action:

1 - Dedicate to CHARON
2 - Release to host
3 - Create Bridge with TAPs
4 - Remove Bridge
5 - Add VLAN
6 - Remove VLAN
7 - Print status
8 - Exit

:> 8
```

Enter "8" to quit "ncu" utility.

[Back to Table of Contents](#)

## Removing VLAN interface

Login a root. Start "ncu" utility:

```
# ncu
CHARON Network Configuration Utility, STROMASYS (c) 2016 Version 1.6

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
=====
eth1.111
=====

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

:> 6
```

Enter "6" then enter the VLAN interface for remove:

```
Specify the VLAN interface, which be removed: eth1.111

Removed VLAN -:eth1.111:-

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

:> 8
```

Enter "8" to quit "ncu" utility.

[Back to Table of Contents](#)

# CHARON Guest Utilities for OpenVMS

## Table of Contents

- Description
- Installation
- Performance optimization
- Virtual tapes management
- Defining keys
- Displaying version

## Description

The "CHARON Guest Utilities for OpenVMS" (CHARONCP) package contains several utilities for managing virtual tapes, changing the emulator speed and creating useful definitions for that operations.

This set of utilities is located in the "charoncp011.vdisk" disk file in the "/opt/charon/disks" folder.

**Supported OpenVMS versions** (depending on platform support): OpenVMS 6.1 and above.

 In case of OpenVMS upgrade, CHARONCP will have to be re-installed.

## Installation

Specify this image in the CHARON configuration file, boot from the system disk and mount the disk with the following OpenVMS command:

```
$ MOUNT <device name> /OVERRIDE=IDENTIFICATION
```

Issue the following commands to install the package (example given for OpenVMS V8.4):

```
$ @SYS$UPDATE:VMSINSTAL
...
* Are you satisfied with the backup of your system disk [YES]? YES
* Where will the distribution volumes be mounted: <device name>:[CHARONCP011.KIT]

Enter the products to be processed from the first distribution volume set.
* Products: CHARONCP011

* Enter installation options you wish to use (none): <press enter>
...
Do you want to install this product [NO]? YES
...
* Where should the CHARONCP root directory be located ? [SYS$SYSDEVICE:[CHARONCP]]: <press enter>
...
* Do you want to purge files replaced by this installation [YES]? <press enter>
```

Select all the components included to the package:

```

Component Selection

Select the CHARONCP components you wish to install from the menu below.
An asterisk appears next to the packages that have already been
selected. You can remove a package from the list by selecting it
again. You may enter more than one selection by separating your
choices with commas.

1. [*] CHARONCP Guest Utility (REQUIRED)
2. [*] Compatability Utilities
3. [*] Install DCL Commands & Help

4. Exit

* Your choice [4]: 1,2,3

...

* Your choice [4]: 4

...

* Is this correct [YES]: <press enter>

...

* Products: <press enter>

VMSINSTAL procedure done at hh:mn

$

```

Proceed with installation using all the default options.

Once the installation is completed, add the following line to the "SYS\$STARTUP:SYSTARTUP\_VMS.COM" ("SYS\$STARTUP:SYSTARTUP\_V5.COM" for VMS 5.5) file for the package to be loaded automatically at system startup:

```
$ @SYS$STARTUP:CHARONCP_STARTUP
```

After that the package will be loaded automatically on startup.

[Back to Table of Contents](#)

## Performance optimization

CHARON takes 100% of host CPU even in case of idle state of guest OpenVMS operating system. To get rid of such resources consumption there is a specific option provided by CHARON Guest Utilities - "idle" mode.

\$ CHARONCP SET IDLE /ENABLE	Loads the OpenVMS idle loop detection software. This allows CHARON to detect when the emulated CPU(s) are idle and use the host power saving instructions to reduce power usage.
\$ CHARONCP SET IDLE /DISABLE	Unloads the OpenVMS idle loop detection software.

[Back to Table of Contents](#)

## Virtual tapes management

Specify mapping to tape container in the following way in the CHARON configuration file:

```
set <adapter name> container[<unit name>] = ".vtape" removable[<unit name>] = true
```

### Example:

```
set PKA container[600] = ".vtape" removable[600] = true
```

Once it is done using the following commands it is possible to manage virtual tapes attached to CHARON:

<pre>\$ CHARONCP SET MAGTAPE &lt;device&gt; /LOAD="&lt;filename&gt;.vtape"</pre>	<p>Create the specified host-file (if it does not already exist) and attach it to the specified virtual tape device.</p> <p><b>Example:</b></p> <pre>\$ CHARONCP SET MAGTAPE MKA600: /LOAD="backup_01.vtape"</pre>
<pre>\$ CHARONCP SET MAGTAPE &lt;device&gt; /UNLOAD</pre>	<p>Detach any file currently attached to the specified virtual tape device.</p> <p><b>Example:</b></p> <pre>\$ CHARONCP SET MAGTAPE MKA600: /UNLOAD</pre>

### Possible errors:

BADFILENAME	The filename specified as a value to the qualifier /LOAD was either too long or does not have a file extension of ".vtape".
DEVNOTDISM	<p>Attempting to execute a SET MAGTAPE/LOAD when a file is already attached.</p> <p>Perform a SET MAGTAPE/UNLOAD first. If a SET MAGTAPE/LOAD command has not previously been executed, then the CHARON configuration container specification for the tape device may contain a full path. Doing this will create and attach an initial tape container file. To avoid this, remove the file name from the specification (leaving only a file extension of ".vtape" and optional directory).</p>

**i** If some tape container has been already specified in the CHARON configuration file use the command "CHARONCP SET MAGTAPE <device> /UNLOAD" to unload it first.

[Back to Table of Contents](#)

## Defining keys

It is possible to define certain keys on the terminal keyboard for fast access to the CHARONCP functionality while you are in CHARONCP:

<pre>\$ CHARONCP CHARONCP&gt; DEFINE /KEY &lt;key-name&gt; &gt; &lt;equivalence-string&gt;</pre>	<p>Defines an equivalence string and a set of attributes with a key on the terminal keyboard.</p> <p>You can have a set of keys defined automatically for use with the CHARONCP utility by placing DEFINE/KEY commands in the file SYS\$LOGIN:CHARONCP_KEYDEFS.INI</p> <p><b>Example:</b></p> <pre>\$ CHARONCP CHARONCP&gt; DEFINE /KEY F1 "SET MAGTAPE MKA600: /UNLOAD"</pre>
<pre>\$ CHARONCP CHARONCP&gt; SHOW KEY &lt;key-name&gt;</pre>	<p>Displays key definitions created with the DEFINE/KEY command. Refer to the DCL help entry for SHOW KEY for further information.</p> <p><b>Example:</b></p> <pre>\$ CHARONCP CHARONCP&gt; SHOW KEY F1 DEFAULT key state definitions: F1 = "set magtape mka600: /unload" CHARONCP&gt;</pre>

 For more information refer to the OpenVMS DCL Dictionary (DEFINE/KEY section).

[Back to Table of Contents](#)

## Displaying version

<pre>\$ CHARONCP SHOW VERSION</pre>	<p>Displays the CHARONCP package version number and architecture. This can be useful for customers reporting issues with the CHARONCP software.</p> <p><b>Example:</b></p> <pre>\$ CHARONCP SHOW VERSION  CHARONCP version id is: V1.1</pre>
-------------------------------------	----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

[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

- General Settings
- Core Devices
- Console
- Placement of peripheral devices on PCI bus
- Disks and tapes
  - KZPBA PCI SCSI adapter
  - KGPSA-CA PCI Fibre Channel adapter
  - Acer Labs 1543C IDE/ATAPI CD-ROM adapter
  - PCI I/O Bypass controller
  - Finding the target "/dev/sg" device
- Networking
- Sample configuration files
  - HP AlphaServer 800 configuration file
  - HP AlphaServer 4000 configuration file
  - HP AlphaServer DS20 configuration file
  - HP AlphaServer ES40 configuration file
  - HP AlphaServer GS80 configuration file

[Back to Table of Contents](#)

# General Settings

## Table of Contents

- Session
  - hw\_model
  - configuration\_name
  - log
  - log\_method
  - log\_file\_size
  - log\_rotation\_period
  - license\_key\_id
  - license\_key\_lookup\_retry
  - affinity
  - n\_of\_cpus
  - n\_of\_io\_cpus
- File inclusion

## 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

<b>Parameter</b>	hw_model
<b>Type</b>	Text string
<b>Value</b>	<p>Virtual HP Alpha system hardware model to be emulated.</p> <p>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.</p> <p><b>Example:</b></p> <pre>set session hw_model="AlphaServer_ES40"</pre> <p>Available models are:</p> <ul style="list-style-type: none"> <li>• AlphaServer_AS400</li> <li>• AlphaServer_AS800</li> <li>• AlphaServer_AS1000</li> <li>• AlphaServer_AS1000A</li> <li>• AlphaServer_AS1200</li> <li>• AlphaServer_AS2000</li> <li>• AlphaServer_AS2100</li> <li>• AlphaServer_AS4000</li> <li>• AlphaServer_AS4100</li> <li>• AlphaServer_DS10</li> <li>• AlphaServer_DS15</li> <li>• AlphaServer_DS20</li> <li>• AlphaServer_DS25</li> <li>• AlphaServer_ES40</li> <li>• AlphaServer_ES45</li> <li>• AlphaServer_GS80</li> <li>• AlphaServer_GS160</li> <li>• AlphaServer_GS320</li> </ul> <p> Refer to <a href="#">this section</a> to find how to set a particular HP Alpha model supported by CHARON-AXP.</p>



**configuration\_name**

<b>Parameter</b>	configuration_name
<b>Type</b>	Text string
<b>Value</b>	<p>Name of the CHARON-AXP instance (it must be unique):</p> <pre>set session configuration_name="MSCDV1"</pre> <p>The value of this parameter is used as a prefix to the event log file name. (see below).</p> <p>From the example above, the CHARON-AXP log file will have the following name:</p> <pre>MSCDV1-YYYY-MM-DD-hh-mm-ss-xxxxxxxxxx.log</pre> <p>xxxxxxxx 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).</p> <p> It is strictly recommended to use the "configuration_name" parameter if more than one CHARON instance runs on the same server.</p>


## log

<b>Parameter</b>	log
<b>Type</b>	Text string
<b>Value</b>	<p>The log file or directory name is where the log file for each CHARON-AXP execution session is stored.</p> <div style="border: 1px solid #0070C0; padding: 10px;"> <h3 style="text-align: center; background-color: #0070C0; color: white; margin: 0;">Log specified as a file name</h3> <p>It is possible to overwrite the existing log file or to extend it using the "log_method" parameter.</p> <p><b>i</b> The "log_method" parameter is effective only when a single log file is specified, not a directory.</p> <p><b>Example:</b></p> <pre style="border: 1px solid #ccc; padding: 5px;">set session log="/charon/es40prod.log"</pre> </div> <div style="border: 1px solid #0070C0; padding: 10px; margin-top: 10px;"> <h3 style="text-align: center; background-color: #0070C0; color: white; margin: 0;">Log specified as a directory</h3> <p>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. In these two cases, the log rotation mode is enabled, meaning a new log file is created each time the virtual machine is started and when the log file size exceeds the one specified (see <a href="#">log_file_size</a>) and/or when the log file is older than a specified number of days (see <a href="#">log_rotation_period</a>).</p> <p><b>i</b> A symbolic link located in the same directory will be created, pointing to the active log file. Its name is based on the <code>hw_model</code> parameter or the <code>configuration_name</code> parameter if specified.</p> <p>If the "configuration_name" parameter of the session is specified, the log file name is composed as follows:</p> <pre style="border: 1px solid #ccc; padding: 5px;">&lt;configuration_name&gt;-YYYY-MM-DD-hh-mm-ss-xxxxxxxxx.log</pre> <p>If the "configuration_name" parameter is omitted, the log file name will have the following format:</p> <pre style="border: 1px solid #ccc; padding: 5px;">&lt;hw_model&gt;-YYYY-MM-DD-hh-mm-ss-xxxxxxxxx.log</pre> <p>where "xxxxxxxxx" is an increasing decimal integer, starting from 00000000 to separate log files with the same time of creation (in case the log is being created faster than one log file per second).</p> <p><b>!</b> Only existing directory can be specified. If the directory specified does not exist, this will be considered as a flat file.</p> <p><b>Example:</b></p> <pre style="border: 1px solid #ccc; padding: 5px;">set session configuration_name="es40prod" set session log="/charon/logs"</pre> <p>The execution of the virtual machine will create a log file, named <code>/charon/logs/es40prod-2016-10-13-10-00-00-000000000.log</code> (for example) and a symbolic link named <code>/charon/logs/es40prod.log</code> pointing to this file. The link will be updated when the log rotation will occur.</p> </div>

## log\_method

<b>Parameter</b>	log_method
<b>Type</b>	Text string
<b>Value</b>	<ul style="list-style-type: none"> <li>"append" (default)</li> <li>"overwrite"</li> </ul> <p>Determines if the previous log information is maintained or overwritten.</p> <div style="border: 1px solid black; padding: 5px; margin: 10px 0;"> <p>This parameter must be specified only in addition to "log" parameter on the same line.</p> </div> <p>This parameter is applicable only if the CHARON-AXP log is stored to a file that is specified explicitly with the "log" parameter.</p> <p><b>Example:</b></p> <div style="border: 1px solid black; padding: 5px; margin: 10px 0;"> <pre>set session log="log.txt" log_method="overwrite"</pre> </div>

## log\_file\_size

<b>Parameter</b>	log_file_size
<b>Type</b>	Text string
<b>Value</b>	<p>If log rotation is enabled, the log_file_size parameter determines the log file size threshold at which the log is automatically rotated.</p> <ul style="list-style-type: none"> <li>"unlimited" or "0" (default) - the feature is disabled</li> <li>"default" - default size is used (4Mb)</li> <li>&lt;size&gt;[KMG] - size of the current log file in bytes with additional multipliers: <ul style="list-style-type: none"> <li>K - Kilobyte - multiply by 1024</li> <li>M - Megabyte - multiply by 1024*1024</li> <li>G - Gigabyte - multiply by 1024*1024*1024</li> </ul> </li> </ul> <p><b>Examples:</b></p> <div style="border: 1px solid black; padding: 5px; margin: 10px 0;"> <pre>set session log_file_size="default"</pre> </div> <div style="border: 1px solid black; padding: 5px; margin: 10px 0;"> <pre>set session log_file_size=10M</pre> </div> <p> Minimum LOG File size is 64K, maximum is 1G. Setting size less than 64K effectively makes the LOG File unlimited.</p>

## log\_rotation\_period

<b>Parameter</b>	log_rotation_period
<b>Type</b>	Text string
<b>Value</b>	<ul style="list-style-type: none"> <li>"default" - default value, 7 days. This values is used even if the "log_rotation_period" is not specified.</li> <li>"daily" or "1"</li> <li>"weekly" or "7"</li> <li>"never" or "0"</li> <li>&lt;N&gt; - in N days</li> </ul> <p>If the rotation log mode is enabled this parameter controls switching to next log file based on period of time passed. If enabled the switching to next log file appears at midnight.</p> <p><b>Examples:</b></p> <div style="border: 1px solid black; padding: 5px; margin: 10px 0;"> <pre>set session log_rotation_period="weekly"</pre> </div> <div style="border: 1px solid black; padding: 5px; margin: 10px 0;"> <pre>set session log_rotation_period=14</pre> </div>

## license\_key\_id

<b>Parameter</b>	license_key_id[N] N=0 or 1
<b>Type</b>	Numeric
<b>Value</b>	<p>An integer (decimal Sentinel Key ID) that specifies the regular (N=0) and backup (N=1) license keys to be used by CHARON-AXP.</p> <pre>set session license_key_id[0]=1877752571 set session license_key_id[1]=354850588</pre> <p>It is also possible to specify both regular and backup keys in one line:</p> <pre>set session license_key_id[0]=1877752571 license_key_id[1]=354850588</pre> <p>Based on the presence of the regular and/or backup license key IDs in the configuration file, CHARON-AXP behaves as follows:</p> <ol style="list-style-type: none"> <li><b>No keys are specified</b> CHARON-AXP performs an unqualified search for any suitable key. If no key is found, CHARON-AXP exits.</li> <li><b>Both keys are specified</b> 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.</li> <li><b>Only regular key is specified</b> 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.</li> <li><b>Only backup key is specified</b> CHARON-AXP performs an unqualified search for any suitable key. If no key is found, CHARON-AXP exits.</li> </ol>

## license\_key\_lookup\_retry

<b>Parameter</b>	license_key_lookup_retry
<b>Type</b>	Text String
<b>Value</b>	<p>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.</p> <p>Syntax:</p> <pre>set session license_key_lookup_retry = "N [, T]"</pre> <p>Options:</p> <ul style="list-style-type: none"> <li>• N - Number of retries to look for license keys.</li> <li>• T - Time between retries in seconds. If not specified 60 seconds are used</li> </ul> <p><b>Example 1</b></p> <pre>set session license_key_lookup_retry = 1</pre> <p>If license key is not found during initial scan, do only one more attempt after 60 seconds.</p> <p><b>Example 2</b></p> <pre>set session license_key_lookup_retry = "1,30"</pre> <p>Same as above but retry in 30 seconds.</p> <p><b>Example 3</b></p> <pre>set session license_key_lookup_retry = "3,10"</pre> <p>If license key is not found during initial scan, do 3 more attempts waiting 10 seconds between them.</p> <p><b>Example 4</b></p> <pre>set session license_key_lookup_retry = "5"</pre> <p>If license key is not found during the initial scan, do 5 more attempts waiting 60 seconds between them.</p>

## affinity

<b>Parameter</b>	affinity
<b>Type</b>	Text string
<b>Value</b>	<p>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.</p> <p>Used for soft partitioning of the host CPU resources and/or for isolating host CPUs for other applications.</p> <p>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.</p> <div style="border: 1px solid #f0e68c; padding: 10px; margin: 10px 0;"> <p>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).</p> </div> <p>Host CPUs are enumerated as a comma separated list of host system assigned CPU numbers:</p> <pre>set session affinity="0, 2, 4, 6"</pre>

## n\_of\_cpus

<b>Parameter</b>	n_of_cpus																																						
<b>Type</b>	Numeric																																						
<b>Value</b>	<p>Limits the number of emulated CPUs.</p> <p><b>Example:</b></p> <pre style="border: 1px solid black; padding: 5px;">set session n_of_cpus=3</pre> <p>The maximum number of CPUs enabled by CHARON-AXP is specified by the license key, but cannot exceed the original hardware restrictions:</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="text-align: left;">HP Alpha Model</th> <th style="text-align: left;">Number of emulated CPUs</th> </tr> </thead> <tbody> <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> </tbody> </table>	HP Alpha Model	Number of emulated CPUs	AlphaServer_AS400	1	AlphaServer_AS800	1	AlphaServer_AS1000	1	AlphaServer_AS1000A	1	AlphaServer_AS1200	2	AlphaServer_AS2000	2	AlphaServer_AS2100	4	AlphaServer_AS4000	2	AlphaServer_AS4100	4	AlphaServer_DS10	1	AlphaServer_DS15	1	AlphaServer_DS20	2	AlphaServer_DS25	2	AlphaServer_ES40	4	AlphaServer_ES45	4	AlphaServer_GS80	8	AlphaServer_GS160	16	AlphaServer_GS320	32
HP Alpha Model	Number of emulated CPUs																																						
AlphaServer_AS400	1																																						
AlphaServer_AS800	1																																						
AlphaServer_AS1000	1																																						
AlphaServer_AS1000A	1																																						
AlphaServer_AS1200	2																																						
AlphaServer_AS2000	2																																						
AlphaServer_AS2100	4																																						
AlphaServer_AS4000	2																																						
AlphaServer_AS4100	4																																						
AlphaServer_DS10	1																																						
AlphaServer_DS15	1																																						
AlphaServer_DS20	2																																						
AlphaServer_DS25	2																																						
AlphaServer_ES40	4																																						
AlphaServer_ES45	4																																						
AlphaServer_GS80	8																																						
AlphaServer_GS160	16																																						
AlphaServer_GS320	32																																						

## n\_of\_io\_cpus

<b>Parameter</b>	n_of_io_cpus
<b>Type</b>	Numeric
<b>Value</b>	<p>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.</p> <p>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.</p> <p><b>Example:</b></p> <pre style="border: 1px solid black; padding: 5px;">set session n_of_io_cpus=2</pre>

## File inclusion

It is possible to include a configuration file into an existing one using the "include" command. The file extension is usually `.icfg`.

### Format:

```
include "file.icfg"
```

### Example:

```
include "/charon/commonpart.icfg"
```

# Core Devices

## Table of Contents

- CPU
  - enabled
  - cpu\_architecture
  - cache\_size
- RAM
  - size
- TOY
  - container
  - sync\_to\_host
- ROM
  - container
  - system\_name
  - system\_serial\_number
  - dsrdb
  - version
- Virtual HP Alpha interval timer
- Setting of a particular HP Alpha model
  - HP AlphaStation 200 - 400
  - HP AlphaServer 600 - 800
  - HP AlphaServer 1000
  - HP AlphaServer 1000A
  - HP AlphaServer 1200 and AlphaStation 1200
  - HP AlphaServer 2000
  - HP AlphaServer 2100
  - HP AlphaServer 4000
  - HP AlphaServer 4100
  - HP AlphaServer/AlphaStation DS10 and HP AlphaServer DS10L
  - HP AlphaServer DS15 and HP AlphaStation DS15
  - HP AlphaServer DS20 and HP AlphaStation DS20
  - HP AlphaServer DS25 and HP AlphaStation DS25
  - HP AlphaServer ES40 and AlphaStation ES40
  - HP AlphaServer ES45
  - HP AlphaServer GS80
  - HP AlphaServer GS160
  - HP AlphaServer GS320
- Auto Boot
  - auto\_action restart
- Setting System Marketing Model (SMM)

[Back to Table of Contents](#)



## CPU

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

### enabled

<b>Parameter</b>	enabled
<b>Type</b>	Boolean
<b>Value</b>	<p>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).</p> <p>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.</p> <p>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.</p> <p><b>Example:</b></p> <pre>set ace enabled = false</pre> <p>"set ace enabled=true" is ignored when the license does not permit ACE operation.</p>

### cpu\_architecture

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

### cache\_size

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

[Back to Table of Contents](#)

## RAM

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

### size

<b>Parameter</b>	size
<b>Type</b>	Numeric
<b>Value</b>	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

<b>Parameter</b>	container
<b>Type</b>	Text string
<b>Value</b>	<p>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).</p> <p>By default it is left unspecified.</p> <p><b>Example:</b></p> <pre>set toy container="/Charon/my_virtual_system.dat"</pre> <p> It is recommended to specify the full path to the TOY file.</p>

## sync\_to\_host

<b>Parameter</b>	sync_to_host														
<b>Type</b>	Text string														
<b>Value</b>	<p>Specifies whether and how the guest OS time is synchronized with the CHARON host time.</p> <p>Syntax:</p> <pre>set TOY sync_to_host = "{as_vms   as_tru64   as_is}[ , nowrite]"</pre> <p>where:</p> <table border="1"> <thead> <tr> <th>Parameter</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>as_vms</td> <td>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.</td> </tr> <tr> <td>as_tru64</td> <td>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.</td> </tr> <tr> <td>as_is</td> <td>If the TOY date and time must be set to the host's UTC date and time.</td> </tr> <tr> <td>nowrite</td> <td>Forbid updates to the TOY from the guest OS.</td> </tr> </tbody> </table> <p><b>Example:</b></p> <pre>set TOY sync_to_host = "as_vms, nowrite"</pre> <p>To synchronize the guest OS with TOY, use the following commands (from "SYSTEM"/"root" account):</p> <table border="1"> <thead> <tr> <th>On OpenVMS/AXP</th> <th>On Tru64 UNIX</th> </tr> </thead> <tbody> <tr> <td><pre>\$ set time</pre></td> <td><pre># date -u `consvar -g date   cut -f 3 -d ' '`</pre></td> </tr> </tbody> </table> <p>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.</p> <p> If "sync_to_host" parameter is specified there is no need to specify "container" parameter in addition.</p>	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	Forbid updates to the TOY from the guest OS.	On OpenVMS/AXP	On Tru64 UNIX	<pre>\$ set time</pre>	<pre># date -u `consvar -g date   cut -f 3 -d ' '`</pre>
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	Forbid updates to the TOY from the guest OS.														
On OpenVMS/AXP	On Tru64 UNIX														
<pre>\$ set time</pre>	<pre># date -u `consvar -g date   cut -f 3 -d ' '`</pre>														


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

<b>Parameter</b>	container
<b>Type</b>	Text string
<b>Value</b>	<p>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.</p> <p>By default it is left unspecified.</p> <p> it is recommended to specify the full path to this file</p> <p><b>Example:</b></p> <pre>set rom container="/Charon/my_virtual_system.rom"</pre>

### system\_name

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

### system\_serial\_number

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

## dsrdb

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

## version

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

[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).

<b>Parameter</b>	clock_period
<b>Type</b>	Numeric
<b>Value</b>	<p>Specifies period of interval timer, in microseconds. Only two values are supported:</p> <ul style="list-style-type: none"> <li>• 10000 (which corresponds to 100Hz interval timer)</li> <li>• 1000 (which corresponds to 1000Hz interval timer)</li> </ul> <p>By default it is set to 10000.</p>

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

<b>Parameter</b>	auto_action restart
<b>Type</b>	Text string
<b>Value</b>	<p>Determines whether CHARON-AXP boots automatically if the correct boot flags are set (and saved in the HP Alpha console files).</p> <p><b>Example:</b></p> <pre>&gt;&gt;&gt;set bootdef_dev dka0 &gt;&gt;&gt;set auto_action restart</pre>

[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

- General Description
- Console Parameters
  - communication
  - line
  - Example
- Mapping Serial line controllers to system resources
  - physical\_serial\_line
    - line
    - baud
    - break\_on
    - stop\_on
    - log
    - log\_file\_size
    - Example
  - virtual\_serial\_line
    - host
    - port
    - break\_on
    - stop\_on
    - log
    - log\_file\_size
    - Example
    - Notes on "virtual\_serial\_line" options
  - operator\_console
    - break\_on, stop\_on
    - Example
  - "ttyY" notation specifics

[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

<b>Parameter</b>	communication
<b>Type</b>	Text string
<b>Value</b>	<ul style="list-style-type: none"> <li>"ascii" - for connection to terminals (default)</li> <li>"binary" - for binary (packet) protocols, which are used mainly for communicating with PLCs</li> </ul>

## line

<b>Parameter</b>	line
<b>Type</b>	Identifier
<b>Value</b>	<p>This parameter is used to connect a particular serial line mapping interface to the controller.</p> <p><b>Example:</b></p> <pre>set COM2 line=TTA0</pre>

## 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

<b>Parameter</b>	line										
<b>Type</b>	Text string										
<b>Value</b>	<p>A defined TTY port on host system:</p> <table border="1"> <thead> <tr> <th>Syntax</th> <th>Description</th> </tr> </thead> <tbody> <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&lt;XXX&gt;"</td> <td>Proprietary (depending on a driver) devices such as DIGI or MOXA cards</td> </tr> </tbody> </table> <p>A specific account for running CHARON ("charon") does not allow usage of physical consoles "/dev/tty&lt;N&gt;" as CHARON consoles. If you plan to map CHARON console to "/dev/tty&lt;N&gt;" use only "root" account for CHARON running.</p>	Syntax	Description	"/dev/tty<N>"	Virtual serial lines	"/dev/ttyS<N>"	Onboard serial lines	"/dev/ttyUSB<N>"	Modem or USB serial lines adapters	"/dev/tty<XXX>"	Proprietary (depending on a driver) devices such as DIGI or MOXA cards
Syntax	Description										
"/dev/tty<N>"	Virtual serial lines										
"/dev/ttyS<N>"	Onboard serial lines										
"/dev/ttyUSB<N>"	Modem or USB serial lines adapters										
"/dev/tty<XXX>"	Proprietary (depending on a driver) devices such as DIGI or MOXA cards										


### baud

<b>Parameter</b>	baud
<b>Type</b>	Numeric
<b>Value</b>	<p>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.</p> <p><b>Example:</b></p> <pre>set OPA0 baud=38400</pre>

## break\_on

<b>Parameter</b>	break_on
<b>Type</b>	Text string
<b>Value</b>	<p>Specifies what byte sequences received over the physical serial line will trigger a HALT command.</p> <p>This parameter works only for the console line.</p> <p>Specify the following values: "Ctrl-P" , "Break" or "none" ("none" disables triggering a HALT condition).</p> <div style="border: 1px solid #fde725; padding: 10px; margin: 10px 0;"> <p>If your guest operating system is OpenVMS in addition to "none" setting you have to set a specific console parameter "controlp" to "off" in the following way:</p> <pre style="border: 1px solid #ccc; padding: 5px; margin: 5px 0;">&gt;&gt;&gt; set controlp off &gt;&gt;&gt; power off</pre> <p>The second line is to preserve the ROM settings.</p> </div> <p><b>Example:</b></p> <pre style="border: 1px solid #ccc; padding: 5px; margin: 5px 0;">set OPA0 break_on="Ctrl-P"</pre> <p>The default value is "Break".</p> <p> This parameter can be specified only for COM1 (OPA0) console</p>


## stop\_on

<b>Parameter</b>	stop_on
<b>Type</b>	Text string
<b>Value</b>	<p>Specifies what byte sequences received over the physical serial line will trigger a STOP condition. The STOP condition causes CHARON-AXP to exit.</p> <p>Specify either "F6" or "none" ("none" disables triggering a STOP condition).</p> <p><b>Example:</b></p> <pre style="border: 1px solid #ccc; padding: 5px; margin: 5px 0;">set OPA0 stop_on="F6"</pre> <p>The default value is "none".</p> <p>Setting "F6" triggers the STOP condition upon receipt of the "&lt;ESC&gt;[17~" sequence. Terminals usually send these sequences by pressing the <b>F6</b> button</p> <p> This parameter can be specified only for COM1 (OPA0) console</p>

## log

<b>Parameter</b>	log
<b>Type</b>	Text string
<b>Value</b>	<p>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.</p> <p>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.</p> <p><b>Examples:</b></p> <pre>set OPA0 log="log.txt"</pre> <pre>set OPA0 log="/Charon/Logs"</pre> <p>Only existing directory can be used as a value of the "log" parameter.</p>

## log\_file\_size

<b>Parameter</b>	log_file_size
<b>Type</b>	Text string
<b>Value</b>	<p>If log rotation is enabled, the log_file_size parameter determines the log file size threshold at which the log is automatically rotated.</p> <ul style="list-style-type: none"> <li>"unlimited" or "0" (default) - the feature is disabled</li> <li>"default" - default size is used (4Mb)</li> <li>&lt;size&gt;[KMG] - size of the current log file in bytes with additional multipliers: <ul style="list-style-type: none"> <li>K - Kilobyte - multiply by 1024</li> <li>M - Megabyte - multiply by 1024*1024</li> <li>G - Gigabyte - multiply by 1024*1024*1024</li> </ul> </li> </ul> <p><b>Examples:</b></p> <pre>set OPA0 log_file_size="default"</pre> <pre>set OPA0 log_file_size=10M</pre> <p> Minimum log file size is 64K, maximum is 1G. Setting size less than 64K effectively makes the log file unlimited.</p>

## 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

<b>Parameter</b>	host
<b>Type</b>	Text string
<b>Value</b>	<p>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.</p> <p>Specify the value in the following form:</p> <pre>set OPA0 host="&lt;host-name&gt;[:&lt;port-no&gt;]"</pre> <p>If the "&lt;port-no&gt;" is not specified, the virtual serial line uses the TCP/IP port number specified by the "port" parameter (see below).</p>


### port

<b>Parameter</b>	port
<b>Type</b>	Numeric
<b>Value</b>	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

<b>Parameter</b>	break_on
<b>Type</b>	Text string
<b>Value</b>	<p>Specifies what byte sequences received over a virtual serial line triggers a HALT command.</p> <p>Specify one of the following values: "Ctrl-P" , "Break" or "none" to disable triggering a HALT condition. The commands are case insensitive.</p> <div style="border: 1px solid #ccc; padding: 10px; margin: 10px 0;"> <p>If your guest operating system is OpenVMS in addition to "none" setting you have to set a specific console parameter "controlp" to "off" in the following way:</p> <pre>&gt;&gt;&gt; set controlp off &gt;&gt;&gt; power off</pre> <p>The second line is to preserve the ROM settings.</p> </div> <p><b>Example:</b></p> <pre>set OPA0 break_on="Ctrl-P"</pre> <p>The default value is "Break".</p> <p> This parameter can be specified only for COM1 (OPA0) console</p>


## stop\_on

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

## log

<b>Parameter</b>	log
<b>Type</b>	Text string
<b>Value</b>	<p>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.</p> <p>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.</p> <p><b>Examples:</b></p> <pre>set OPA0 log="log.txt"</pre> <pre>set OPA0 log="/Charon/Logs"</pre> <p style="text-align: center;">Only existing directory can be used as a value of the "log" parameter.</p>

## log\_file\_size

<b>Parameter</b>	log_file_size
<b>Type</b>	Text string
<b>Value</b>	<p>If log rotation is enabled, the log_file_size parameter determines the log file size threshold at which the log is automatically rotated.</p> <ul style="list-style-type: none"> <li>• "unlimited" or "0" (default) - the feature is disabled</li> <li>• "default" - default size is used (4Mb)</li> <li>• &lt;size&gt;[KMG] - size of the current log file in bytes with additional multipliers: <ul style="list-style-type: none"> <li>• K - Kilobyte - multiply by 1024</li> <li>• M - Megabyte - multiply by 1024*1024</li> <li>• G - Gigabyte - multiply by 1024*1024*1024</li> </ul> </li> </ul> <p><b>Examples:</b></p> <pre>set OPA0 log_file_size="default"</pre> <pre>set OPA0 log_file_size=10M</pre> <p> Minimum log file size is 64K, maximum is 1G. Setting size less than 64K effectively makes the log file unlimited.</p>

## 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

<b>Parameter</b>	break_on, stop_on
<b>Type</b>	Text string
<b>Value</b>	<p>These two parameters are hardcoded to the following values and cannot be changed:</p> <pre>stop_on="F6" break_on="Ctrl-P,F5"</pre>

### 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
<p><code>"/dev/tty&lt;N&gt;"</code> where N is from 0 to 11</p>	Linux virtual tty	<p>Those tty devices must be free from the Linux "getty/mgetty" and similar programs (specified in "/etc/inittab")</p> <p><b>Example:</b></p> <pre>"/dev/tty1"</pre>
<p><code>"/dev/ttyS&lt;N&gt;"</code> where N is a number</p>	Onboard serial lines	<p><b>Example:</b></p> <pre>"/dev/ttyS1"</pre>
<p><code>"/dev/tty&lt;XXX&gt;"</code> where XXX is a complex letter/number notation</p>	Proprietary (depending on a driver) devices	<p><b>Example for a first port of a MOXA card:</b></p> <pre>"/dev/ttyR01"</pre> <p><b>Example for a first port of a DIGI card:</b></p> <pre>"/dev/ttyaa"</pre>
<p><code>"/dev/ttyUSB&lt;N&gt;"</code> where N is a number</p>	Modem or USB serial lines adapters	<p><b>Example:</b></p> <pre>"/dev/ttyUSB1"</pre>

[Back to Table of Contents](#)

# Placement of peripheral devices on PCI bus

## Table of Contents

- General Description
  - bus
  - device
  - function
  - irq\_bus
  - irq
- Available PCI slots per each HP Alpha model emulated by CHARON-AXP
  - AlphaServer 400 (3 PCI slots)
  - AlphaServer 800 (4 PCI slots)
  - AlphaServer 1000 (3 PCI slots)
  - AlphaServer 1000A (7 PCI slots)
  - AlphaServer 1200 (6 PCI slots)
  - AlphaServer 2000 (3 PCI slots)
  - AlphaServer 2100 (3 PCI slots)
  - AlphaServer 4000 (16 PCI slots)
  - AlphaServer 4100 (8 PCI slots)
  - AlphaServer DS10 (4 PCI slots)
  - AlphaServer DS10L (1 PCI slot)
  - AlphaServer DS15 (4 PCI slots)
  - AlphaServer DS20 (6 PCI slots)
  - AlphaServer DS25 (6 PCI slots)
  - AlphaServer ES40 (10 PCI slots)
  - AlphaServer ES45 (10 PCI slots)
  - AlphaServer GS80 (8 PCI busses)
  - AlphaServer GS160 (16 PCI busses)
  - AlphaServer GS320 (32 PCI busses)

[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

<b>Parameter</b>	bus						
<b>Type</b>	Text string						
<b>Value</b>	<p>Value formats:</p> <table border="1"> <thead> <tr> <th>Models</th> <th>Format</th> </tr> </thead> <tbody> <tr> <td>AlphaServer 400-4100, DS, ES</td> <td>"pci_&lt;X&gt;"</td> </tr> <tr> <td>AlphaServer GS</td> <td>"qbb_&lt;X&gt;_pca_&lt;Y&gt;_pci_&lt;Z&gt;"</td> </tr> </tbody> </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><b>Example (AlphaServer ES40):</b></p> <pre>load KZPBA PKA bus=pci_1</pre> <p><b>Example (AlphaServer GS80):</b></p> <pre>load KZPBA PKA bus=qbb_1_pca_1_pci_0</pre>	Models	Format	AlphaServer 400-4100, DS, ES	"pci_<X>"	AlphaServer GS	"qbb_<X>_pca_<Y>_pci_<Z>"
Models	Format						
AlphaServer 400-4100, DS, ES	"pci_<X>"						
AlphaServer GS	"qbb_<X>_pca_<Y>_pci_<Z>"						

## device

<b>Parameter</b>	device
<b>Type</b>	Numeric
<b>Value</b>	<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><b>Example:</b></p> <pre>load KZPBA PKA device=2</pre>

## function

<b>Parameter</b>	function
<b>Type</b>	Numeric
<b>Value</b>	<p>When specified, the function configuration parameter specifies position of a virtual PCI adapter on virtual PCI bus.</p> <p>By default the function configuration parameter is not specified.</p> <p>If the function configuration parameter is not specified, the CHARON software connects the virtual PCI a dapter at the first available position of the virtual PCI bus.</p> <p><b>Example:</b></p> <pre>load KZPBA PKA function=0</pre>

## irq\_bus

<b>Parameter</b>	irq_bus
<b>Type</b>	Text string
<b>Value</b>	<p>When specified, the "irq_bus" configuration parameter specifies virtual bus routing interrupt requests from virtual PCI adapter to CHARON-AXP virtual Alpha CPUs.</p> <p>By default the "irq_bus" configuration parameter is not specified.</p> <p>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).</p> <p><b>Example:</b></p> <pre>load KZPBA PKA irq_bus=isa</pre>

## irq

<b>Parameter</b>	irq
<b>Type</b>	Numeric
<b>Value</b>	<p>When specified, the "irq" configuration parameter assigns interrupt request to the virtual PCI adapter in HP Alpha system.</p> <p>By default the irq configuration parameter is not specified.</p> <p>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.</p> <p><b>Example:</b></p> <pre>load KZPBA PKA irq=24</pre>

Note that 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 specifies 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
<i>PCI0 (bus=pci_0)</i>						
-	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
<i>PCI0 (bus=pci_0)</i>						
-	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
<i>PCI0 (bus=pci_0)</i>						
-	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
<i>PCI0 (bus=pci_0)</i>						
-	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	
<i>PCI1 (bus=pci_1)</i>						
-	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 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 EWA bus=pci_0 device=11 function=0
load DE500BA EWB bus=pci_0 device=11 function=1
load DE500BA EWC bus=pci_0 device=11 function=2
load DE500BA 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 EWA bus=pci_1 device=1 function=1
```

 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
<i>PCI1 (bus=pci_1)</i>						
-	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	
<i>PCI0 (bus=pci_0)</i>						
-	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
```

 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
<i>PCI0 (bus=pci_0)</i>						
-	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 EWB 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 EWB bus=pci_0 device=6 function=0
load DE500BA/dec21x4x EWC bus=pci_0 device=6 function=1
load DE500BA/dec21x4x EWD bus=pci_0 device=6 function=2
load DE500BA/dec21x4x EWE bus=pci_0 device=6 function=3
```

**i** In the above examples device name is EWB as there is a built-in EW-like PCI Ethernet Adapter located "closer" to CPU and therefore assigned name EWA.

**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 EWB 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
<i>PCI0 (bus=pci_0)</i>						
-	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 EWB 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 EWB bus=pci_0 device=6 function=0
load DE500BA/dec21x4x EWC bus=pci_0 device=6 function=1
load DE500BA/dec21x4x EWD bus=pci_0 device=6 function=2
load DE500BA/dec21x4x EWE bus=pci_0 device=6 function=3
```

**i** In the above examples device name is EWB as there is a built-in EW-like PCI Ethernet Adapter located “closer” to CPU and therefore assigned name EWA.

**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 EWB 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
<i>PCI1 (bus=pci_1)</i>						
-	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	
<i>PCI0 (bus=pci_0)</i>						
-	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	
<i>PCI3 (bus=pci_3)</i>						
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	
<i>PCI2 (bus=pci_2)</i>						
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
```

 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
<i>PCI1 (bus=pci_1)</i>						
-	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	
<i>PCI0 (bus=pci_0)</i>						
-	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
```

 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
<i>PCI1 (bus=pci_0)</i>						
-	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 EWC 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 EWC bus=pci_0 device=14 function=0
load DE500BA/dec21x4x EWD bus=pci_0 device=14 function=1
load DE500BA/dec21x4x EWE bus=pci_0 device=14 function=2
load DE500BA/dec21x4x EWF bus=pci_0 device=14 function=3
```

**i** In the above examples device name is EWC as there are built-in EW-like PCI Ethernet Adapters located “closer” to CPU and therefore assigned names EWA and EWB.

**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 PKB as there is a built-in PK-like PCI SCSI Adapter located “closer” to CPU and therefore assigned name PKA, 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
<i>PCI1 (bus=pci_0)</i>						
-	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 EWC 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 EWC bus=pci_0 device=17 function=0
load DE500BA/dec21x4x EWD bus=pci_0 device=17 function=1
load DE500BA/dec21x4x EWE bus=pci_0 device=17 function=2
load DE500BA/dec21x4x EWF bus=pci_0 device=17 function=3
```

**i** In the above examples device name is EWC as there are built-in EW-like PCI Ethernet Adapters located "closer" to CPU and therefore assigned names EWA and EWB.

**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 PKB as there is a built-in PK-like PCI SCSI Adapter located "closer" to CPU and therefore assigned name PKA, 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
<i>PCI0 (bus=pci_0)</i>						
-	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	
<i>PCI2 (bus=pci_2)</i>						
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 EWC 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 EWC bus=pci_2 device=8 function=0
load DE500BA/dec21x4x EWD bus=pci_2 device=8 function=1
load DE500BA/dec21x4x EWE bus=pci_2 device=8 function=2
load DE500BA/dec21x4x EWF bus=pci_2 device=8 function=3
```

**i** In the above examples device name is EWC as there are built-in EW-like PCI Ethernet Adapters located “closer” to CPU and therefore assigned names EWA and EWB.

**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 EWC bus=pci_2 device=9 function=1
```

**i** In the above example device name is PKC as there are 2 built-in PK-like PCI SCSI Adapter located “closer” to CPU and therefore assigned name PKA and PKB, 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 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
<i>PCI1 (bus=pci_1)</i>						
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	
<i>PCI0 (bus=pci_0)</i>						
-	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
```

 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
<i>PCI0 (bus=pci_0)</i>						
-	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	
<i>PCI1 (bus=pci_1)</i>						
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	
<i>PCI2 (bus=pci_2)</i>						
-	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)
<i>PCI3 (bus=pci_3)</i>						
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 EWC 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 EWC bus=pci_3 device=1 function=0
load DE500BA/dec21x4x EWD bus=pci_3 device=1 function=1
load DE500BA/dec21x4x EWE bus=pci_3 device=1 function=2
load DE500BA/dec21x4x EWF bus=pci_3 device=1 function=3
```

**i** In the above examples device name is EWC as there are built-in EW-like PCI Ethernet Adapters located "closer" to CPU and therefore assigned names EWA and EWB.

**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 EWC bus=pci_3 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, 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 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
<i>PCI1 (bus=pci_1)</i>						
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	
<i>PCI0 (bus=pci_0)</i>						
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
<i>PCI0 (bus=pci_0)</i>						
-	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	
<i>PCI1 (bus=pci_1)</i>						
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	
<i>PCI2 (bus=pci_2)</i>						
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	
<i>PCI3 (bus=pci_3)</i>						
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
<i>QBB0.PCA0.PCI0 (bus=qbb_0_pca_0_pci_0)</i>						
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	
<i>QBB0.PCA0.PCI1 (bus=qbb_0_pca_0_pci_1)</i>						
4	1	4	0	48	<option>	
5	1	5	0	52	<option>	
6	1	6	0	56	<option>	
7	1	7	0	60	<option>	
<i>QBB0.PCA1.PCI0 (bus=qbb_0_pca_1_pci_0)</i>						
0/1	2	0	0	32	<option>	
2	2	2	0	40	<option>	
3	2	3	0	44	<option>	
<i>QBB0.PCA1.PCI1 (bus=qbb_0_pca_1_pci_1)</i>						
4	3	4	0	48	<option>	
5	3	5	0	52	<option>	
6	3	6	0	56	<option>	
7	3	7	0	60	<option>	
<i>QBB1.PCA0.PCI0 (bus=qbb_1_pca_0_pci_0)</i>						
0/1	8	0	0	32	<option>	
2	8	2	0	40	<option>	
3	8	3	0	44	<option>	
<i>QBB1.PCA0.PCI1 (bus=qbb_1_pca_0_pci_1)</i>						
4	9	4	0	48	<option>	
5	9	5	0	52	<option>	
6	9	6	0	56	<option>	
7	9	7	0	60	<option>	
<i>QBB1.PCA1.PCI0 (bus=qbb_1_pca_1_pci_0)</i>						
0/1	10	0	0	32	<option>	
2	10	2	0	40	<option>	
3	10	3	0	44	<option>	
<i>QBB1.PCA1.PCI1 (bus=qbb_1_pca_1_pci_1)</i>						
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
<i>QBB0.PCA0.PCI0 (bus=qbb_0_pca_0_pci_0)</i>						
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	
<i>QBB0.PCA0.PCI1 (bus=qbb_0_pca_0_pci_1)</i>						
4	1	4	0	48	<option>	
5	1	5	0	52	<option>	
6	1	6	0	56	<option>	
7	1	7	0	60	<option>	
<i>QBB0.PCA1.PCI0 (bus=qbb_0_pca_1_pci_0)</i>						
0/1	2	0	0	32	<option>	
2	2	2	0	40	<option>	
3	2	3	0	44	<option>	
<i>QBB0.PCA1.PCI1 (bus=qbb_0_pca_1_pci_1)</i>						
4	3	4	0	48	<option>	
5	3	5	0	52	<option>	
6	3	6	0	56	<option>	
7	3	7	0	60	<option>	
<i>QBB1.PCA0.PCI0 (bus=qbb_1_pca_0_pci_0)</i>						
0/1	8	0	0	32	<option>	
2	8	2	0	40	<option>	
3	8	3	0	44	<option>	
<i>QBB1.PCA0.PCI1 (bus=qbb_1_pca_0_pci_1)</i>						
4	9	4	0	48	<option>	
5	9	5	0	52	<option>	
6	9	6	0	56	<option>	
7	9	7	0	60	<option>	
<i>QBB1.PCA1.PCI0 (bus=qbb_1_pca_1_pci_0)</i>						
0/1	10	0	0	32	<option>	
2	10	2	0	40	<option>	
3	10	3	0	44	<option>	
<i>QBB1.PCA1.PCI1 (bus=qbb_1_pca_1_pci_1)</i>						
4	11	4	0	48	<option>	
5	11	5	0	52	<option>	

6	11	6	0	56	<option>	
7	11	7	0	60	<option>	
<i>QBB2.PCA0.PCI0 (bus=qbb_2_pca_0_pci_0)</i>						
0/1	16	0	0	32	<option>	
2	16	2	0	40	<option>	
3	16	3	0	44	<option>	
<i>QBB2.PCA0.PCI1 (bus=qbb_2_pca_0_pci_1)</i>						
4	17	4	0	48	<option>	
5	17	5	0	52	<option>	
6	17	6	0	56	<option>	
7	17	7	0	60	<option>	
<i>QBB2.PCA1.PCI0 (bus=qbb_2_pca_1_pci_0)</i>						
0/1	18	0	0	32	<option>	
2	18	2	0	40	<option>	
3	18	3	0	44	<option>	
<i>QBB2.PCA1.PCI1 (bus=qbb_2_pca_1_pci_1)</i>						
4	19	4	0	48	<option>	
5	19	5	0	52	<option>	
6	19	6	0	56	<option>	
7	19	7	0	60	<option>	
<i>QBB3.PCA0.PCI0 (bus=qbb_3_pca_0_pci_0)</i>						
0/1	24	0	0	32	<option>	
2	24	2	0	40	<option>	
3	24	3	0	44	<option>	
<i>QBB3.PCA0.PCI1 (bus=qbb_3_pca_0_pci_1)</i>						
4	25	4	0	48	<option>	
5	25	5	0	52	<option>	
6	25	6	0	56	<option>	
7	25	7	0	60	<option>	
<i>QBB3.PCA1.PCI0 (bus=qbb_3_pca_1_pci_0)</i>						
0/1	26	0	0	32	<option>	
2	26	2	0	40	<option>	
3	26	3	0	44	<option>	
<i>QBB0.PCA0.PCI1 (bus=qbb_0_pca_0_pci_1)</i>						
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
<i>QBB0.PCA0.PCI0 (bus=qbb_0_pca_0_pci_0)</i>						
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	
<i>QBB0.PCA0.PCI1 (bus=qbb_0_pca_0_pci_1)</i>						
4	1	4	0	48	<option>	
5	1	5	0	52	<option>	
6	1	6	0	56	<option>	
7	1	7	0	60	<option>	
<i>QBB0.PCA1.PCI0 (bus=qbb_0_pca_1_pci_0)</i>						
0/1	2	0	0	32	<option>	
2	2	2	0	40	<option>	
3	2	3	0	44	<option>	
<i>QBB0.PCA1.PCI1 (bus=qbb_0_pca_1_pci_1)</i>						
4	3	4	0	48	<option>	
5	3	5	0	52	<option>	
6	3	6	0	56	<option>	
7	3	7	0	60	<option>	
<i>QBB1.PCA0.PCI0 (bus=qbb_1_pca_0_pci_0)</i>						
0/1	8	0	0	32	<option>	
2	8	2	0	40	<option>	
3	8	3	0	44	<option>	
<i>QBB1.PCA0.PCI1 (bus=qbb_1_pca_0_pci_1)</i>						
4	9	4	0	48	<option>	
5	9	5	0	52	<option>	
6	9	6	0	56	<option>	
7	9	7	0	60	<option>	
<i>QBB1.PCA1.PCI0 (bus=qbb_1_pca_1_pci_0)</i>						
0/1	10	0	0	32	<option>	
2	10	2	0	40	<option>	
3	10	3	0	44	<option>	
<i>QBB1.PCA1.PCI1 (bus=qbb_1_pca_1_pci_1)</i>						
4	11	4	0	48	<option>	
5	11	5	0	52	<option>	

6	11	6	0	56	<option>	
7	11	7	0	60	<option>	
<i>QBB2.PCA0.PCI0 (bus=qbb_2_pca_0_pci_0)</i>						
0/1	16	0	0	32	<option>	
2	16	2	0	40	<option>	
3	16	3	0	44	<option>	
<i>QBB2.PCA0.PCI1 (bus=qbb_2_pca_0_pci_1)</i>						
4	17	4	0	48	<option>	
5	17	5	0	52	<option>	
6	17	6	0	56	<option>	
7	17	7	0	60	<option>	
<i>QBB2.PCA1.PCI0 (bus=qbb_2_pca_1_pci_0)</i>						
0/1	18	0	0	32	<option>	
2	18	2	0	40	<option>	
3	18	3	0	44	<option>	
<i>QBB2.PCA1.PCI1 (bus=qbb_2_pca_1_pci_1)</i>						
4	19	4	0	48	<option>	
5	19	5	0	52	<option>	
6	19	6	0	56	<option>	
7	19	7	0	60	<option>	
<i>QBB3.PCA0.PCI0 (bus=qbb_3_pca_0_pci_0)</i>						
0/1	24	0	0	32	<option>	
2	24	2	0	40	<option>	
3	24	3	0	44	<option>	
<i>QBB3.PCA0.PCI1 (bus=qbb_3_pca_0_pci_1)</i>						
4	25	4	0	48	<option>	
5	25	5	0	52	<option>	
6	25	6	0	56	<option>	
7	25	7	0	60	<option>	
<i>QBB3.PCA1.PCI0 (bus=qbb_3_pca_1_pci_0)</i>						
0/1	26	0	0	32	<option>	
2	26	2	0	40	<option>	
3	26	3	0	44	<option>	
<i>QBB3.PCA1.PCI1 (bus=qbb_3_pca_1_pci_1)</i>						
4	27	4	0	48	<option>	
5	27	5	0	52	<option>	
6	27	6	0	56	<option>	
7	27	7	0	60	<option>	
<i>QBB4.PCA0.PCI0 (bus=qbb_4_pca_0_pci_0)</i>						
0/1	32	0	0	32	<option>	
2	32	2	0	40	<option>	

3	32	3	0	44	<option>	
<i>QBB4.PCA0.PCI1 (bus=qbb_4_pca_0_pci_1)</i>						
4	33	4	0	48	<option>	
5	33	5	0	52	<option>	
6	33	6	0	56	<option>	
7	33	7	0	60	<option>	
<i>QBB4.PCA1.PCI0 (bus=qbb_4_pca_1_pci_0)</i>						
0/1	34	0	0	32	<option>	
2	34	2	0	40	<option>	
3	34	3	0	44	<option>	
<i>QBB4.PCA1.PCI1 (bus=qbb_4_pca_1_pci_1)</i>						
4	35	4	0	48	<option>	
5	35	5	0	52	<option>	
6	35	6	0	56	<option>	
7	35	7	0	60	<option>	
<i>QBB5.PCA0.PCI0 (bus=qbb_5_pca_0_pci_0)</i>						
0/1	40	0	0	32	<option>	
2	40	2	0	40	<option>	
3	40	3	0	44	<option>	
<i>QBB5.PCA0.PCI1 (bus=qbb_5_pca_0_pci_1)</i>						
4	41	4	0	48	<option>	
5	41	5	0	52	<option>	
6	41	6	0	56	<option>	
7	41	7	0	60	<option>	
<i>QBB5.PCA1.PCI0 (bus=qbb_5_pca_1_pci_0)</i>						
0/1	42	0	0	32	<option>	
2	42	2	0	40	<option>	
3	42	3	0	44	<option>	
<i>QBB5.PCA1.PCI1 (bus=qbb_5_pca_1_pci_1)</i>						
4	43	4	0	48	<option>	
5	43	5	0	52	<option>	
6	43	6	0	56	<option>	
7	43	7	0	60	<option>	
<i>QBB6.PCA0.PCI0 (bus=qbb_6_pca_0_pci_0)</i>						
0/1	48	0	0	32	<option>	
2	48	2	0	40	<option>	
3	48	3	0	44	<option>	
<i>QBB6.PCA0.PCI1 (bus=qbb_6_pca_0_pci_1)</i>						
4	49	4	0	48	<option>	
5	49	5	0	52	<option>	
6	49	6	0	56	<option>	

7	49	7	0	60	<option>	
<i>QBB6.PCA1.PCI0 (bus=qbb_6_pca_1_pci_0)</i>						
0/1	50	0	0	32	<option>	
2	50	2	0	40	<option>	
3	50	3	0	44	<option>	
<i>QBB6.PCA1.PCI1 (bus=qbb_6_pca_1_pci_1)</i>						
4	51	4	0	48	<option>	
5	51	5	0	52	<option>	
6	51	6	0	56	<option>	
7	51	7	0	60	<option>	
<i>QBB7.PCA0.PCI0 (bus=qbb_7_pca_0_pci_0)</i>						
0/1	56	0	0	32	<option>	
2	56	2	0	40	<option>	
3	56	3	0	44	<option>	
<i>QBB7.PCA0.PCI1 (bus=qbb_7_pca_0_pci_1)</i>						
4	57	4	0	48	<option>	
5	57	5	0	52	<option>	
6	57	6	0	56	<option>	
7	57	7	0	60	<option>	
<i>QBB7.PCA1.PCI0 (bus=qbb_7_pca_1_pci_0)</i>						
0/1	58	0	0	32	<option>	
2	58	2	0	40	<option>	
3	58	3	0	44	<option>	
<i>QBB7.PCA1.PCI1 (bus=qbb_7_pca_1_pci_1)</i>						
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
- KGPSA-CA PCI Fibre Channel adapter
- Acer Labs 1543C IDE/ATAPI CD-ROM adapter
- PCI I/O Bypass controller
- Finding the target "/dev/sg" device

## KZPBA PCI SCSI adapter

### Table of Contents

- General description
- Loading KZPBA storage adapter
- Configuration parameters
  - scsi\_id
  - host, port
  - container
  - media\_type
  - removable
  - geometry
  - use\_io\_file\_buffering

[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
```

In AlphaStation 400 configuration use the following syntax for KZPBA storage adapter loading:

```
load KZPBA PKB irq_bus=isa
```

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

<b>Parameter</b>	scsi_id
<b>Type</b>	Numeric
<b>Value</b>	<p>Specifies SCSI ID of KZPBA PCI SCSI Adapter in a range 0..7</p> <p>By default the "scsi_id" configuration parameter is set to 7.</p> <p><b>Example:</b></p> <pre>set PKA scsi_id=0</pre>

### host, port

<b>Parameter</b>	host, port						
<b>Type</b>	Text String						
<b>Value</b>	<p>These parameters are used in SCSI cluster configurations.</p> <table border="1"> <thead> <tr> <th>Parameter</th> <th>Description</th> </tr> </thead> <tbody> <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> </tbody> </table> <p>By default the "host" and "port" configuration options are not specified.</p> <p>Syntax:</p> <pre>port[connection-number]=&lt;local port&gt; host[connection-number]="&lt;host-name{:tcpip-port-no}&gt;"</pre> <p>where: <math>connection\_number = remote\_scsi\_id * 100 + lun\_id</math></p> <p><b>Example:</b></p> <pre>set PKA port[600]=17060 host[600]="localhost:16070"</pre>	Parameter	Description	host	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.	port	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.
Parameter	Description						
host	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.						
port	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.						

## container

<b>Parameter</b>	container[N]  N is "XXYY" number, where: <ul style="list-style-type: none"> <li>▪ XX - SCSI ID (0..15)</li> <li>▪ YY - LUN (00..07)</li> </ul>
<b>Type</b>	Text String
<b>Value</b>	<p>Possible values of the parameter are strings in one of the following forms:</p> <ul style="list-style-type: none"> <li>▪ <b>Physical disk</b> <ul style="list-style-type: none"> <li>▪ <code>"/dev/sd&lt;L&gt;"</code> where "L" is letter, for example <code>"/dev/sdb"</code></li> <li>▪ <code>"/dev/disk/by-id/..."</code> - addressing by the disk ID, for example <code>"/dev/disk/by-id/ata-ST1000DM003-9YN162_S1D01QJ4"</code></li> <li>▪ <code>"/dev/disk/by-label/..."</code> - addressing by the disk label, for example <code>"/dev/disk/by-label/MyStorage"</code></li> <li>▪ <code>"/dev/disk/by-uuid/..."</code> - addressing by the disk UUID, for example <code>"/dev/disk/by-uuid/0e808a2f-cdd3-4944-a245-f729ffd73882"</code></li> </ul> <p> Please note that existing data on such a disk may be destroyed, depending on how it is used in the emulator.</p> <div style="border: 1px solid #ffc107; padding: 5px; margin: 5px 0;"> <p><code>"/dev/sd&lt;L&gt;"</code> addressing is not persistent, so it is strongly recommended to use <code>"/dev/disk/by-id/wwn-..."</code> syntax instead to refer the disk by its WWID - especially in the environments utilizing FC and SAN storages.</p> </div> <p>These disks must not be formatted by the host OS.</p> <p><b>Example:</b></p> <div style="border: 1px solid #ccc; padding: 5px; margin: 5px 0;"> <pre>set PKA container[0]="/dev/disk/by-id/ata-ST1000DM003-9YN162_S1D01QJ4"</pre> </div> <p>It is also possible to use not a whole disk, but previously created partitions on it. In this case the syntax is the following: <code>"/dev/sd&lt;L&gt;&lt;N&gt;"</code> where N is the number of partition to be used.</p> <p><b>Example:</b></p> <div style="border: 1px solid #ccc; padding: 5px; margin: 5px 0;"> <pre>set PKA container[0]="/dev/sdc1"</pre> </div> </li> <li>▪ <b>Multipath disk</b> <ul style="list-style-type: none"> <li>▪ <code>"/dev/dm-&lt;N&gt;"</code></li> <li>▪ <code>"/dev/mapper/mpath&lt;N&gt;"</code></li> <li>▪ <code>"/dev/mapper/disk&lt;N&gt;"</code></li> </ul> <p> Be careful not to destroy all the information from the disk dedicated to CHARON-AXP by mistake.</p> <p>These disks must not be formatted by the host OS.</p> <p><b>Example:</b></p> <div style="border: 1px solid #ccc; padding: 5px; margin: 5px 0;"> <pre>set PKA container[100]="/dev/dm-0"</pre> </div> </li> <li>▪ <b>Loop (virtual block) devices</b> <ul style="list-style-type: none"> <li>▪ <code>"/dev/loop&lt;N&gt;"</code></li> </ul> <p><b>Example:</b></p> <div style="border: 1px solid #ccc; padding: 5px; margin: 5px 0;"> <pre>set PKA container[200]="/dev/loop0"</pre> </div> </li> <li>▪ <b>Direct mapping to some SCSI device, for example, a SCSI disk or tape reader</b> <ul style="list-style-type: none"> <li>▪ <code>"/dev/sg&lt;N&gt;"</code></li> </ul> <p><b>Example:</b></p> <div style="border: 1px solid #ccc; padding: 5px; margin: 5px 0;"> <pre>set PKA container[300]="/dev/sg0"</pre> </div> </li> </ul>

- **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 "mkdiskcmd" 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"
```

### How the Emulator Maps Guest-OS Operations to the Virtual Tape Drive

Guest-OS operations	Emulator Action
Open device for reading	Create a container file if none exists. open for reading and lock container file
Open device for writing	Create a container file if necessary; open for writing and lock the container file
Unload (eject) tape from drive	Close a container file if open and unlock it - this allows copy/move/delete operations on CHARON host

The container file associated with a virtual tape drive can be compared to the tape cartridge used in a physical tape drive. Both store the data written to the tape device by the guest OS.

The size of virtual tape container files is limited only by space available in the emulator host file system.

 Prerequisite to the examples below: a virtual tape device has been configured in the CHARON configuration file and it is not in use by the guest OS.

**To perform backup:**

1. The tape device may be issued the "unload" command and the container-file moved/deleted to insure proper status
2. Initialize the tape device using standard guest OS procedure.
3. Perform backup.
4. Issue "unload" command to the tape device in the guest OS.
5. On the emulator host, move the \*.vtape container file containing backup data for storage or further backup.

**To restore from a backup:**

1. The tape device may be issued the "unload" command to insure proper status.
2. On the emulator host, move or copy a \*.vtape container file containing backup data onto the filename specified in the CHARON configuration file.
3. Perform restore.
4. Issue the "unload" command to the tape device in the guest OS.
5. Delete or move the container file in preparation for the next vtape operation.

CHARON does not support multi-volume backup for tape images. If some multi-volume set (in form of tape images) has to be restored it is recommended to configure several tape drives in CHARON configuration file, assign each tape image to each tape drive and use them in the following way (OpenVMS example):


```
$ BACKUP MKA100:BACKUP.BCK,MKA200,MKA300,MKA4000/SAVE_SET DKA0:...
```

This parameter is initially not set, thus creating NO storage elements on the controller.

## media\_type

<b>Parameter</b>	<p>media_type[N]</p> <p>N is "XXYY" number, where:</p> <ul style="list-style-type: none"> <li>▪ XX - SCSI ID (0..15)</li> <li>▪ YY - LUN (00..07)</li> </ul>
<b>Type</b>	Text String
<b>Value</b>	<p>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.</p> <p>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).</p> <p>Initially is not specified.</p> <p><b>Example:</b></p> <pre>set PKA media_type[0]="HSZ70"</pre>

## removable


<b>Parameter</b>	<p>removable[N]</p> <p>N is "XXYY" number, where:</p> <ul style="list-style-type: none"> <li>▪ XX - SCSI ID (0..15)</li> <li>▪ YY - LUN (00..07)</li> </ul>
<b>Type</b>	Boolean
<b>Value</b>	<p>When set to "true", the removable configuration parameter instructs CHARON-AXP to report the corresponding virtual SCSI device as removable.</p> <p>By default the removable configuration parameter is set to "false".</p> <p><b>Example:</b></p> <pre>set PKA removable[400]=true</pre> <p> Note that virtual SCSI tapes and CD-ROM devices are always reported as removable regardless of the "removable" configuration parameter.</p>

## geometry

<b>Parameter</b>	geometry[N]  N is "XXYY" number, where: <ul style="list-style-type: none"> <li>▪ XX - SCSI ID (0..15)</li> <li>▪ YY - LUN (00..07)</li> </ul>																
<b>Type</b>	Text String																
<b>Value</b>	<p>This formatted string value specifies the explicit geometry of the disk storage element. This parameter is not applicable to tape storage elements.</p> <p>The string format is &lt;X&gt;"/"&lt;Y&gt;["&lt;Z&gt;] or &lt;X&gt;"/"&lt;Y&gt;["&lt;Z&gt;"]["&lt;B&gt;] where:</p> <table border="1" style="width: 100%; border-collapse: collapse; margin-bottom: 10px;"> <thead> <tr> <th style="width: 15%;">Parameter</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>X</td> <td>The number of sectors per track</td> </tr> <tr> <td>Y</td> <td>The number of tracks per cylinder</td> </tr> <tr> <td>Z</td> <td>           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.         </td> </tr> <tr> <td>B</td> <td>           The total size of the disk (in blocks) reported to the guest OS.             If omitted it is calculated automatically.             This is an optional parameter.         </td> </tr> </tbody> </table> <p>If this parameter is not set, CHARON-AXP will configure the geometry based on the most probable disk type.</p> <p>Initially not set.</p> <div style="border: 1px solid #ccc; padding: 10px; margin: 10px 0;"> <p style="text-align: center;">It is possible to specify each parameter independently of another one. The following syntax is used for that:</p> <pre style="border: 1px solid #ccc; padding: 5px; display: inline-block;">set PKA geometry[300]="*,*,*,16777210"</pre> </div> <p>The syntax described above is applicable only to disk storage elements. If the container is a tape image, the following format is used instead:</p> <p>Syntax:</p> <pre style="border: 1px solid #ccc; padding: 5px; display: inline-block;">"&lt;image-size&gt;[, &lt;early-warning-zone-size&gt;]"</pre> <p>where:</p> <table border="1" style="width: 100%; border-collapse: collapse; margin-bottom: 10px;"> <thead> <tr> <th style="width: 20%;">Parameter</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>image-size</td> <td>The tape size in MB</td> </tr> <tr> <td>early-warning-zone-size</td> <td>           The size (in KB) of the space left on the tape when a warning to the OS is issued.             If omitted, 64K is assumed.         </td> </tr> </tbody> </table> <p><b>Example:</b></p> <pre style="border: 1px solid #ccc; padding: 5px; display: inline-block;">set PKA geometry[603] = "255/255"</pre>	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.	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.
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.																
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.																



## use\_io\_file\_buffering

<b>Parameter</b>	<p>use_io_file_buffering[N]</p> <p>N is "XXYY" number, where:</p> <ul style="list-style-type: none"> <li>▪ XX - SCSI ID (0..15)</li> <li>▪ YY - LUN (00..07)</li> </ul>
<b>Type</b>	Boolean
<b>Value</b>	<p>When set to "true", instructs CHARON-AXP to enable host operating system I/O cache on reading/writing operations.</p> <p> Note that this caching has a significant effect only in case of mapping to disk and tape containers, not physical drives.</p> <p>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.</p> <p>Initially is set to "false".</p> <p><b>Example:</b></p> <pre>set PKA use_io_file_buffering[603]=true</pre>

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 "mkdiskcmd" 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

- General description
- Loading KGPSA storage adapter
- Configuration parameters
  - host\_bus\_location
  - wwid
  - container
  - media\_type
  - removable
  - geometry
  - use\_io\_file\_buffering
- Mapping to host resources
  - Fabric virtualization mode
  - Pass Through mode
    - Pass Through mode establishing sequence
    - Building PPT driver for EMULEX LightPulse PCI/PCI-X/PCIe FC
    - Installation of PPT driver for EMULEX LightPulse PCI/PCI-X/PCIe FC
    - Adding PPT driver for EMULEX LightPulse PCI/PCI-X/PCIe FC to Linux startup

[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 two 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
```

In AlphaStation 400 configuration use the following syntax for KGPSA-CA storage adapter loading:

```
load KGPSA FGA irq_bus=isa
```

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

<b>Parameter</b>	host_bus_location
<b>Type</b>	Text String
<b>Value</b>	<p> Pass through mode only!</p> <p>Establish connection between virtual DEC-KGPSA-CA PCI FC adapter and physical EMULEX LightPulse PCI/PCI-X/PCIe FC adapter (pass through mode)</p> <p>Syntax:</p> <pre>load KGPSA &lt;controller name&gt; host_bus_location="/dev/kgpsa&lt;X&gt;"</pre> <p><b>Example:</b></p> <pre>load KGPSA FGA host_bus_location="/dev/kgpsa0"</pre>

### wwid

<b>Parameter</b>	wwid[N] N is 0..32766 (no more than 255 units)
<b>Type</b>	Text String
<b>Value</b>	<p>Sets WWID for emulated KGPSA adapter unit.</p> <p>Syntax:</p> <pre>set &lt;controller name&gt; wwid[unit-number]="XXXX-XXXX-XXXX-XXXX-XXXX-XXXX-XXXX-XXXX"</pre> <p><b>Example:</b></p> <pre>set FGA wwid[2]="6008-05F3-0005-2950-BF8E-0B86-A0C7-0001"</pre>

**container**

<b>Parameter</b>	container[N] N is 0..32766 (no more than 255 units)
<b>Type</b>	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, for example `"/dev/sdb"`
- `"/dev/disk/by-id/..."` - addressing by the disk ID, for example `"/dev/disk/by-id/ata-ST1000DM003-9YN162_S1D01QJ4"`
- `"/dev/disk/by-label/..."` - addressing by the disk label, for example `"/dev/disk/by-label/MyStorage"`
- `"/dev/disk/by-uuid/..."` - addressing by the disk UUID, for example `"/dev/disk/by-uuid/0e808a2f-cdd3-4944-a245-f729ffd73882"`

⚠ Please note that existing data on such a disk may be destroyed, depending on how it is used in the emulator.

`"/dev/sd<L>"` addressing is not persistent, so it is strongly recommended to use `"/dev/disk/by-id/wwn-..."` syntax instead to refer the disk by its WWID - especially in the environments utilizing FC and SAN storages.

These disks must not be formatted by the host OS.

**Example:**

```
set FGA container[0]="/dev/disk/by-id/ata-ST1000DM003-9YN162_S1D01QJ4"
```

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"
```

- **Loop (virtual block) devices**

- `"/dev/loop<N>"`

**Example:**

```
set FGA
container[100]="/dev/loop0"
```

- **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 "mkdiskcmd" 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.

## media\_type

<b>Parameter</b>	media_type[N] N is 0..32766 (no more than 255 units)
<b>Type</b>	Text String
<b>Value</b>	<p>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.</p> <p>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).</p> <p>Initially is not specified.</p> <p><b>Example:</b></p> <pre>set FGA media_type[0]="HSZ70"</pre>


## removable

<b>Parameter</b>	removable[N] N is 0..32766 (no more than 255 units)
<b>Type</b>	Boolean
<b>Value</b>	<p>When set to "true", the removable configuration parameter instructs CHARON-AXP to report the corresponding virtual FC device as removable.</p> <p>By default the removable configuration parameter is set to "false".</p> <p><b>Example:</b></p> <pre>set FGA removable[400]=true</pre>

## geometry

<b>Parameter</b>	geometry[N] N is 0..32766 (no more than 255 units)										
<b>Type</b>	Text String										
<b>Value</b>	<p>This formatted string value specifies the explicit geometry of the disk storage element.</p> <p>The string format is &lt;X&gt;"/"&lt;Y&gt;["/"&lt;Z&gt;] or &lt;X&gt;"/"&lt;Y&gt;["/"&lt;Z&gt;]["/"&lt;B&gt;] where:</p> <table border="1"> <thead> <tr> <th>Parameter</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>X</td> <td>The number of sectors per track</td> </tr> <tr> <td>Y</td> <td>The number of tracks per cylinder</td> </tr> <tr> <td>Z</td> <td>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.</td> </tr> <tr> <td>B</td> <td>The total size of the disk (in blocks) reported to the guest OS.  If omitted it is calculated automatically.  This is an optional parameter.</td> </tr> </tbody> </table> <p>If this parameter is not set, CHARON-AXP will configure the geometry based on the most probable disk type.</p> <p>Initially not set.</p> <p><b>Example:</b></p> <pre>set FGA geometry[201] = "255/255"</pre> <p>It is possible to specify each parameter independently of another one. The following syntax is used for that:</p> <pre>set FGA geometry[300]="*,*,*,16777210"</pre>	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.
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.										

## use\_io\_file\_buffering

<b>Parameter</b>	use_io_file_buffering[N] N is 0..32766 (no more than 255 units)
<b>Type</b>	Boolean
<b>Value</b>	<p>When set to "true", instructs CHARON-AXP to enable host operating system I/O cache on reading/writing operations.</p> <p> Note that this caching has a significant effect only in case of mapping to disk containers, not physical drives.</p> <p>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.</p> <p>Initially is set to "false".</p> <p><b>Example:</b></p> <pre>set FGA use_io_file_buffering[300]=true</pre>

When a disk image connected to an emulated KGPSA-CA controller is dismantled 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 "mkdiskcmd" 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/PCle FC adapter (see below for a list of supported models) to some spare PCI/PCI-X/PCle slot of the host system
2. Build PPT driver for EMULEX LightPulse PCI/PCI-X/PCle FC
3. Install PPT driver for EMULEX LightPulse PCI/PCI-X/PCle FC
4. Add PPT driver for EMULEX LightPulse PCI/PCI-X/PCle FC to Linux startup
5. Map KGPSA-CA adapter(-s) to EMULEX LightPulse PCI/PCI-X/PCle 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/PCle FC

To build PPT driver for EMULEX LightPulse PCI/PCI-X/PCle FC do the following:

Step	Description
1	<p>Make sure that the required building tools and include files are installed. If they are absent install them:</p> <pre># yum groupinstall "Development Tools" # yum install kernel-headers kernel-devel</pre> <p>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 "<code>rpm -q -a   grep kernel-</code>")</p> <p>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 "<code>uname -a</code>" command.</p>
2	<p>Open xterm and change the default directory to "<code>/opt/charon/drivers/kgpsa</code>":</p> <pre># cd /opt/charon/drivers/kgpsa</pre>
3	<p>Issue "make clean; make" commands to build kernel object:</p> <pre># make clean; make</pre> <p> It is prohibited to use a module built on a certain version of kernel on another one.</p>
4	<p>Check that there are no compilation errors and the file "<code>ppt_kgpsa.ko</code>" has been built</p>

[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: <pre># rmmod lpfc</pre>
2	Load "ppt_kgpsa.ko" driver; to do that issue the following command: <pre># insmod ppt_kgpsa.ko</pre>
3	Issue "dmesg" command and check that no error appeared during the driver loading, also check that the driver has found all KGPSA devices: <pre># dmesg</pre>

[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 "blacklist 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: <pre># cp /opt/charon/drivers/kgpsa/ppt_kgpsa.ko /lib/modules/3.10.9-200.fc20.x86_64/kernel/drivers/scsi/</pre> <p> The particular path may be different, depending on the kernel version and Linux distribution.</p>
3	Enable auto load of the module: <pre># echo ppt_kgpsa &gt; /etc/modules-load.d/ppt_kgpsa.conf</pre>
4	Regenerate new "initramfs" image with "mkinitrd": <pre># mkinitrd -f /boot/initramfs-3.10.9-200.fc20.x86_64.img 3.10.9-200.fc20.x86_64</pre> <p> The particular path may be different, depending on the kernel version and Linux distribution.</p>

[Back to Table of Contents](#)

## Acer Labs 1543C IDE/ATAPI CD-ROM adapter

### Table of Contents

- General description
- Loading Acer Labs 1543C IDE/ATAPI adapter
- Configuration parameters
  - container
  - Example

[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

<b>Parameter</b>	container
<b>Type</b>	Text String
<b>Value</b>	<ul style="list-style-type: none"> <li>• "/dev/sg&lt;N&gt;"</li> </ul> <p>Specifies a physical device correspondent to ATAPI or SATA CD/DVD-ROM drive attached to the host system. By default it is left unspecified.</p> <p> <a href="#">Read this article</a> on how to find the physical device name for mapping.</p>

## 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 be 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](#)

## PCI I/O Bypass controller

### Table of Contents

- General description
- Prerequisites
- Installation
- Loading PCI I/O bypass controller
- Configuration parameters
  - container
  - removable
  - geometry
  - use\_io\_file\_buffering
- Deinstallation

[Back to Table of Contents](#)

### General description

CHARON-AXP supports PCI I/O bypass controller for accessing to disk images and host physical disks. PCI I/O bypass controller requires a specific driver to be installed.

PCI I/O bypass controller support is available only for OpenVMS guest operating system.

[Back to Table of Contents](#)

### Prerequisites

This release supports VMS version V6.2-1H3 and higher. Bypass disks can not be used as a boot device in V6.2-1H3, higher versions do not have this restriction.

Make sure that the latest Bypass controller kit has been installed, especially for VMS versions before V7.3-2.

[Back to Table of Contents](#)

## Installation

1. Open CHARON-AXP configuration file and attach the virtual disk "ovms\_tool.vdisk" located by default in the "/opt/charon/disks" directory:

```
set PKA container[400] = "/opt/charon/disks/ovms_tools.vdisk"
```

2. Run CHARON, boot guest OpenVMS operating system.
3. Use the POLYCENTER Software Installation (PCSI) utility to install the Bypass Driver. The following example demonstrates the "PCSI PRODUCT INSTALL" command to execute and the expected output (the example assumes the utilities virtual disk image is attached as DKA400):

```
$ PRODUCT INSTALL CHARON_DISK/SOURCE=DKA400:[BYPASS]
%PCSI-I-CANNOTVAL, cannot validate DKA400:[BYPASS]SRI-AXPVMS-CHARON_DISK-V0104--1.PCSI;1
-PCSI-I-NOTSIGNED, product kit is not signed and therefore has no manifest file

The following product has been selected:
  SRI AXPVMS CHARON_DISK V1.4 Layered Product

Do you want to continue? [YES] YES

Configuration phase starting ...

You will be asked to choose options, if any, for each selected product and for any products that may be
installed to satisfy software dependency requirements.

SRI AXPVMS CHARON_DISK V1.4: Charon disk driver V1.4 for OpenVMS Alpha.

  Copyright (C) 1976, 2009 Software Resources International

  CHARON_DISK was produced by Software Resources International

* This product does not have any configuration options.

Execution phase starting ...

The following product will be installed to destination:
  SRI AXPVMS CHARON_DISK V1.4 DISK$TARDISSYSTEM:[VMS$COMMON.]

Portion done: 0%...10%...20%...80%...100%

The following product has been installed:
  SRI AXPVMS CHARON_DISK V1.4 Layered Product
```

[Back to Table of Contents](#)

## Loading PCI I/O bypass controller

Syntax for loading PCI I/O bypass storage adapter:

```
load pci_io_bypass <name>
```

The <name> can be DI<x>, DR<x> or DU<x>

where x is selected according to VMS naming scheme, i.e. A stands for the first controller of given type, B - for the second, etc.

**Example:**

```
load pci_io_bypass DIA
```

In AlphaStation 400 configuration use the following syntax for PCI I/O bypass storage adapter loading:

```
load pci_io_bypass DIA irq_bus=isa
```

The adapter instance name ("DIA" in the example above) is used then for parametrization, for example:

```
set DIA container[0]="/Mydisks/vms_distributive.vdisk"
```

The numbers in the square brackets represent a number of device on PCI I/O Bypass controller.

The maximum number of I/O Bypass controller devices is 64.

By default I/O Bypass controller uses PCI slot corresponded to the <x> parameter (see above). If instead some particular slot is needed, refer to [this section](#) for details of specific placement of PCI peripherals on CHARON Virtual Machine (VM) PCI bus (note that "irq\_bus" and "irq" parameters are ignored for I/O Bypass controller). In this case the <x> will be changed automatically according to custom position of I/O Bypass controller on PCI bus.



 I/O Bypass controller is implemented for OpenVMS only.

[Back to Table of Contents](#)

## Configuration parameters

The I/O Bypass controller has the following configuration parameters:

## container

<b>Parameter</b>	container[N]  N is 0..32766 (no more than 64 units)
<b>Type</b>	Text String
<b>Value</b>	<p>Possible values of the parameter are strings in one of the following forms:</p> <ul style="list-style-type: none"> <li>▪ <b>Physical disk</b> <ul style="list-style-type: none"> <li>▪ <code>/dev/sd&lt;L&gt;</code> where "L" is letter.</li> <li>▪ <code>/dev/disk/by-id/...</code> - addressing by the disk ID, for example <code>/dev/disk/by-id/ata-ST1000DM003-9YN162_S1D01QJ4</code></li> <li>▪ <code>/dev/disk/by-label/...</code> - addressing by the disk label, for example <code>/dev/disk/by-label/MyStorage</code></li> <li>▪ <code>/dev/disk/by-uuid/...</code> - addressing by the disk UUID, for example <code>/dev/disk/by-uuid/0e808a2f-cdd3-4944-a245-f729ffd73882</code></li> </ul> <p> Be careful not to destroy all the information from the disk dedicated to CHARON-AXP by mistake.</p> <div style="border: 1px solid #f4a460; padding: 5px; margin: 10px 0;"> <p><code>/dev/sd&lt;L&gt;</code> addressing is not persistent, so it is strongly recommended to use <code>/dev/disk/by-id/wwn-*</code> syntax instead to refer the disk by its WWID - especially in the environments utilizing FC and SAN storages.</p> </div> <p>These disks must not be formatted by the host OS.</p> <p><b>Example:</b></p> <div style="border: 1px solid #ccc; padding: 5px; margin: 10px 0;"> <pre>set DIA container[0]="/dev/sdb"</pre> </div> <p>It is also possible to use not a whole disk, but previously created partitions on it. In this case the syntax is the following: <code>/dev/sd&lt;L&gt;&lt;N&gt;</code> where N is the number of partition to be used.</p> <p><b>Example:</b></p> <div style="border: 1px solid #ccc; padding: 5px; margin: 10px 0;"> <pre>set DIA container[0]="/dev/sdc1"</pre> </div> </li> <li>▪ <b>Multipath disk</b> <ul style="list-style-type: none"> <li>▪ <code>/dev/dm-&lt;N&gt;</code></li> <li>▪ <code>/dev/mapper/mpath&lt;N&gt;</code></li> <li>▪ <code>/dev/mapper/disk&lt;N&gt;</code></li> </ul> <p> Be careful not to destroy all the information from the disk dedicated to CHARON-AXP by mistake.</p> <p>These disks must not be formatted by the host OS.</p> <p><b>Example:</b></p> <div style="border: 1px solid #ccc; padding: 5px; margin: 10px 0;"> <pre>set DIA container[100]="/dev/dm-0"</pre> </div> </li> <li>▪ <b>Loop (virtual block) devices</b> <ul style="list-style-type: none"> <li>▪ <code>/dev/loop&lt;N&gt;</code></li> </ul> <p><b>Example:</b></p> <div style="border: 1px solid #ccc; padding: 5px; margin: 10px 0;"> <pre>set DIA container[200]="/dev/loop0"</pre> </div> </li> <li>▪ <b>Direct mapping to some SCSI disks</b> <ul style="list-style-type: none"> <li>▪ <code>/dev/sg&lt;N&gt;</code></li> </ul> <p><b>Example:</b></p> <div style="border: 1px solid #ccc; padding: 5px; margin: 10px 0;"> <pre>set DIA container[300]="/dev/sg0"</pre> </div> </li> </ul>



- **ISO file for reading distribution CD-ROM image**

- `[<drive>:\<path-name>\<file-name>[<file-name>].iso]`

Mapping may also include the full path (recommended), for example: `"/my_disks/vms_distributive.iso"`

**Example:**

```
set DIA container[600]="/my_disks/vms_distributive.iso"
```

- **File representing a physical disk of the HP Alpha system (disk image)**

- `[<drive>:\<path-name>\<file-name>[<file-name>.vdisk]`

These files can be created from scratch with "mkdiskcmd" 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 DIA container[401]="my_dka401.vdisk"
```

This parameter is initially not set, thus creating NO storage elements on the controller.

## removable

<b>Parameter</b>	removable[N] N is 0..32766 (no more than 64 units)
<b>Type</b>	Boolean
<b>Value</b>	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".  <b>Example:</b> <pre>set DIA removable[400]=true</pre>

## geometry

<b>Parameter</b>	geometry[N] N is 0..32766 (no more than 64 units)
<b>Type</b>	Text String
<b>Value</b>	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 <code>"*/*/&lt;B&gt;"</code> or <code>"*,*,&lt;B&gt;"</code> where B is the total size of the disk (in blocks) reported to the guest OS. If omitted it is calculated automatically.  If this parameter is not set, CHARON VM will configure the geometry based on the most probable disk type.  Initially not set.

## use\_io\_file\_buffering

<b>Parameter</b>	use_io_file_buffering[N] N is 0..32766 (no more than 64 units)
<b>Type</b>	Boolean
<b>Value</b>	<p>When set to "true", instructs CHARON-AXP to enable host operating system I/O cache on reading/writing operations.</p> <p><b>i</b> Note that this caching has a significant effect only in case of mapping to disk and tape containers, not physical drives.</p> <p>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.</p> <p>Initially is set to "false".</p> <p><b>Example:</b></p> <pre>set DIA use_io_file_buffering[603]=true</pre>

When a disk image connected to an I/O Bypass controller is disconnected 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 "mkdiskcmd" 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 version.

[Back to Table of Contents](#)

## Deinstallation

1. Do a conversational boot. Please refer to your OpenVMS system administration guide for instructions.
2. Set the NOAUTOCONFIG system parameter to 1
3. Boot OpenVMS
4. Remove the product with the "\$ PRODUCT REMOVE CHARON\_DISK" command.
5. Set the NOAUTOCONFIG system parameter to 0 and reboot.

[Back to Table of Contents](#)

## Finding the target "/dev/sg" device

### Table of Contents

- General description
- Procedures of finding the target "/dev/sg" device
  - First method
  - Second method

[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

- General description
- Configuration steps
- Configuration parameters
  - interface
  - station\_address
  - rx\_fifo\_size
  - adapter\_mode
  - Example
  - DE602 and DE602AA network adapters link speed and duplex settings
- Packet Port
  - interface
  - port\_enable\_mac\_addr\_change
  - port\_retry\_on\_tx
  - port\_pending\_rx\_number
  - port\_pending\_tx\_number
  - log
  - Example

[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.

 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
<code>load DE500BA/dec21x4x NIC</code>	<code>load DE602/i8255x NIC</code>

In AlphaStation 400 configuration use the following syntax for network adapter loading:

```
load DE500AA/dec21x4x NIC irq_bus=isa
```

**i** By default each loaded virtual network 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.

## 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
```

The interface name can be either "(disabled)" for a disabled interface or "<network interface name>"

**Examples:**

```
load packet_port/chnetwrk NIC1 interface="(disabled)"
```

```
load packet_port/chnetwrk NIC2 interface="ens33"
```

[Back to Table of Contents](#)

## Configuration parameters

Each virtual network controller has the following parameters that are specified with the "set" command:

### interface

<b>Parameter</b>	interface
<b>Type</b>	Text String
<b>Value</b>	Name of the corresponding instance of the "packet_port" or "tap_port" component

### station\_address

<b>Parameter</b>	station_address
<b>Type</b>	Text String
<b>Value</b>	<p>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.</p> <p>Format:</p> <pre>XX-XX-XX-XX-XX-XX</pre> <p>or</p> <pre>XX:XX:XX:XX:XX:XX</pre> <p><b>Example:</b></p> <pre>set EWA station_address="AF:01:AC:78:1B:CC"</pre>

## rx\_fifo\_size

<b>Parameter</b>	rx_fifo_size
<b>Type</b>	Numeric
<b>Value</b>	<p>"rx_fifo_size" sets the receive FIFO size.</p> <p>The value is specified in Kb and, by default, is pre-calculated from the connected port's size of the receive queue.</p> <p>Typically, you do not need to change the "rx_fifo_size" parameter. It is available for extended tuning and debugging purposes.</p>

## adapter\_mode

<b>Parameter</b>	adapter_mode												
<b>Type</b>	Text String												
<b>Value</b>	<p>Defines the link speed and the duplex settings of the virtual network adapter (except for DE602/DE602AA - see <a href="#">below</a>).</p> <p>The values are:</p> <table border="1"> <thead> <tr> <th>Parameter</th> <th>Description</th> </tr> </thead> <tbody> <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> </tbody> </table> <p>Defines the link speed and the duplex settings of the virtual network adapter (except for DE602/DE602AA - see <a href="#">below</a>).</p> <p>The values are:</p> <p><b>Example:</b></p> <pre>set EWA adapter_mode="100BaseT-HD"</pre>	Parameter	Description	"Auto"	Auto-negotiate (default)	"10BaseT-HD"	10Mbps half duplex	"10BaseT-FD"	10Mbps full duplex	"100BaseT-HD"	100Mbps half duplex	"100BaseT-FD"	100Mbps full duplex
Parameter	Description												
"Auto"	Auto-negotiate (default)												
"10BaseT-HD"	10Mbps half duplex												
"10BaseT-FD"	10Mbps full duplex												
"100BaseT-HD"	100Mbps half duplex												
"100BaseT-FD"	100Mbps full duplex												

## 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 ppl
```

"packet\_port" uses several configuration parameters to control its behavior.

## interface

<b>Parameter</b>	interface
<b>Type</b>	Text string
<b>Value</b>	<p>This parameter identifies an Ethernet adapter of the host system dedicated to CHARON-AXP.</p> <p>Syntax:</p> <pre>set &lt;name&gt; interface="&lt;adapter&gt;"</pre> <p><b>Example:</b></p> <pre>set ppl interface="eth1"</pre>



## port\_enable\_mac\_addr\_change

<b>Parameter</b>	port_enable_mac_addr_change
<b>Type</b>	Boolean
<b>Value</b>	<p>If "true" is specified (default value), CHARON-AXP sets the appropriate Ethernet address automatically.</p> <p>If "false" is specified, set the Ethernet address manually.</p> <p><b>Example:</b></p> <pre>set ppl port_enable_mac_addr_change=false</pre>

## port\_retry\_on\_tx

<b>Parameter</b>	port_retry_on_tx
<b>Type</b>	Numeric
<b>Value</b>	<p>The "port_retry_on_tx" parameter controls the number of times a port will attempt to transmit a packet before giving up.</p> <p>By default, the value is 3.</p> <p>Increasing this value may introduce problems in carrier loss logic, because not all NIC drivers support a carrier status query.</p> <p>Typically, you do not need to increase the value.</p> <p><b>Example:</b></p> <pre>set ppl port_retry_on_tx=8</pre>

## port\_pending\_rx\_number

<b>Parameter</b>	port_pending_rx_number
<b>Type</b>	Numeric
<b>Value</b>	<p>The "port_pending_rx_number" parameter sets the number of pending receive buffers.</p> <p>The default value is 63. The maximum value allowed is 195.</p> <p>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.</p> <p>Typically, you do not need to change this parameter.</p> <p><b>Example:</b></p> <pre>set ppl port_pending_rx_number=128</pre>

## port\_pending\_tx\_number

<b>Parameter</b>	port_pending_tx_number
<b>Type</b>	Numeric
<b>Value</b>	<p>The "port_pending_tx_number" parameter sets the number of buffers the port uses to transmit.</p> <p>The default value is 62.</p> <p>You may want to increase the "port_pending_tx_number" value if the log file indicates dropped TX packets due to TX queue overflow.</p> <p>Typically, you do not need to change this parameter.</p> <p><b>Example:</b></p> <pre>set ppl port_pending_tx_number=128</pre>

## log

<b>Parameter</b>	log
<b>Type</b>	Text string
<b>Value</b>	<p>If this parameter is set to some valid file name or a directory where the log files for each individual session will be stored CHARON logs Recv and Xmit packets at the emulated port layer.</p> <p>If an existing directory is specified, CHARON 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 does not create log.</p> <p>In certain situations enabling this parameter may help to detect loss of packets.</p> <p><b>Example:</b></p> <pre>set ppl log="ppl.log"</pre> <pre>set ppl log="/charon/logs"</pre> <p>Only existing directory can be used as a value of the "log" parameter.</p>

## Example

```
load DE500BA/dec21x4x EWA
load packet_port/chnetwrk ppl interface="eth1"
set EWA interface=ppl
```

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
- HP AlphaServer 4000 configuration file
- HP AlphaServer DS20 configuration file
- HP AlphaServer ES40 configuration file
- HP AlphaServer GS80 configuration file

## HP AlphaServer 800 configuration file

```

#
# Copyright (C) 1999-2016 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) 1999-2016 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) 1999-2016 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=EWCO
#load packet_port/chnetwrk EWCO 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) 1999-2016 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=EWCO
#load packet_port/chnetwrk EWCO 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) 1999-2016 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 the 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 aksusbd `rpm -q -a | grep charon`
```

[Back to Table of Contents](#)

# Appendixes

## Contents

- glibc.i686 installation without Internet connection
- How to implement time synchronisation between CHARON-AXP Host OS and Guest OS



# glibc.i686 installation without Internet connection

## Contents

- Description
- Step-by-step guide
  - Introduction
  - Installation steps
    - Mounting the DVD
    - Installation using the rpm utility
    - Installation using the yum utility

## Description

The `glibc.i686` package is required for the CHARON-AXP, CHARON-VAX, CHARON-SSP and CHARON-HPA products. When connected to the Internet, and – for Red Hat Enterprise Linux – registered, this package can be installed as described in the product documentation. Usually the following command is used:

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

When an Internet connection is not available, or if a server running Red Hat Enterprise Linux is not registered, this package can be installed using the operating system installation DVD or ISO file.

This document explains how to install the package from the installation DVD for the following Linux distributions:

- Red Hat Enterprise Linux 6.x and 7.x
- CentOS 7.x - Everything distribution DVD only.
  - ⚠ The Standard distribution DVD of CentOS 7.x does not contain the `glibc.i686` package so the Everything distribution DVD is mandatory if no internet connection is available.

**i** There are several different solutions for installing `glibc.i686` without Internet connection. This document explains some of them. For more information, please contact your system administrator or refer to the administrator's guide for the Linux distribution installed on your CHARON host system.

## Step-by-step guide

### Introduction

Check first if the `glibc.i686` package is installed using the following command:

```
# yum list installed glibc.i686
```

If the package is not installed, the command will report the following message (examples given for Red Hat Enterprise Linux 7.2)

```
Loaded plugins: langpacks, product-id, search-disabled-repos, subscription-manager
This system is not registered to Red Hat Subscription Management. You can use subscription-manager to register.
Error: No matching Packages to list
```

If the package is installed, the command will report the following message:

```
Loaded plugins: langpacks, product-id, search-disabled-repos, subscription-manager
This system is not registered to Red Hat Subscription Management. You can use subscription-manager to register.
Installed Packages
glibc.i686                2.17-105.e17                @<source>
```

Alternatively, the following rpm command can be used:

```
# rpm -qa | grep glibc | grep i686
```

? If the command reports nothing, the package is not installed.

## Installation steps

### Mounting the DVD

Please either insert the installation DVD in the drive or use an ISO file.

If the DVD is mounted automatically, we recommend to unmount it and mount it manually on a mount point with no spaces in its name (the yum-config-manager utility does not handle spaces correctly).

#### Example / Red Hat Enterprise Linux 7.2:

```
# df -k
Filesystem            1K-blocks    Used Available Use% Mounted on
/dev/mapper/rhel-root 37300436 4376776 32923660 12% /
devtmpfs              1925960      0 1925960  0% /dev
tmpfs                 1941228      96 1941132  1% /dev/shm
tmpfs                 1941228     9132 1932096  1% /run
tmpfs                 1941228      0 1941228  0% /sys/fs/cgroup
/dev/sda1             508588 160220 348368 32% /boot
tmpfs                 388248      12 388236  1% /run/user/1000
tmpfs                 388248      0 388248  0% /run/user/0
/dev/sr0              3947824 3947824      0 100% /run/media/stromasys/RHEL-7.2 Server.x86_64
# umount /dev/cdrom
# mkdir -p /media/cdrom
# mount -r /dev/cdrom /media/cdrom
```

i By default /dev/cdrom is linked to /dev/sr0.

If you have an ISO image of the distribution CD, you can mount it using a loopback device:

```
# mount /path/to/ISO-image.iso /media/cdrom -o loop
```

## Installation using the rpm utility

If you use rpm to install the package, any packages on which it depends must be installed manually.

To do so:

1. Switch to the directory containing the packages. This directory depends on your Linux distribution.

**Example:**

```
# cd /media/cdrom/Packages/
```

2. Locate the target "glibc.i686" package and the package on which it depends:

```
# ls -l glibc*i686* nss-softokn-freebl*
```

3. Install the "glibc.i686" package and any others on which it depends (package versions may differ):

**Example / Red Hat Enterprise Linux 7.2:**

```
# rpm -i glibc-2.17-105.el7.i686.rpm nss-softokn-freebl-3.16.2.3-13.el7_1.i686.rpm
```

**Example / CentOS7:**

```
# rpm -i glibc-2.17-157.el7.i686.rpm nss-softokn-freebl-3.16.2.3-14.4.el7.i686.rpm
```

- If the command returns a warning message related to RSA/SHA256 signature, please ignore it. The successful installation of the package can be checked using the "yum list installed glibc.i686" command.

- If the installation command above reports additional unsatisfied dependencies, add the corresponding packages to the above command line.

4. Unmount the CD-ROM or ISO file if necessary:

```
# cd -  
# umount /media/cdrom
```

## Installation using the yum utility

The yum utility will check and install all the necessary dependencies.

Define the operating system installation DVD as a new repository:

### Example / Red Hat Enterprise Linux 7.2:

```
# yum-config-manager --add-repo=file:///media/cdrom

Loaded plugins: langpacks, product-id
adding repo from: file:///media/cdrom

[media_cdrom]
name=added from: file:///media/cdrom
baseurl=file:///media/cdrom
enabled=1

# yum --nogpgcheck install glibc.i686

...
Is this ok [y/d/N]: y

...
Complete!
```

Once the installation is complete, the repository can be disabled (if no other package has to be installed):

```
# yum-config-manager --disable media_cdrom
```

Using this method, installing other packages could require the gpg check to be disabled. To do so:

- either add the `--nogpgcheck` parameter to the `yum install` commands.

#### Example:

```
# yum --nogpgcheck install <package>
```

- or disable the GPG check for the repository in the `/etc/yum.repos.d/media_cdrom.repo` file, by adding the `gpgcheck=0` line
- or disable the GPG check in the `/etc/yum.conf` file, replacing the `gpgcheck=1` line by `gpgcheck=0`


### Example / CentOS 7:

```
# yum --disablerepo=* --enablerepo=c7-media install glibc.i686

...
Is this ok [y/d/N]: y

...
Is this ok [y/d/N]: y

...
Complete!
```

 The `c7-media` repository is a default repository on CentOS 7.x pointing to the `/media/cdrom` folder. Please refer to the administrator's documentation for more information.

# How to implement time synchronisation between CHARON-AXP Host OS and Guest OS



## Table of contents

- Description
- Step-by-step guide
  - Configuration file settings
  - Virtual machine operating system settings
    - On OpenVMS/AXP
      - Using a batch queue
      - Using a detached process
      - Considerations using DECnet-Plus software
    - On Tru64 UNIX
- Related articles

## Description

This document will explain how to implement the time synchronization feature using the "sync\_to\_host" parameter in the configuration file. This parameter allows to keep TOY time always synchronized with the host's time and disable undesirable updates to the TOY from guest OS.

**Restrictions:** Minimum product versions/builds required:

Windows	 CHARON-AXP V4.4 Build 148-02 with patch 148-09 installed
	 CHARON-AXP V4.5 Build 153-03 and 153-05 (patched)
	 CHARON-AXP V4.6 Build 166-03 and later
Linux	 CHARON-AXP V4.6 Build 168-03 and later

## Step-by-step guide


### Configuration file settings

Update the configuration file with the following settings:

**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	Forbid updates to the TOY from the guest OS   If you want guest to synchronize itself using DTSS or NTP for example, remove "nowrite"


**Example:**

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

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

On OpenVMS/AXP	On Tru64 UNIX
<pre>\$ set time</pre>	<pre># date -u `consvar -g date   cut -f 3 -d ' '`</pre>

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.

 The CHARON virtual machine must be restarted in order to take the new parameter into account

## Virtual machine operating system settings

The commands mentioned above used to synchronize the guest OS with TOY are effective only when they are executed. To avoid time difference, these commands must be executed at specified intervals.

You will find below examples on how to implement scripts to perform time synchronization for OpenVMS and Tru64 UNIX.

 If you have NTP running on your OpenVMS or Tru64 UNIX system, you can keep it running even if sync\_to\_host is enabled

### On OpenVMS/AXP

You need first to perform a manual synchronization between the CHARON server and the CHARON virtual machine using the SET TIME= command:

```
$ SET TIME=12:30:00
```

You can use either a batch queue or a detached process to synchronize time. The two methods are described below.

## Using a batch queue

- Create a simple script containing the following lines. The example below will sync time every hour:

**i** In our example, we will create the script in the `SYS$MANAGER` folder and name it `CHARON_SYNCTIME.COM`. The OS version used is OpenVMS 7.3-2

```
$ EDIT SYS$MANAGER:CHARON_SYNCTIME.COM

$ SET NOON
$ SET VERIFY
$ LOOP:
$ SHOW TIME
$ SET TIME
$ SHOW TIME
$ WAIT 01:00:00
$ GOTO LOOP
```

**i** The "`$ SET VERIFY`" line is optional, just used for verifying commands are correctly executed

- A batch queue will be required to create the job:
  - Find an available batch queue or create a new one
  - Execute the following command to view available batch queues (refer to OpenVMS documentation)

```
$ SHOW QUEUE /ALL /BATCH
```

If the command returns the following error message: "%JBC-E-JOBQUEDIS, system job queue manager is not running", you will need to initialize the queue manager:

```
$ START /QUEUE /MANAGER /NEW
%%%%%%%%%% OPCOM 29-MAY-2015 12:30:07.36 %%%%%%%%%%%
Message from user SYSTEM on VMS732
%JBC-I-CREATED, SYS$COMMON:[SYSEXE]QMAN$MASTER.DAT; created
```

- Create a dedicated batch queue for the synchronization job (recommended):

```
$ INIT /QUEUE /BATCH /START SYS$SYNCTIME /JOB_LIMIT=1
$ SHOW QUEUE SYS$SYNCTIME /FULL
Batch queue SYS$SYNCTIME, idle, on VMS732::
  /BASE_PRIORITY=4 /JOB_LIMIT=1 /OWNER=[SYSTEM] /PROTECTION=(S:M,O:D,G:R,W:S)
```

- Submit the job:

```
$ SUBMIT /QUEUE=SYS$SYNCTIME SYS$MANAGER:CHARON_SYNCTIME
Job CHARON_SYNCTIME (queue SYS$SYNCTIME, entry 1) started on SYS$SYNCTIME
```

- Update the systartup script to start the SYS\$SYNCTIME queue and the job at system boot. The two following lines will have to be added at the very end of the script (for example):

```
$ EDIT SYS$STARTUP:SYSTARTUP_VMS.COM

...
$ START /QUEUE SYS$SYNCTIME
$ SUBMIT /QUEUE=SYS$SYNCTIME SYS$MANAGER:CHARON_SYNCTIME
$
$ EXIT
```


**i** Prior to OpenVMS version 6, the systartup script will be named: SYS\$STARTUP:SYSTARTUP\_V<N>.COM where N represents the version number

- Example for OpenVMS Version 5 : SYS\$STARTUP:SYSTARTUP\_V5.COM



## Using a detached process

- Create a simple script containing the following lines. The example below will sync time every hour


 In our example, we will create the script in the `SYS$MANAGER` folder and name it `CHARON_SYNCNCTIME.COM`. The OS version used is OpenVMS 7.3-2

```

$ EDIT SYS$MANAGER:CHARON_SYNCNCTIME.COM

$ PROC = F$ENVIRONMENT ("PROCEDURE")
$ IF F$MODE() .EQS. "INTERACTIVE"
$ THEN
$   RUN SYS$SYSTEM:LOGINOUT /AUTHORIZE -
      /DETACH /UIC=[SYSTEM] -
      /PROCESS_NAME="TIME SYNC" -
      /OUTPUT=NL: /ERROR=NL: -
      /INPUT= 'PROC '
$ ELSE
$   SET NOON
$   SET VERIFY
$LOOP:
$   SHOW TIME
$   SET TIME
$   SHOW TIME
$   WAIT 01:00:00
$   GOTO LOOP
$ ENDIF

```

 The "`$ SET VERIFY`" line is optional, just used for verifying commands are correctly executed. If you let it active, please replace the "NL:" device above by a log file name


- Update the systartup script to start the detached process at system boot. The following line will have to be added at the very end of the script (for example):

```

$ EDIT SYS$STARTUP:SYSTARTUP_VMS.COM

...
$ @SYS$MANAGER:CHARON_SYNCNCTIME
$
$ EXIT

```

 To start the job manually without a reboot, just execute the line above from an interactive session.

## Considerations using DECnet-Plus software

⚠ If you have DECnet-Plus software installed, you will have to disable DTSS before setting time in order to avoid errors like:

```
%SET-E-NOTSET, error modifying time
-SYSTEM-E-TIMENOTSET, time service enabled; enter a time service command to update the time
```

💡 To disable DTSS, you will have to update the CHARON\_SYNCNCTIME.COM script (new lines are green colored):

```
$ SET NOON
$! SET VERIFY
$ LOOP:
$ SHOW TIME
$ RUN SYS$SYSTEM:NCL
DISABLE DTSS
DELETE DTSS
EXIT
$ SET TIME
$ SHOW TIME
$ @SYS$STARTUP:DTSS$STARTUP
$ WAIT 01:00:00
$ GOTO LOOP
```

💡 It is recommended here not to use "SET VERIFY"

📘 More information here: [OpenVMS Frequently Asked Questions](#) (credits: Hewlett-Packard)

## On Tru64 UNIX

**Restrictions:** The synchronization requires the "consvar" command to be available on the Tru64 operating system thus Tru64 UNIX version 4.0F minimum is required

- You need first to perform a manual synchronization between the CHARON server and the CHARON virtual machine using the `date` command.  
Example:

```
# date -u 05291724
```


- Create an entry in the root's `crontab` file as shown below (new lines are green colored):

```
# crontab -e
...
# CHARON time sync_to_host
00 * * * * /sbin/date -u ` /sbin/consvar -g date | cut -f 3 -d ' '`
```

⚠ Full path to `date` and `consvar` commands must be specified

💡 The above command will be executed at minute 0 of every hour. If you want to execute this every 15 minutes for example, use the following line instead:

```
00/15 * * * * /sbin/date -u ` /sbin/consvar -g date | cut -f 3 -d ' '`
```

 [More information at Wikipedia.org - Cron](#)

## Related articles

- [How to implement time synchronisation between CHARON-AXP Host OS and Guest OS \(sync\\_to\\_host\)](#)