



# CHARON-AXP V4.6 for Windows - Users Guide



# Contents

---

Introduction	3
Conventions	6
CHARON-AXP for Windows installation	7
Running CHARON-AXP for Windows	35
CHARON-AXP for Windows configuration	64
Migration to CHARON-AXP for Windows	88
CHARON-AXP for Windows licensing	99
CHARON-AXP for Windows utilities	116
CHARON Launcher	117
CHARON Service Manager	127
Host Device Check	143
Network Control Center	146
MkDisk	179
MkDskCmd	188
HASP View	191
License Update Service	193
mtd	198
CHARON Log Monitor and Dispatcher	199
IDLE Utility	201
HOSTPrint	203
CHARON-AXP for Windows configuration details	205
General Settings	206
Core Devices	212
Console	230
Placement of peripheral devices on PCI bus	238
Disks and tapes	279
KZPBA PCI SCSI adapter	280
KGPSA-CA PCI Fibre Channel adapter	288
Acer Labs 1543C IDE/ATAPI CD-ROM adapter	306
Networking	308
DEFPA PCI FDDI adapter	316
PBXDA-xx series PCI serial adapters	321
Sample configuration files	326
HP AlphaServer 800 configuration file	327
HP AlphaServer 4000 configuration file	333
HP AlphaServer DS20 configuration file	338
HP AlphaServer ES40 configuration file	344
HP AlphaServer GS80 configuration file	349
CHARON-AXP for Windows deinstallation	354

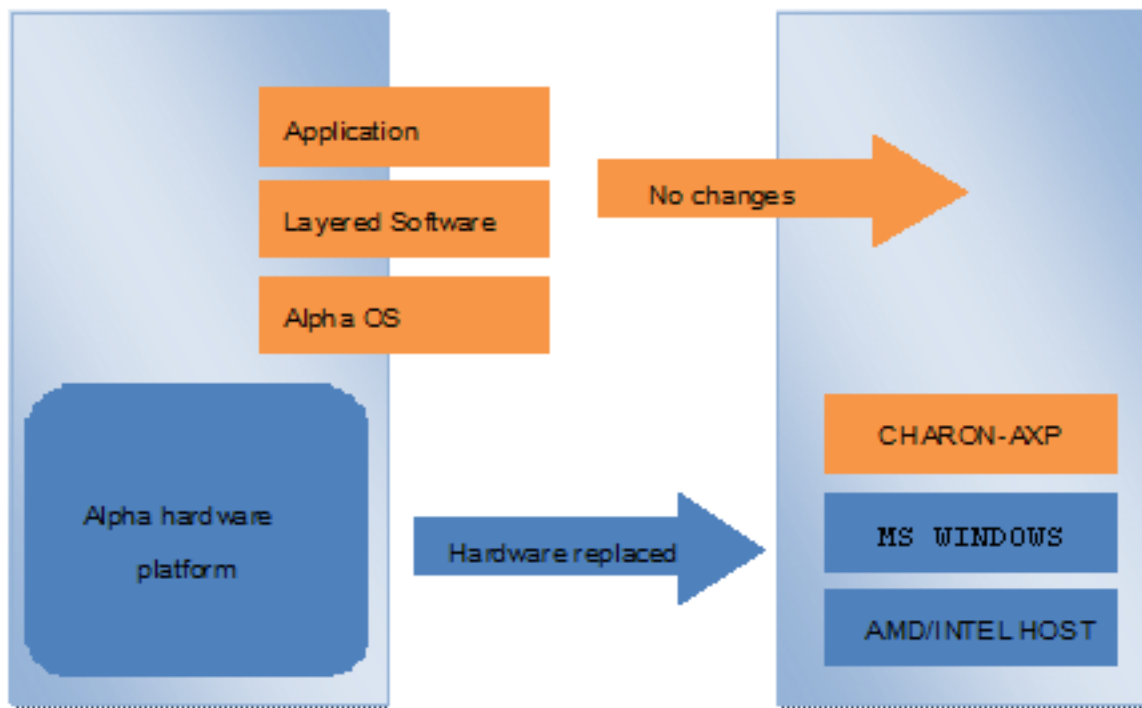
# Introduction

## 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. Our HP Alpha emulator product is currently available in the following variants:

**CHARON-AXP/4100** includes:

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

**CHARON-AXP/DS10** includes:

- AlphaServer DS10
- AlphaServer DS15

**CHARON-AXP/DS20** includes:

- AlphaServer DS20
- AlphaServer DS25

**CHARON-AXP/ES40** includes:

- AlphaServer ES40
- AlphaServer ES45

**CHARON-AXP/GS80** include:

- AlphaServer GS80

**CHARON-AXP/GS160** includes:

- AlphaServer GS160

**CHARON-AXP/GS320** includes:

- AlphaServer GS320

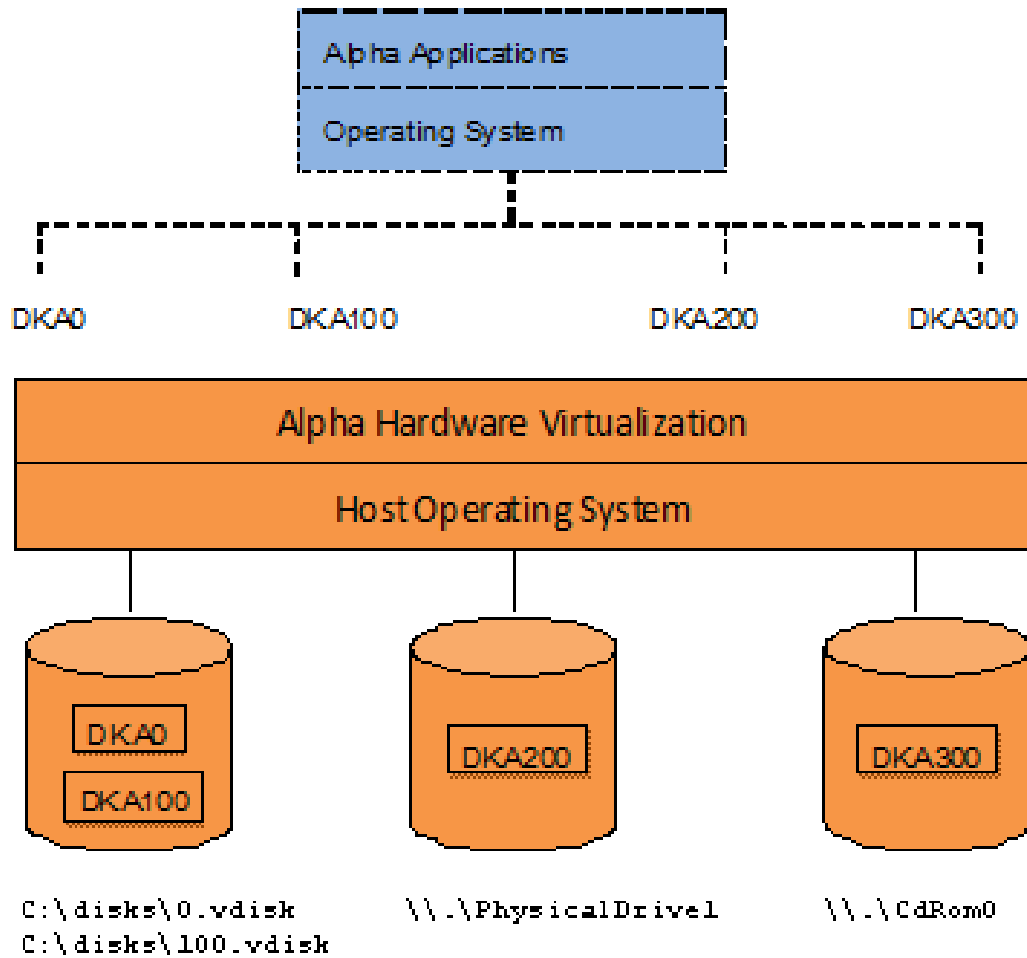
The following table explains which hardware boards CHARON virtualizes:

Subsystem	Covered HP Alpha hardware
Serial Lines Controllers	On-board serial line port COM2, Family of PCI controllers: PBXDA-BA, PBXDA-BB, PBXDA-AC, PBXDA-AC
IDE/ATAPI CD-ROM Controller	Virtual Acer Labs 1543C
PCI Fibre Channel Controller	KGPSA-CA
PCI SCSI Controller	KZPBA
PCI FDDI Controller	DEFPA
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 Windows 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.
	This symbol represents the Enter key without typed user input. Used, for example, to tell the user to select the default value by pressing enter.

The following definitions apply:

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


# CHARON-AXP for Windows 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
- License installation
  - Regular HASP USB dongle
  - Network HASP USB dongle
  - Software license
- License validity verification
  - Troubleshooting
- Network configuration
  - For Windows 2008 R2, Windows 2012 R2, Windows 7, 8 and 8.1
  - For older versions of Windows
- After installation

## Introduction

CHARON-AXP products are distributed in the form of a zip file that contains the CHARON-AXP installation executable ("InstallShell.exe"). The InstallShell procedure offers the possibility to select several modules to install. It is recommended to install all modules (the default).

 Before installing CHARON-AXP, please perform all hardware and software host system checks to ensure the host platform meets minimum CHARON-AXP installation requirements

Optionally, create a specific CHARON username with administrative privileges

CHARON-AXP installation steps:

- Extract the CHARON-AXP installation procedure files from its archive
- Install CHARON-AXP by running "InstallShell.exe" installer and follow the instructions
- Install CHARON-AXP license (hardware dongle or software license)
- Configure CHARON-AXP host system, for example, network adapter, disable some Windows features (see [Host system preparation](#) further), etc.

[Back to Table of Contents](#)

## Hardware Requirements

### Number of CPU cores

Each CHARON-AXP emulated CPU requires a corresponding physical core. The total number of 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 fulfil host operating system needs. If several CHARON instances run in parallel, the required number of CPU cores is summarized.

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

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

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

## CPU type and speed

Since CHARON-AXP utilizes LAHF instructions during the HP Alpha CPU emulation, please avoid using early (pre-2005) AMD64 and Intel 64 CPUs for the CHARON host system since they lack this capability. AMD Athlon 64, Opteron and Turion 64 revision D processors from March 2005 and Intel Pentium 4 G1 stepping from December 2005 are LAHF instruction capable.

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

## Operative memory

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

The minimum host memory is calculated according to the following formula:

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

## Disk storage

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 space required for the normal host OS. Temporary disk storage is often needed when setting up a new server, for saveset storage, software installation kits, etc.

## Ethernet adapters

CHARON-AXP networking assumes dedicated host Ethernet adapters; their number must be equal to the emulated adapters to be configured in CHARON-AXP. One adapter (optionally) can be left to the host for TCP/IP networking etc.

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

[Back to Table of Contents](#)



## Software Requirements

- Windows Server 2012 (R2) Standard Edition , 64 bit version
- Windows 8.1 Pro and Enterprise Editions, 64 bit versions
- Windows 7 Professional and Ultimate (SP1) Editions, 64 bit versions
- Windows Server 2008 R2 (SP1) Standard and Enterprise Editions, 64 bit version

[Back to Table of Contents](#)

## Host system preparation

All antivirus, screen saver, automatic (scheduled) backup and any other CPU consuming software as well as software that is able to temporarily lock CHARON files (such as automatic indexing) must be turned off, uninstalled and disabled.

The power scheme must be set to "High Performance" with all the "Sleep" and "Standby" modes turned off.

The automatic installation of Microsoft updates must be disabled. MS 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.

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

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

Please consult with your Windows User's Guide for details.

If stricter firewall rules are required, it is possible to open the ports 30000-65535 and 1947 only for the "Sentinel HASP License Manager" (hasplms.exe) service (it will be installed by CHARON-AXP ).

In case if any magneto-optical (MO) drive installed on host system is going to be used with CHARON (mapped as "\\.\PhysicalDrive<N>" to some of CHARON emulated disk controllers) the host "MediaChangeNotification" (MSN) service must be switched off manually for this drive according to the following procedure:

1. Type "regedit" in the search field under "Start" menu or press Windows+R and enter "regedit"; press Enter to run the found applet.
2. Find the "AlwaysDisableMCN" parameter in the following way:

```
HKEY_LOCAL_MACHINE
+ SYSTEM
  + CurrentControlSet
    + Enum
      + SCSI
        + Disk&Ven_DEC&Prod_RWZ53____(C)DEC <<< This one depends on Vendor
and Model of your MO drive
          + 5&fd233cf&0&000500 <<< This one depends on
hardware connection (SCSI ID of MO drive, location of SCSI HBA on the host's
mainboard, etc ...)
            + Device Parameters
              + MediaChangeNotification
                + AlwaysDisableMCN = DWORD:00000000
```

3. Change the "AlwaysDisableMCN" parameter to "1", for example: "AlwaysDisableMCN = DWORD:00000001".
4. Reboot the host system.

[Back to Table of Contents](#)

## Before installation

1. Login as the local system administrator ("Administrator") on the host system.
2. Create a special user for running CHARON-AXP (optionally). This user must have all the administrative privileges. Please consult with your Windows User's Guide on details.

For correct work of CHARON the installation procedure will create a special group called CHARON-GRP and the current user will be automatically included to this group. Do not remove this group and do not extract CHARON user from it, otherwise a [problem with virtual memory allocation](#) may appear on CHARON start.



The created user may belong to some domain, but please note that in this case you have to add this user to the CHARON-GRP manually as it is described in [this article](#) and then reboot CHARON host.

3. Stay logged in as local system administrator ("Administrator") or log off and login as the CHARON-AXP user having administrative privileges (if this option has been chosen).
4. Create a special directory for the CHARON-AXP distribution kit and copy the provided files there.

[Back to Table of Contents](#)

## Distribution preparation

1. Extract the contents of the distribution archive to the current directory or copy the contents of the provided distribution directory to the current directory.
2. Run the "InstallShell.exe" file:

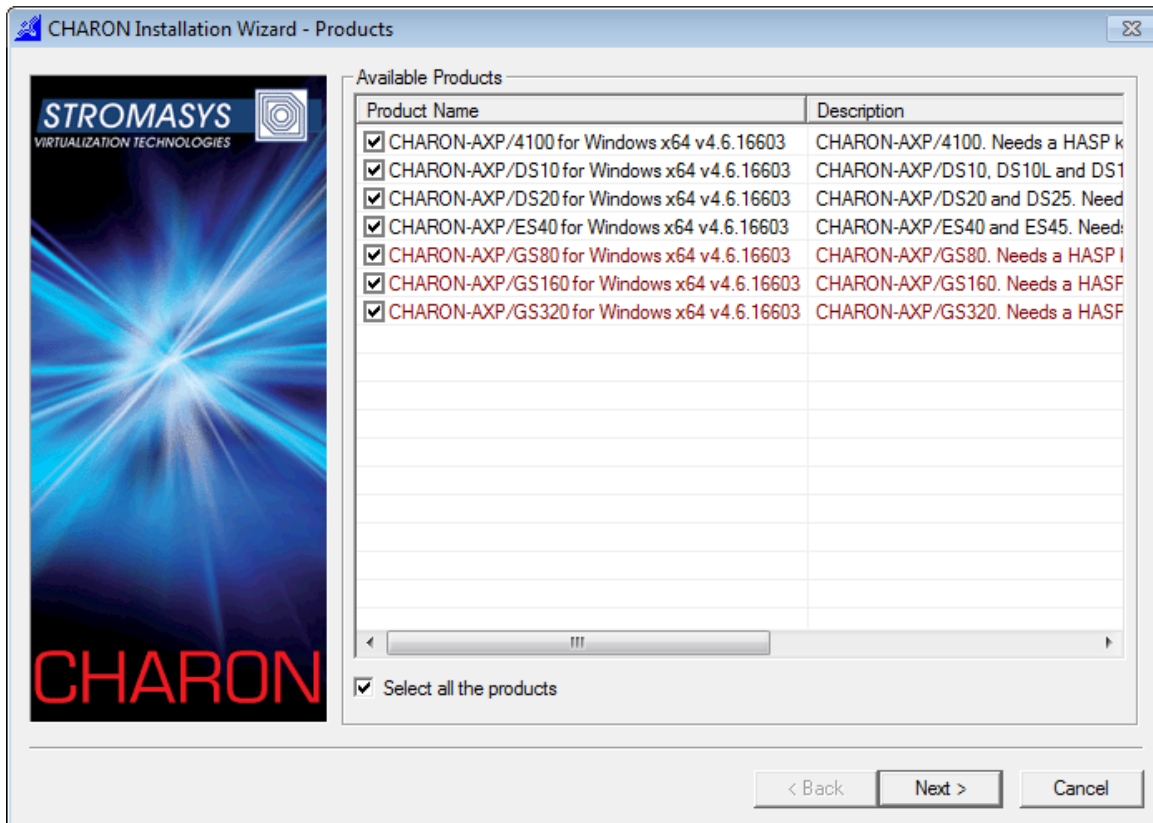
 idle_vms_pkg_2.0.def	19.03.2012 10:13
 InstallShell.exe	03.02.2014 14:00
 InstallShellCHS.dll	03.02.2014 14:00

[Back to Table of Contents](#)

## Installation

The following description assumes this is the first installation of CHARON-AXP on the target host. Installation of additional CHARON products follows the same procedure.

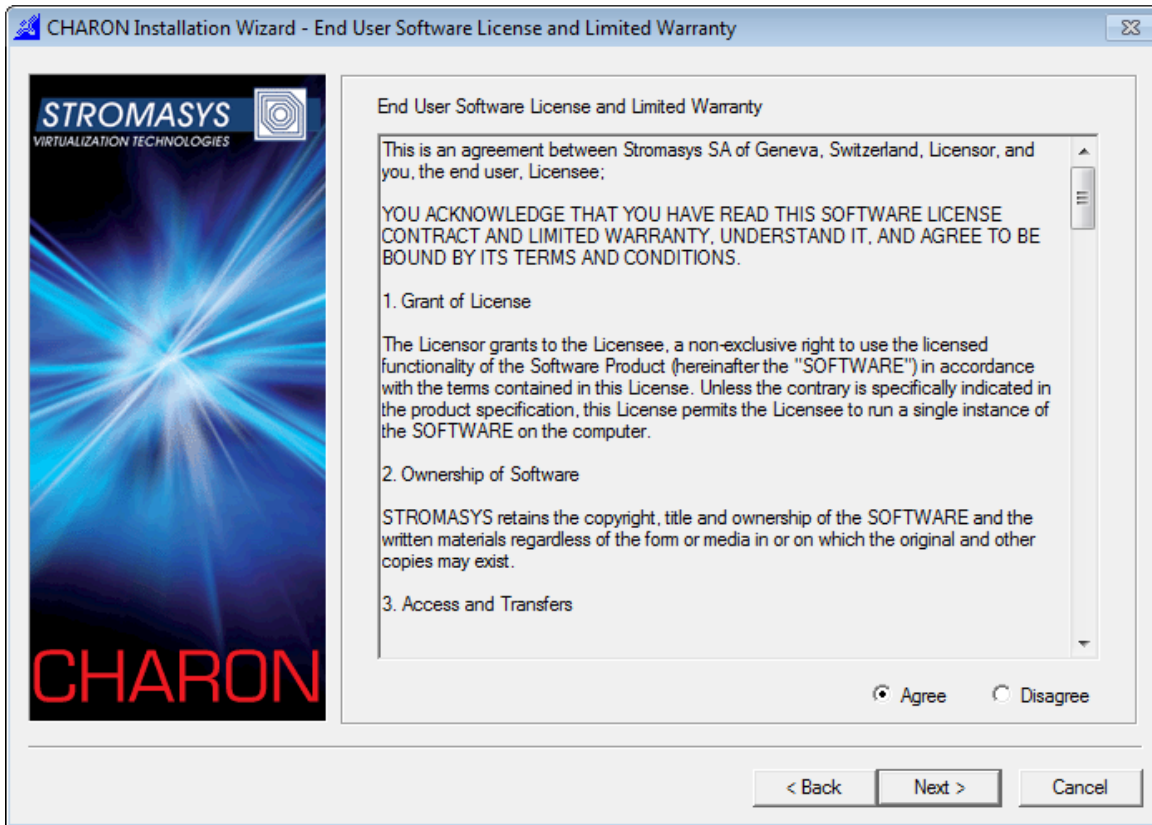
The first dialog lists the CHARON-AXP products for this particular distribution:



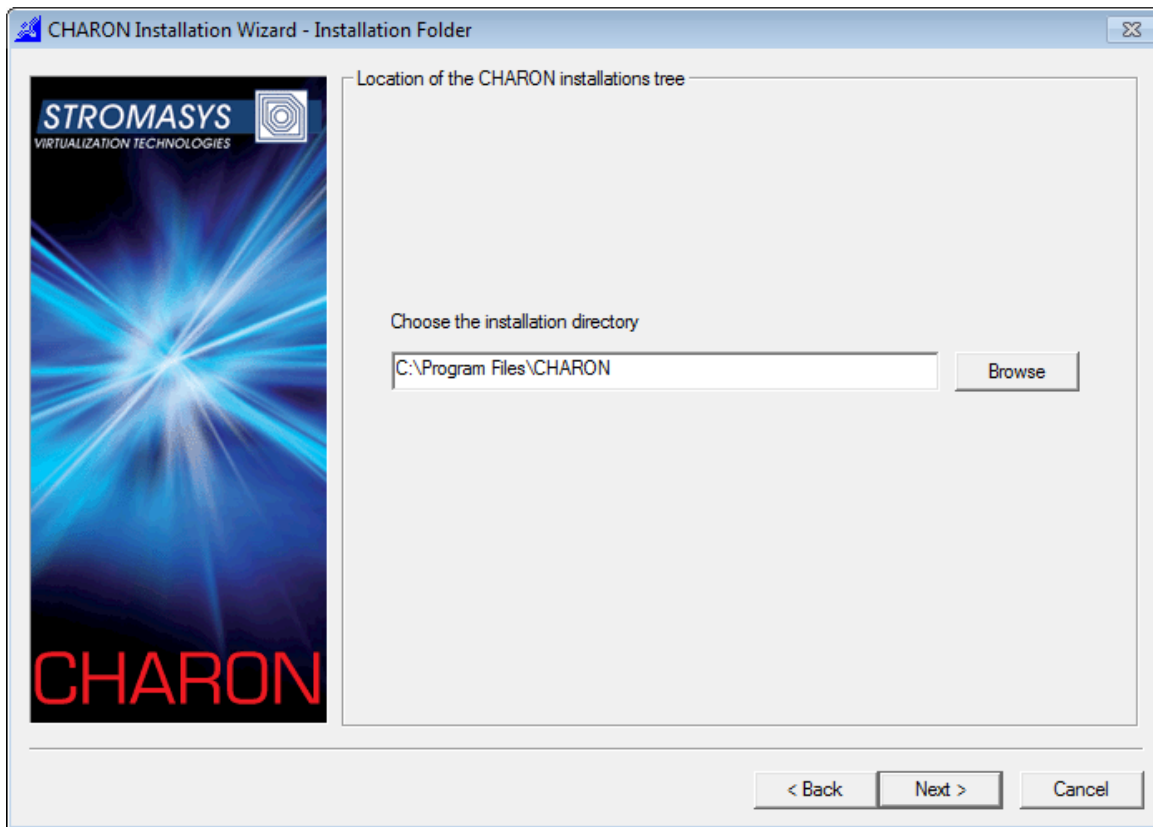
Select the required products or all products by checking the corresponding boxes.

Some of the listed products may be highlighted with a red color. This indicates that the CHARON host configuration does not meet CHARON minimal requirements for this particular product and cannot be installed. To see what requirements are not met, point the mouse cursor on that product and a pop-up box with details will appear.

Press the "Next" button.

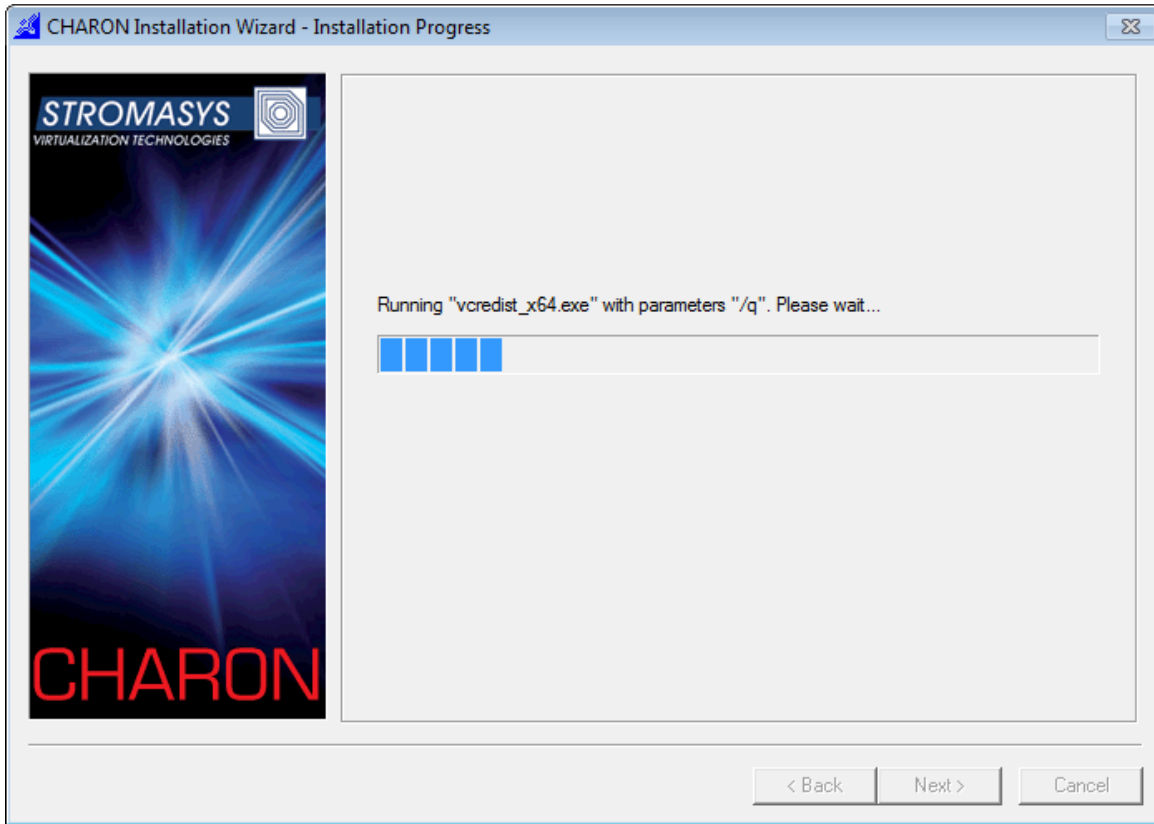


Read the license, check "Agree" if you agree and press the "Next" button.

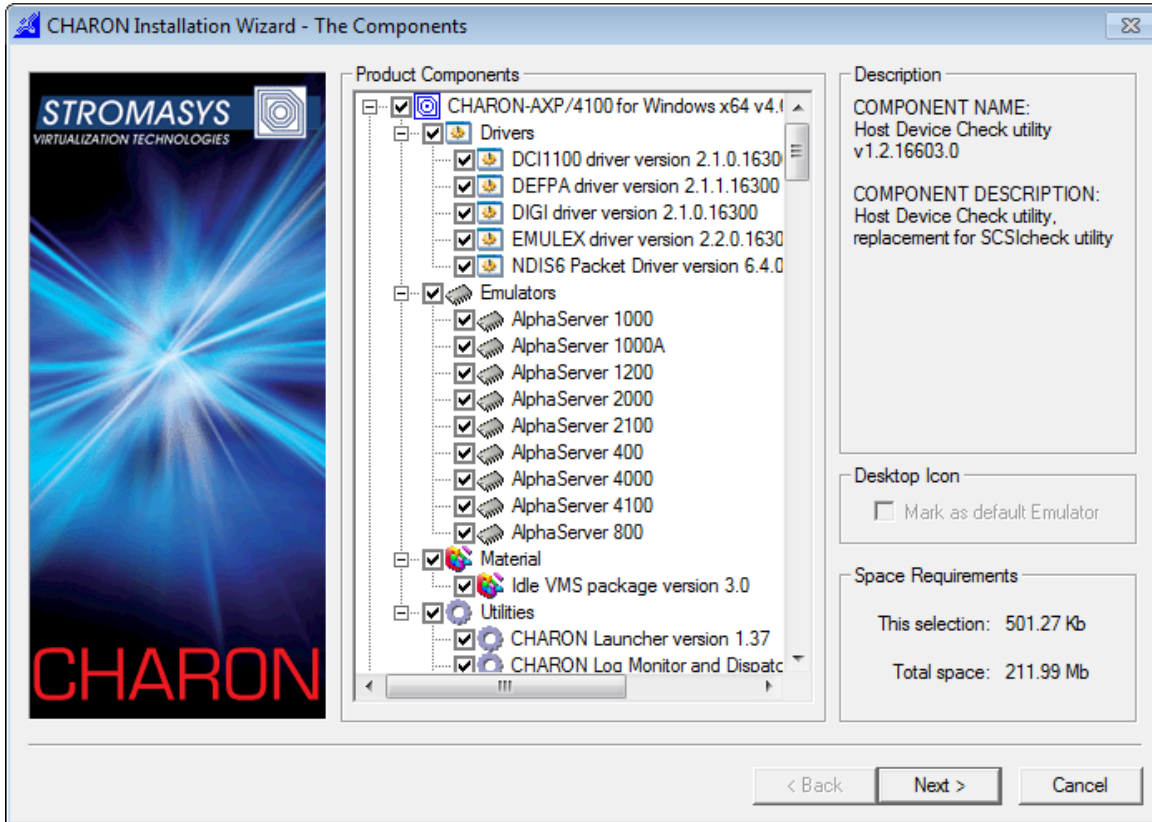



Select a directory to be used for the CHARON installation or use the default. The path can either be entered manually or selected using the "Browse" button. Once done, press the "Next" button.

The CHARON installation procedure will install the software packages required for CHARON-AXP, for example, Microsoft Visual C++ redistributive software and Sentinel HASP Run-time:

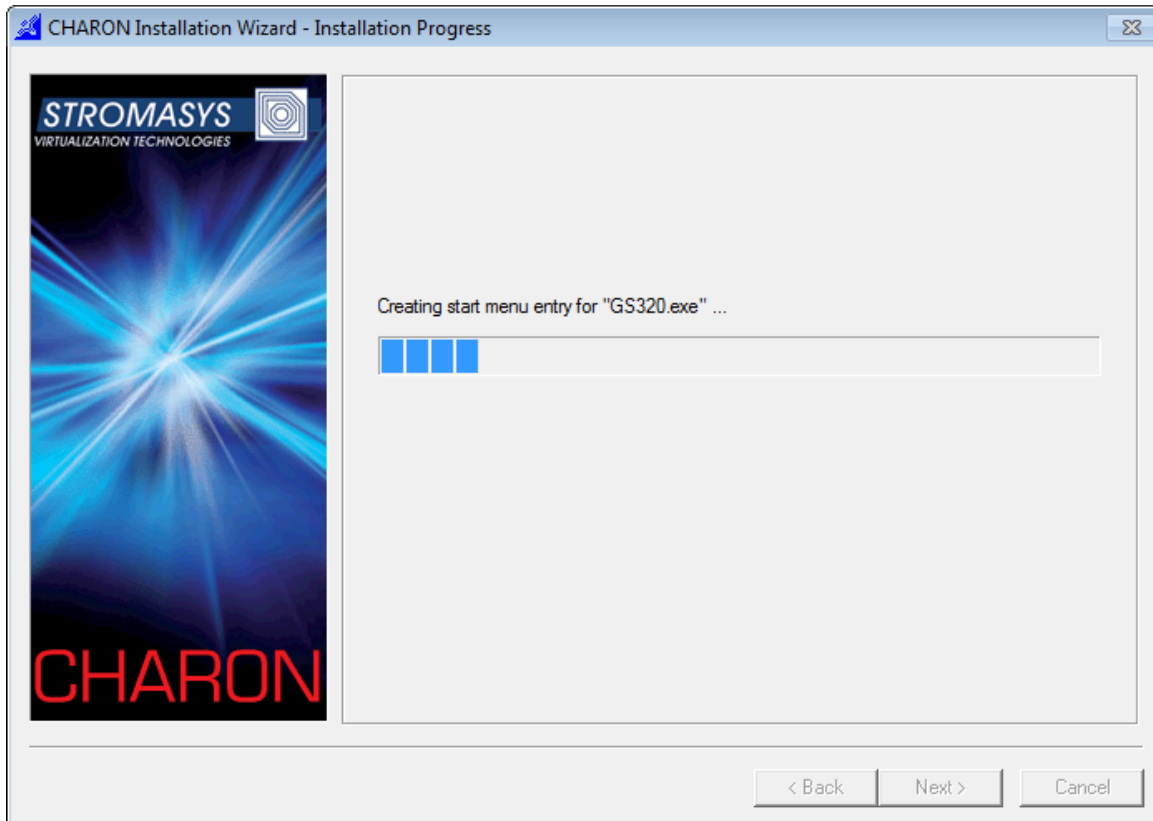


Once the packages are installed, the setup procedure will display a dialog for the chosen CHARON product(s) components:



 It is strongly recommended to leave the selection as it is unless you clearly understand what you're doing if you uncheck some modules.

Once done, press the "Next" button to start the installation of the selected components.

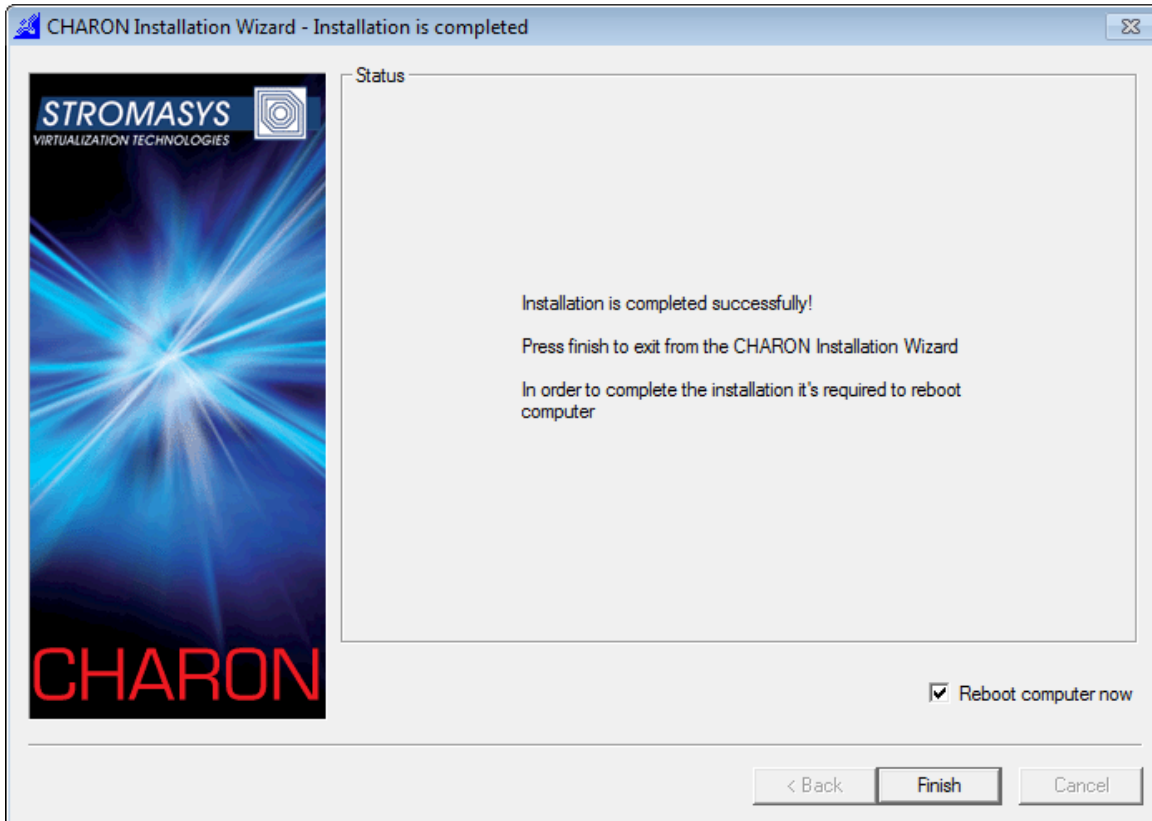




During the CHARON installation procedure, you will have to confirm the installation of the CHARON-specific network driver. Press the "Install" button:

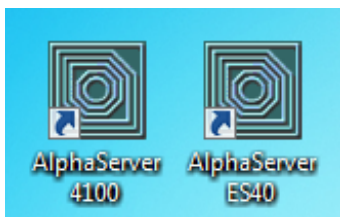


Once the installation is done, the following dialog will be displayed:



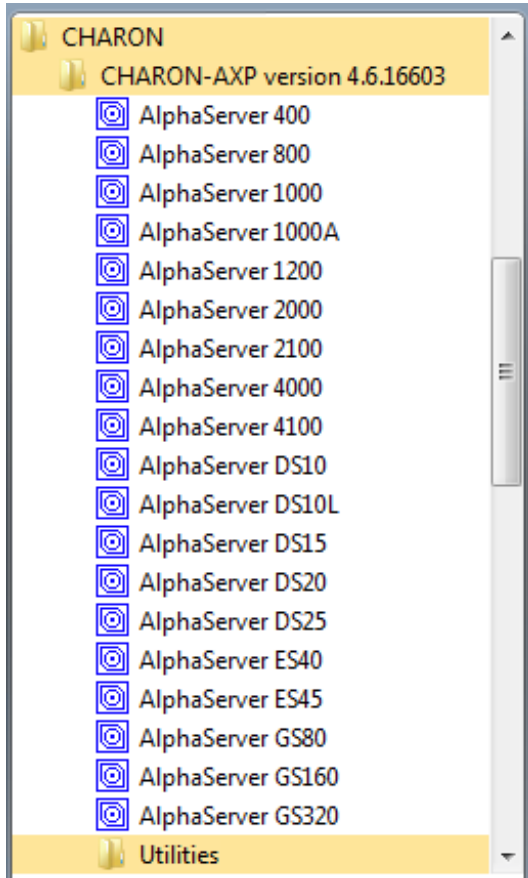
**!** It is strongly recommended to reboot the CHARON host system immediately: leave the "Reboot computer now" box checked and press the "Finish" button.

Once the host system is up again, you will notice new shortcuts that were created, on the desktop, during the installation procedure. For example:

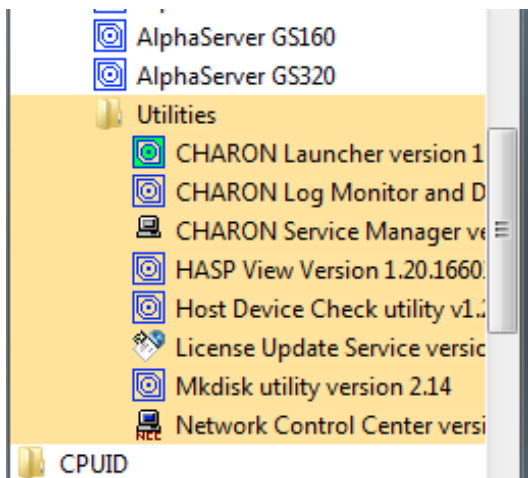


Once the CHARON-AXP license is installed normally (see below) and the configuration file has been modified to map devices, clicking a shortcut will start the corresponding model of the HP Alpha emulator.

The CHARON installation procedure has created the following structure under the "Start" menu:



The programs in the "Utilities" subdirectory are discussed in detail, below.



[Back to Table of Contents](#)

## CHARON-AXP home directory

By default CHARON-AXP is installed in the "C:\Program Files\CHARON" directory (it may also use "C:\Program Files (x86)\CHARON" directory if CHARON-AXP products have been installed on the host prior to CHARON-AXP). It has the following subdirectories:

Directory	Description
Auxiliary	Contains auxiliary material
Build_XXX\x64	Contains product executables, libraries and templates of the configuration files
Drivers	Contains the CHARON drivers
HASP_X.XX	Contains the Sentinel HASP Run-time installation (already installed)
InstallShell	Contains the CHARON installation procedure executables
Logs	Contains the CHARON installation log
Redistributables	Contains the redistributables needed for CHARON running (already installed)
Utilities_X.X.XXXXX\x64	Contains the CHARON utilities designed for Windows x64
Utilities_X.X.XXXXX\x86	Contains the CHARON utilities initially designed for Windows x86
Virtual Disk Images	Contains the CHARON virtual disk images, for example "idle_vms_pkg_v3.0"

The "Build\_XXX\x64" directory contains template configuration files with examples of typical configuration parameters and commentaries. The template files are used to build the emulator instance. We will pay our attention to this subject in the next chapter.

[Back to Table of Contents](#)

## License installation

### Regular HASP USB dongle

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

If the CHARON host is accessed remotely, please note that the contents of a regular HASP license cannot be displayed using RDP. ILO or iDRAC, some other console-capable program must be used.

Also, CHARON cannot be started manually from an RDP connection. In this case, a workaround is possible to install CHARON as a 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 to either connect it to the host USB port (to use it locally providing it to other hosts on local network at the same time) or to install it on a local network "server" for remote access from this particular host.

In case of remote usage:

- Copy "hasp\_install\Sentinel\_LDK\_Run-time\_cmd\_line\_X.XX.zip" file from CHARON distributive to the server in any directory, for example "C:\Temp"
- Extract the contents of this archive to the same directory.
- Login as "*Administrator*" on the server and open "cmd.exe" from the "Start" menu.
- Switch to that directory
- Install the extracted file:

**Example:**

```
...> cd c:\temp
c:\temp> haspdinst.exe -fr -kp -nomsg
c:\temp> haspdinst.exe -install -cm
```

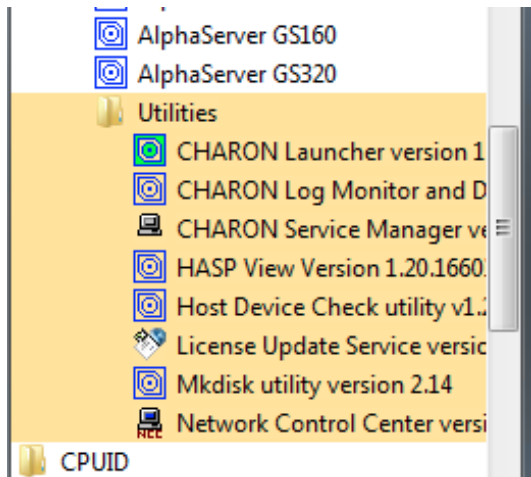
- Connect the network HASP dongle to the server USB port.

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

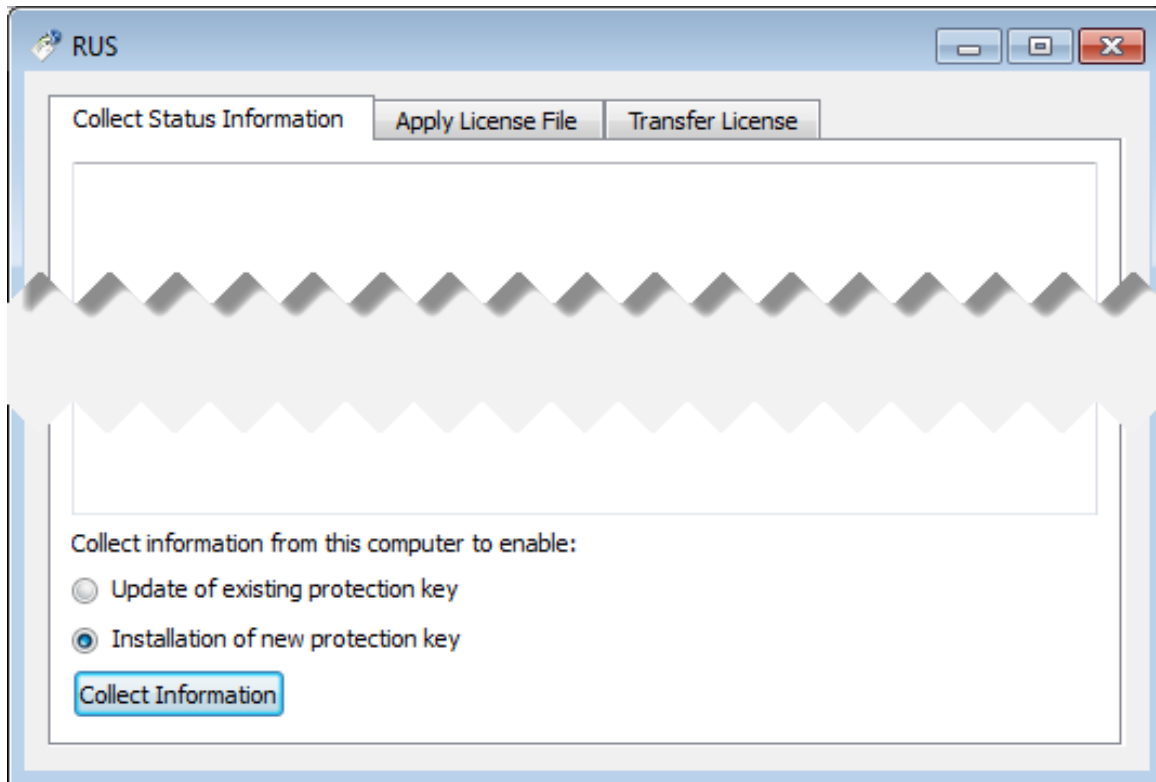
## Software license

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

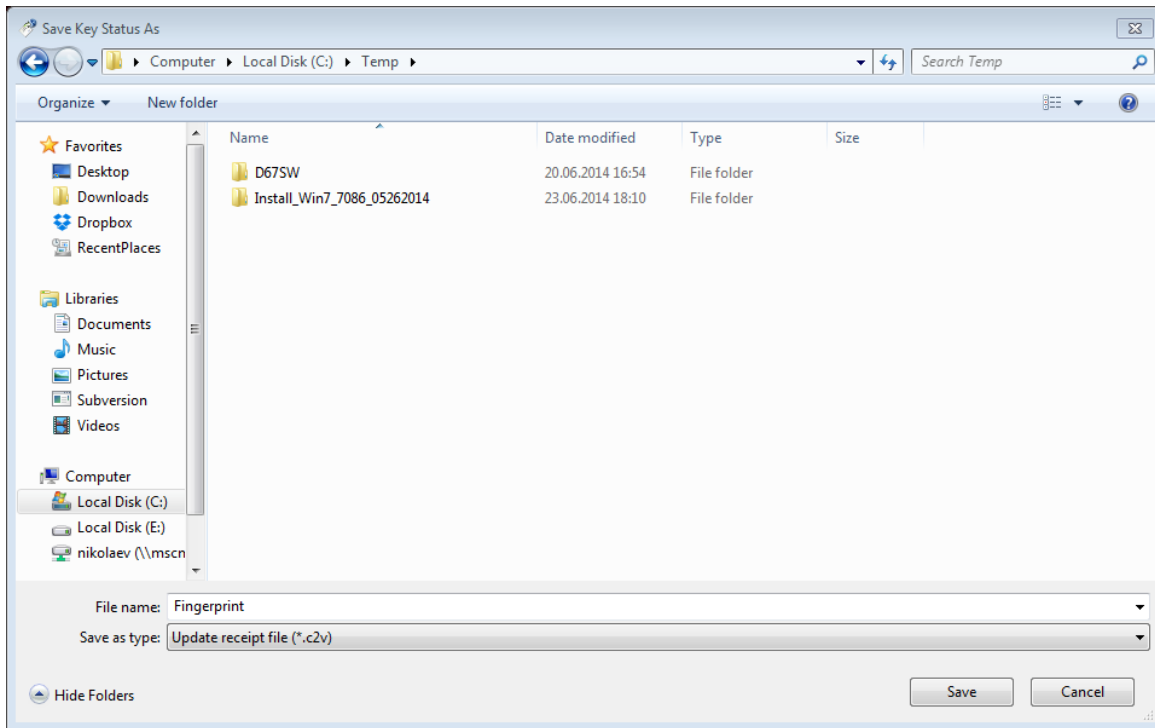
1. Press the Start button, select "All programs", "Charon" then the Charon product/version folder, "Utilities" and execute the "License Update Service" program:



The following default dialog window will appear:

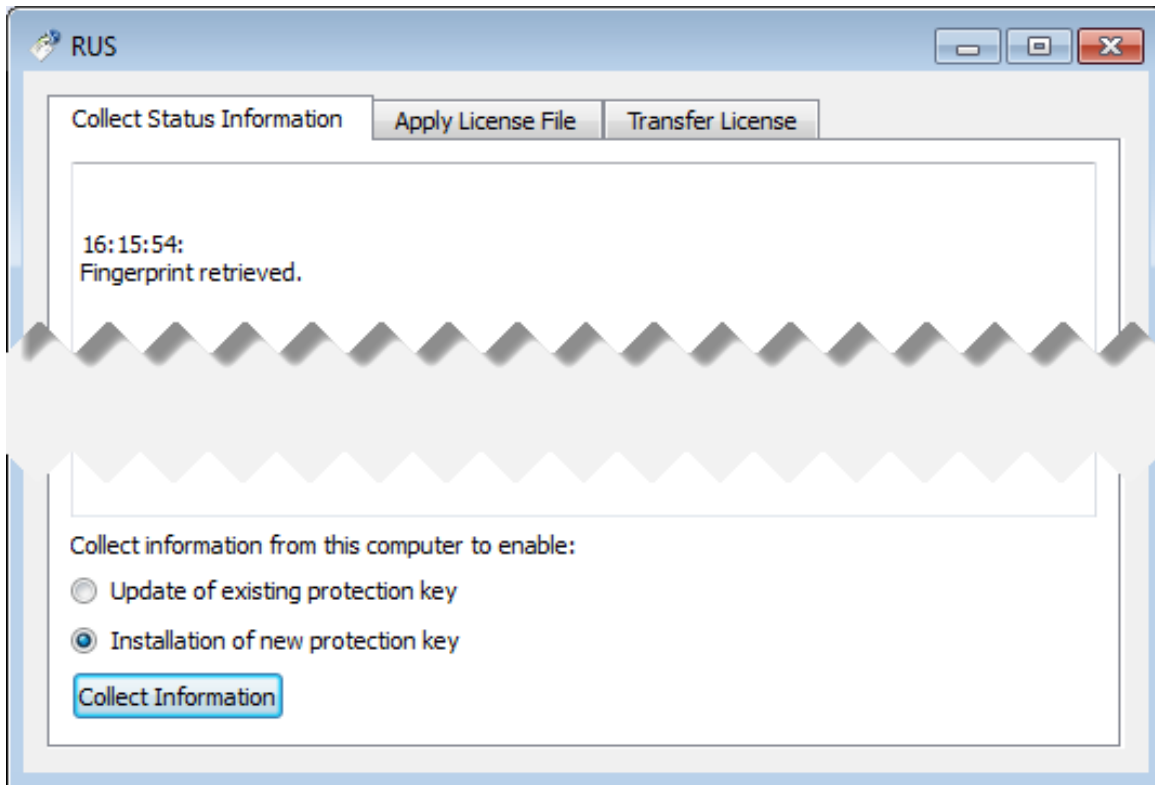


2. Select the "Installation of new protection key" radio-box and press the "Collect Information" button. The following window will appear:

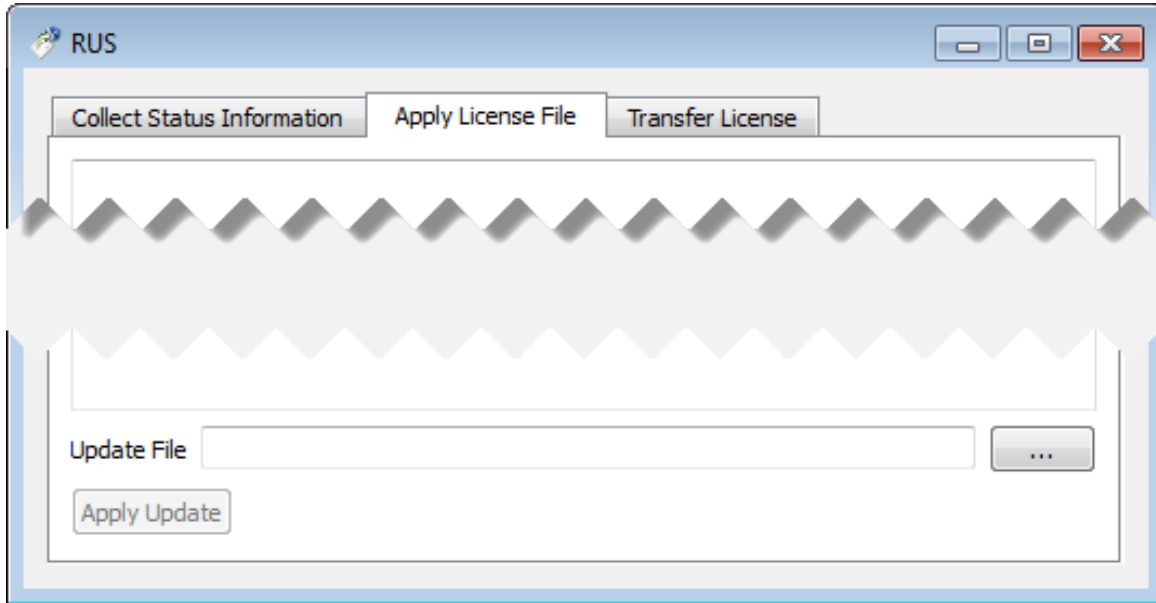


Specify the directory and name of the system "fingerprint" file to be created; press "Save".

3. The "License Update Service" will save the fingerprint file under the given name and report the status:

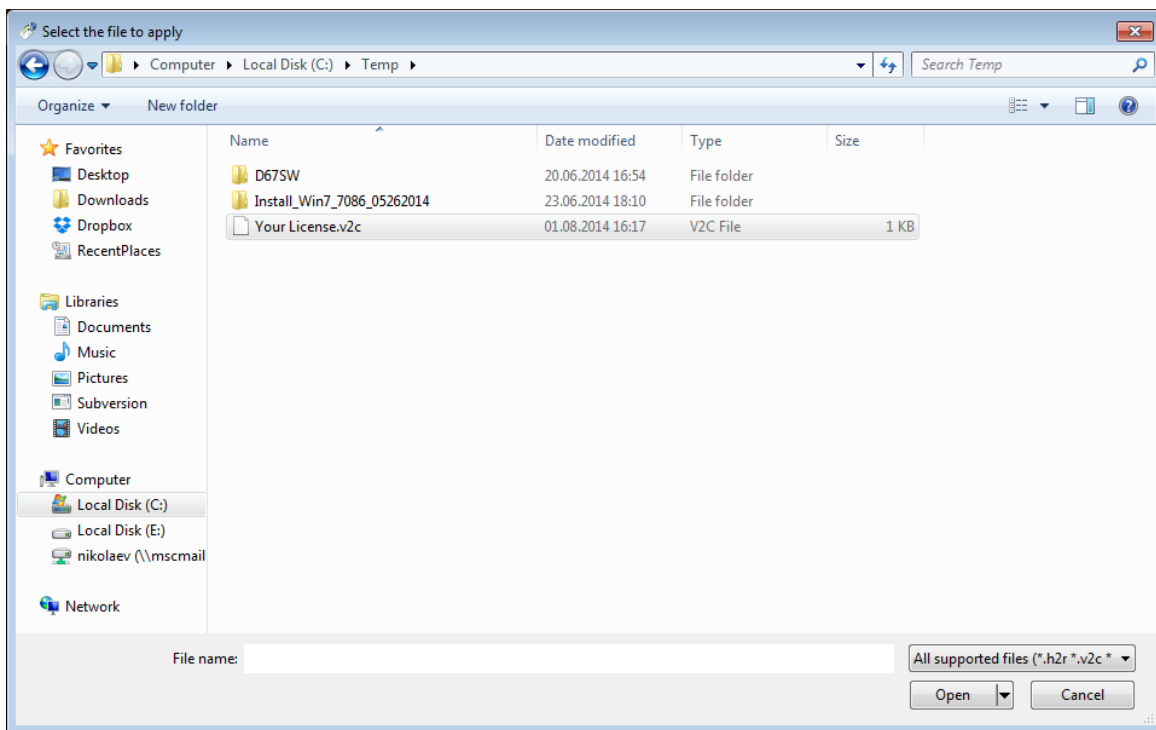


4. Send the resulting file to STROMASYS. In return STROMASYS will provide you with a "\*.v2c" file, for example "Your License.v2c"
5. Copy the received v2c file to the CHARON host in a folder of your choice and select the "Apply License File" tab:



6. Press the "..." button and browse for the received v2c file.

**Example:**



Press "Open" to apply the license.

7. Invoke the system default web browser and enter URL <http://localhost:1947> to display "**Sentinel Admin Control Center**" (ACC) web interface. This interface allows you to view and manage CHARON licenses.
8. Ensure that the software license appears now in the "**Sentinel Keys**" menu of the ACC.

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

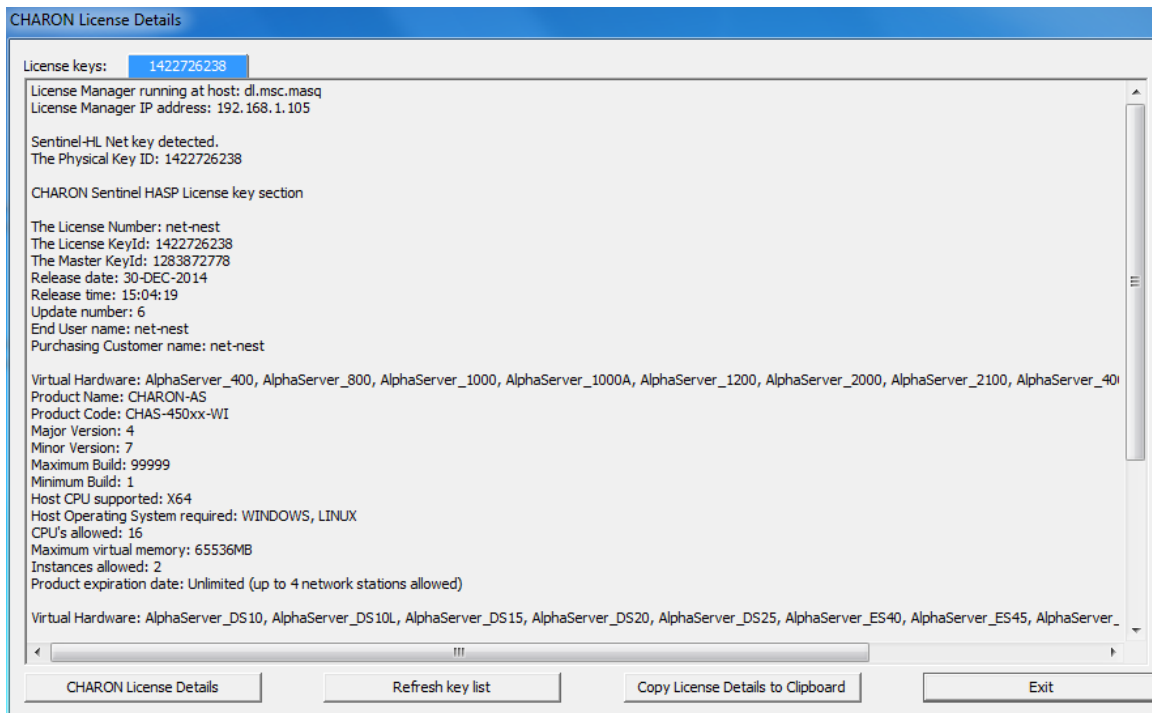
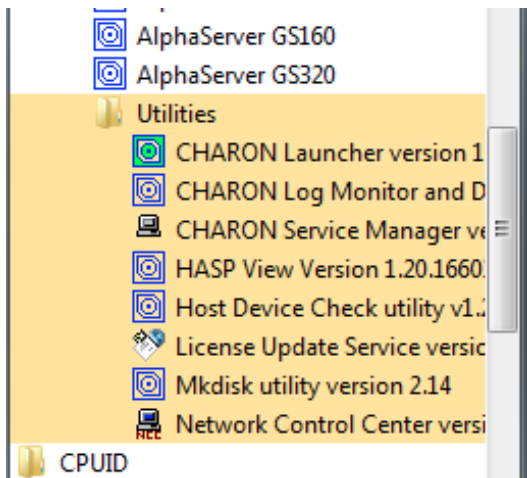
A so-called "Provisional" (demo) license does not require a collected fingerprint. For its installation, proceed directly with action (5) of the sequence above

Some CHARON-AXP distributives may be equipped with a "driverless" Sentinel HL dongle (both regular and network). In this situation, perform the CHARON-AXP installation normally and then manually remove the installed Sentinel drivers from the host to start using that specific key. This operation is described in details in the [CHARON-AXP Licensing](#) chapter of this Guide.

[Back to Table of Contents](#)

## License validity verification

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



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

## Troubleshooting

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

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

If no license is displayed make sure that all the recommendations above about the remote access to the host are fulfilled (no remote access is used), that the HASP USB key is not broken and its LED indicator is lit (meaning that it is used by the host).

If only one License key / SL is seen and its content is incorrect, please contact STROMASYS as soon as possible.

If several License keys / SLs are displayed remove all of them and leave only the one provided by STROMASYS for the just installed version of CHARON. Moving 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 a local network (or by disabling remote access to network licenses from CHARON host - see detailed explanations below).

Software licenses can also be uninstalled using the "License Update Service" utility (see above) in the following way:

1. Open up the "Transfer License" tab of the utility.
2. Press the "..." button beside the "Save recipient information to" message and save the system fingerprint to a directory.
3. Select the software license you are going to delete.
4. Press the "..." button belonging to the "Read the recipient information file from" message and specify the fingerprint file you have just collected.
5. Press the "..." button belonging to the "Generate the license transfer file to" message and enter a name for the license transfer file.
6. Press "Generate License Transfer File"
7. Now the chosen software license is removed from the CHARON host

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

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

**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 not accessible:

```
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 is not accessible, CHARON tries to access the backup license.

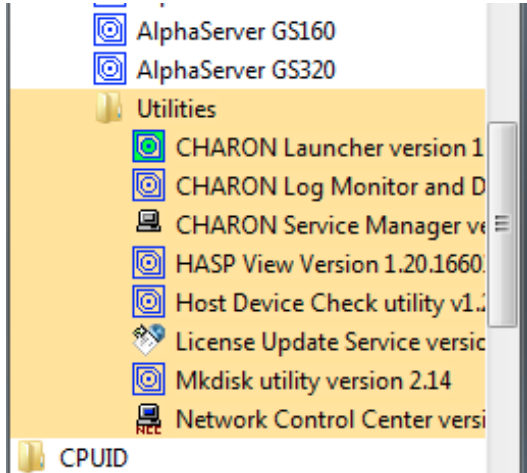
[Back to Table of Contents](#)



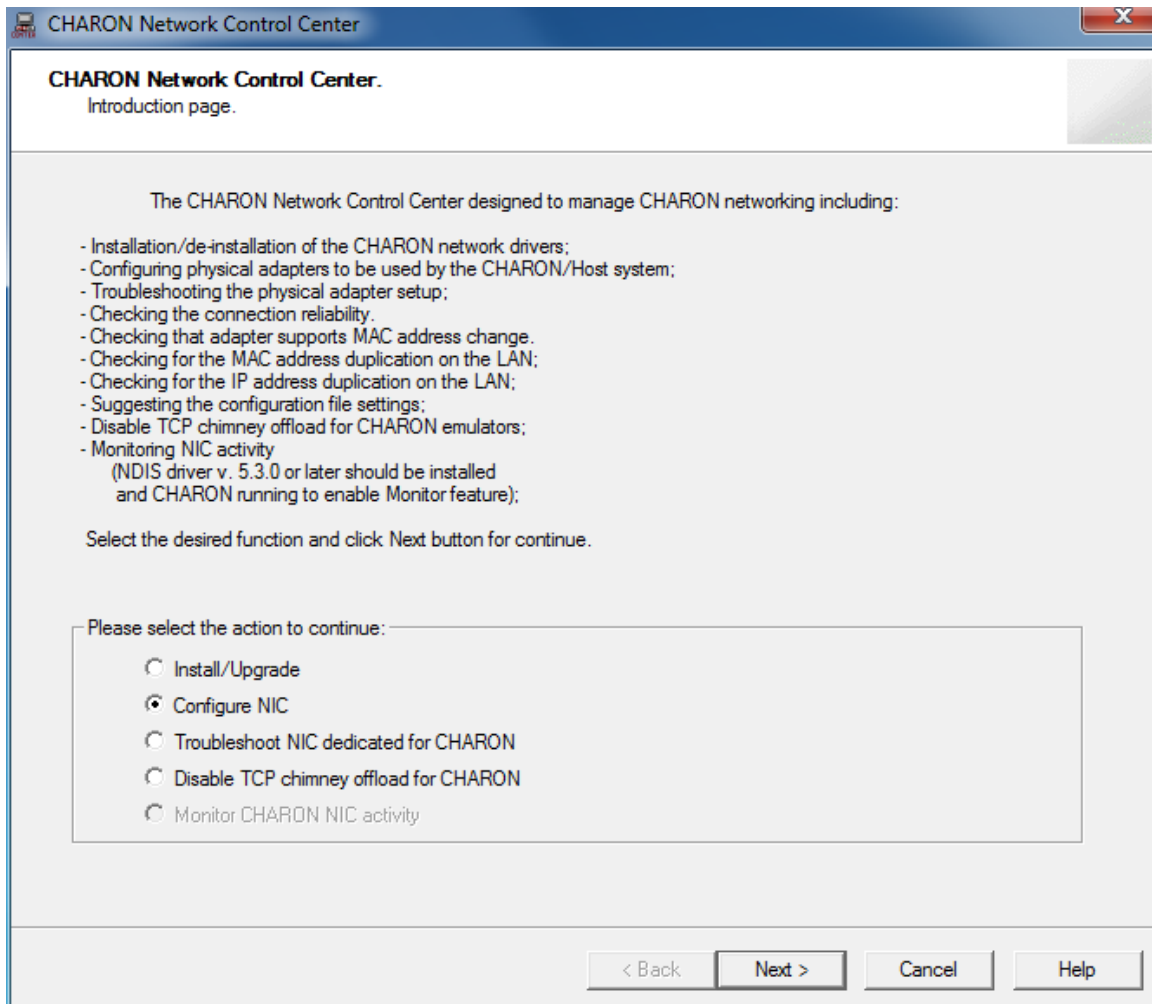
## Network configuration

In most cases, CHARON will use a network. If so, CHARON requires a dedicated network interface cleared from any other protocols including TCP/IP.

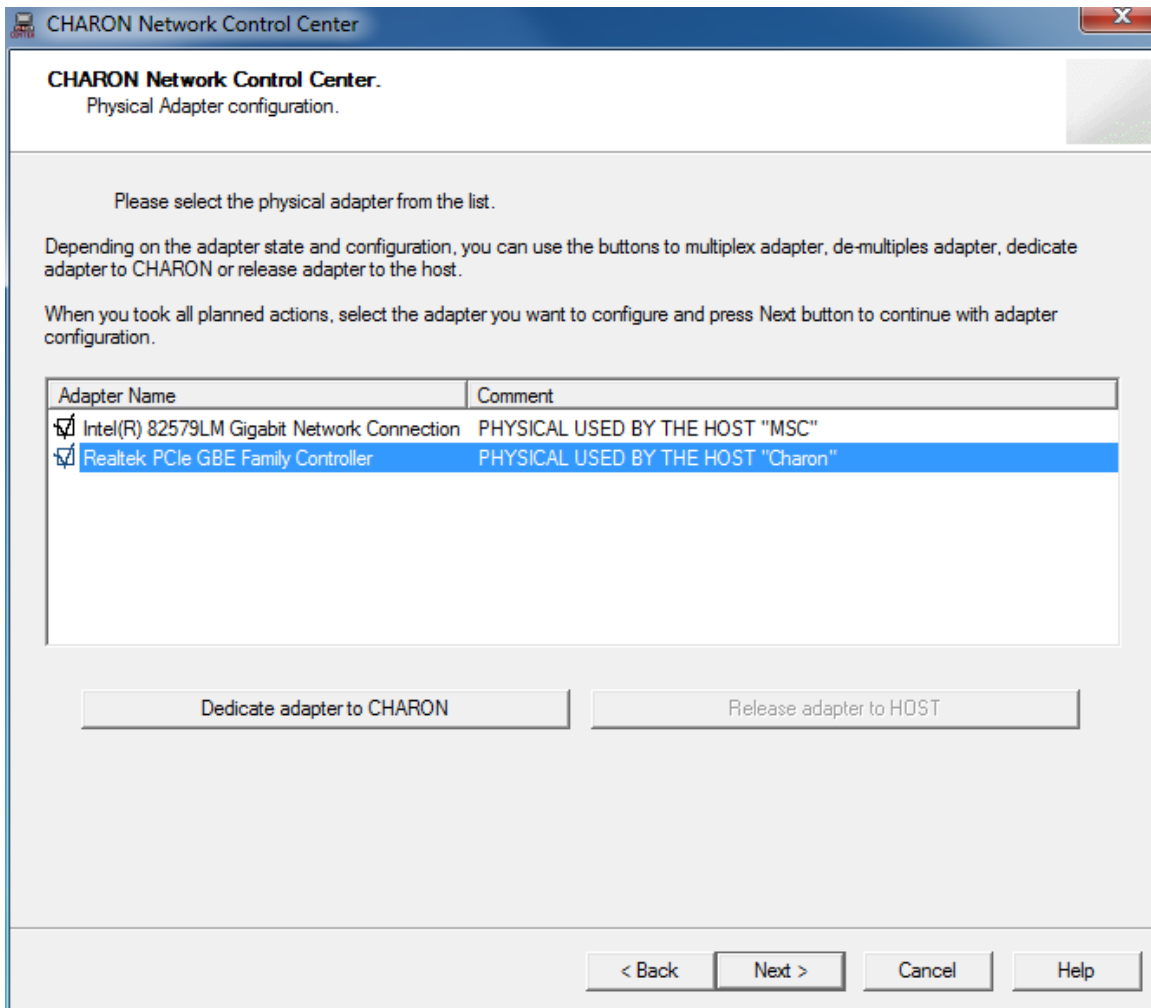
Network configuration is done with the help of the "Network Control Center" (NCC) utility:



Start the "Network Control Center" utility and select "Configure NIC" in the first dialog window:



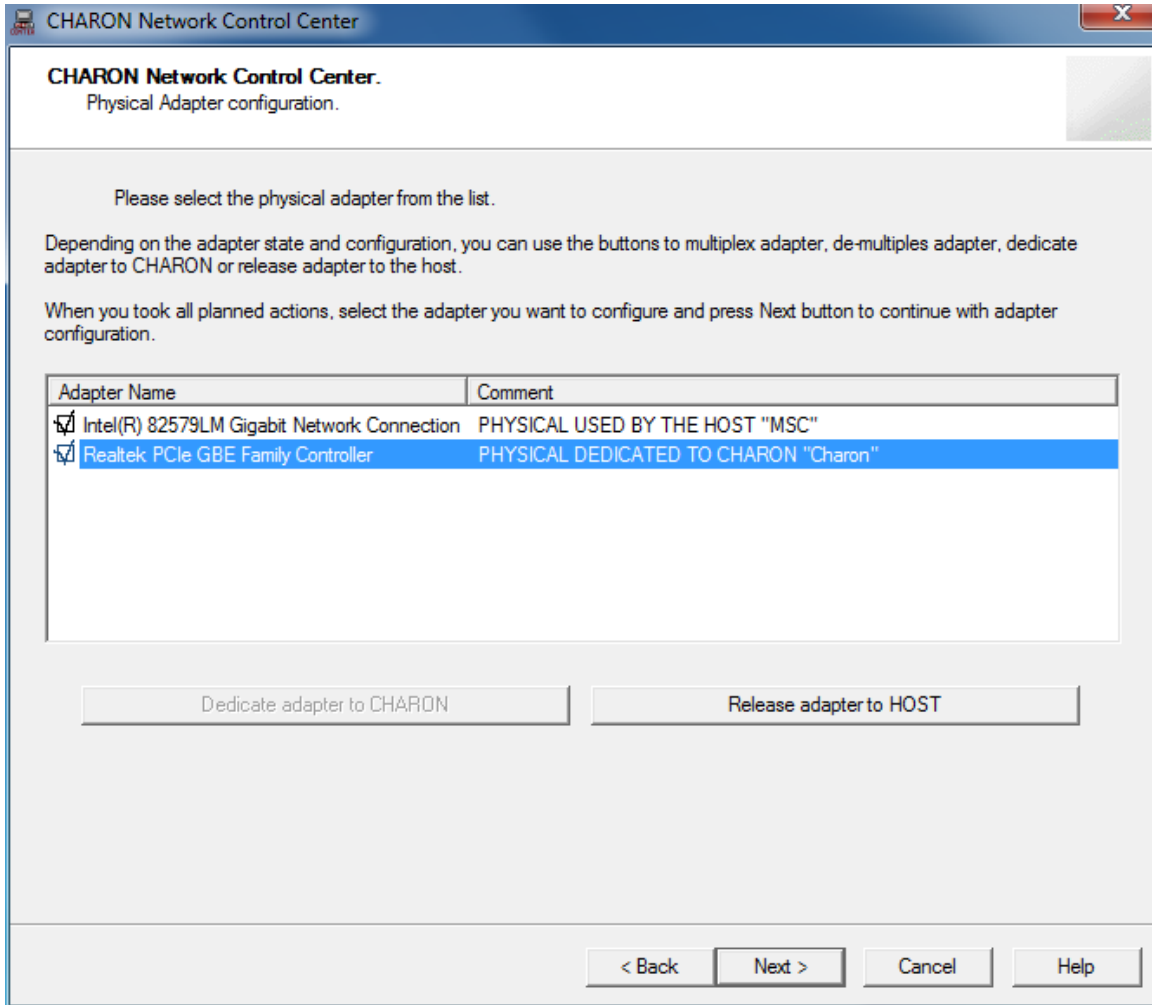
Press the "Next" button, the following dialog will appear:



Select the interface to be dedicated to CHARON (in our example it is Realtek PCIe GBE Family Controller with the name "Charon") and press the button "Dedicate adapter to CHARON".

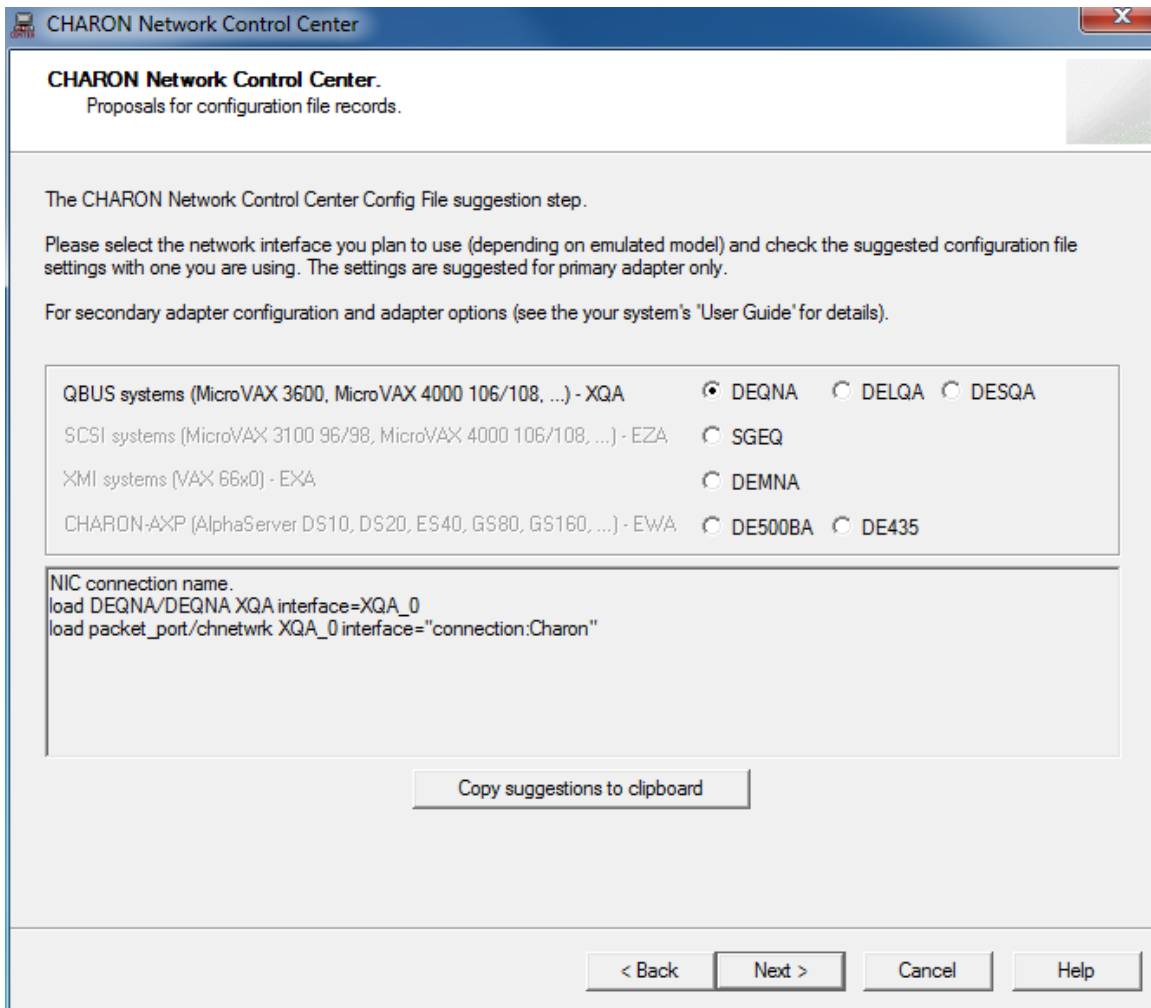
CHARON supports VLAN adapters. If you are going to use them, proceed with their installation and configuration according to the network adapter vendor User's Guide and then select the VLAN interface in the dialog shown above, the same as you do for a regular network interface. There is no difference in configuration between a regular network adapter and a VLAN one, so all the provided instructions are fully applicable for VLAN adapters as well.

After a few seconds the chosen interface will be assigned to CHARON:



Note that it is possible to release the interface back to the CHARON host. To do that, select the target interface and press the "Release adapter to HOST" button.

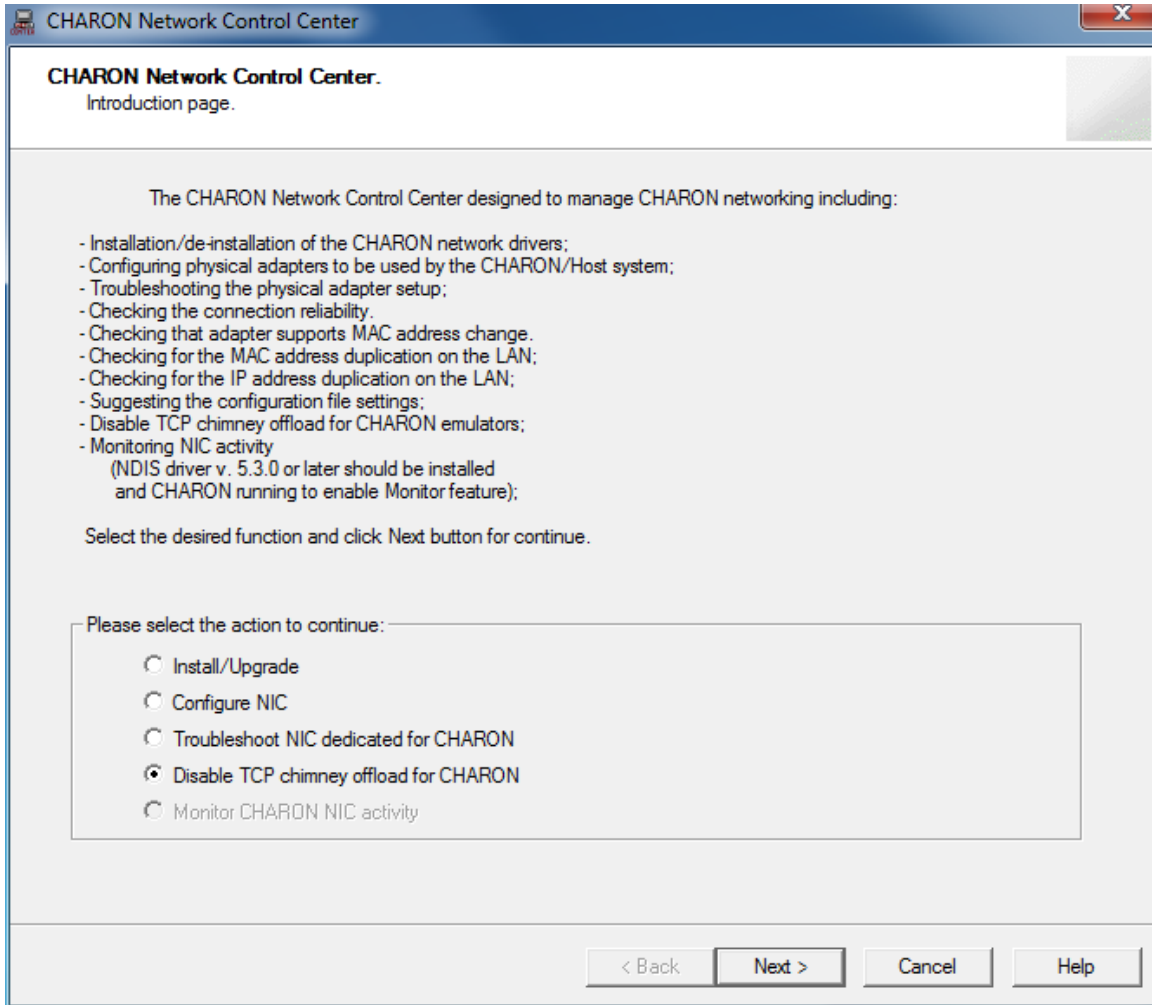
The next step displays text to be included in the CHARON configuration file for the interface dedicated to CHARON. Select the target interface and press the "Next" button. The following dialog will appear:



Selecting the target emulated network interface displays the configuration lines specific for the selected CHARON host network interface. Press the "Copy suggestions to clipboard" button to copy the suggested configuration lines. They can be pasted into the CHARON-AXP configuration file during the next editing session by pressing "Ctrl-V".


Press the "Cancel" button to exit from the "Network Control Center" utility.

Restart the "Network Control Center" utility again and select "Disable TCP chimney offload for CHARON":



Press the "Next" button to apply the "offload" parameters settings.

Press the "Cancel" button to exit from the "Network Control Center" utility.

 When the NIC properties are configured in Windows, a reboot of the CHARON host is required for the changes, specifically "offload" parameters, to take effect.

[Back to Table of Contents](#)

∨ [More information and details on disabling TCP chimney offload...](#)

## For Windows 2008 R2, Windows 2012 R2, Windows 7, 8 and 8.1

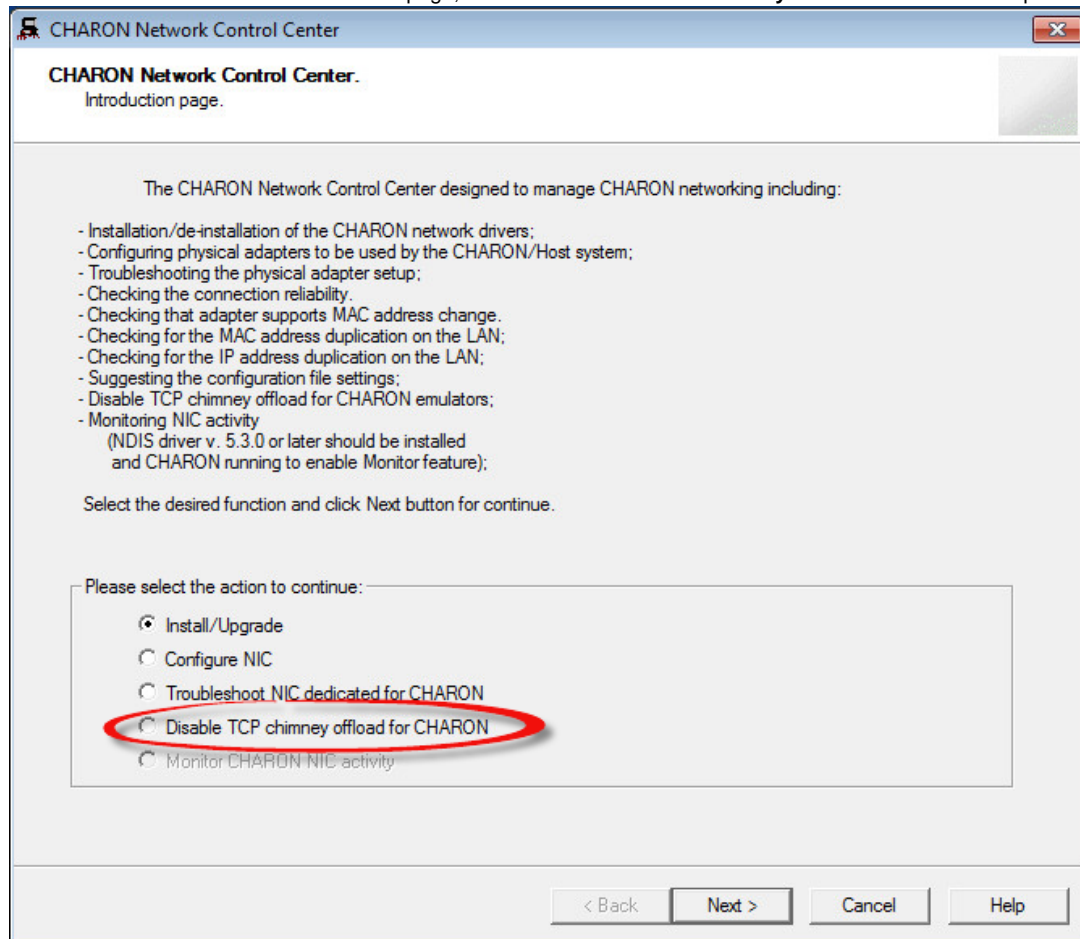
TCP Chimney offload must be disabled.

**i** TCP Chimney Offload is a networking technology that helps transfer the workload from the CPU to a network adapter during network data transfer.

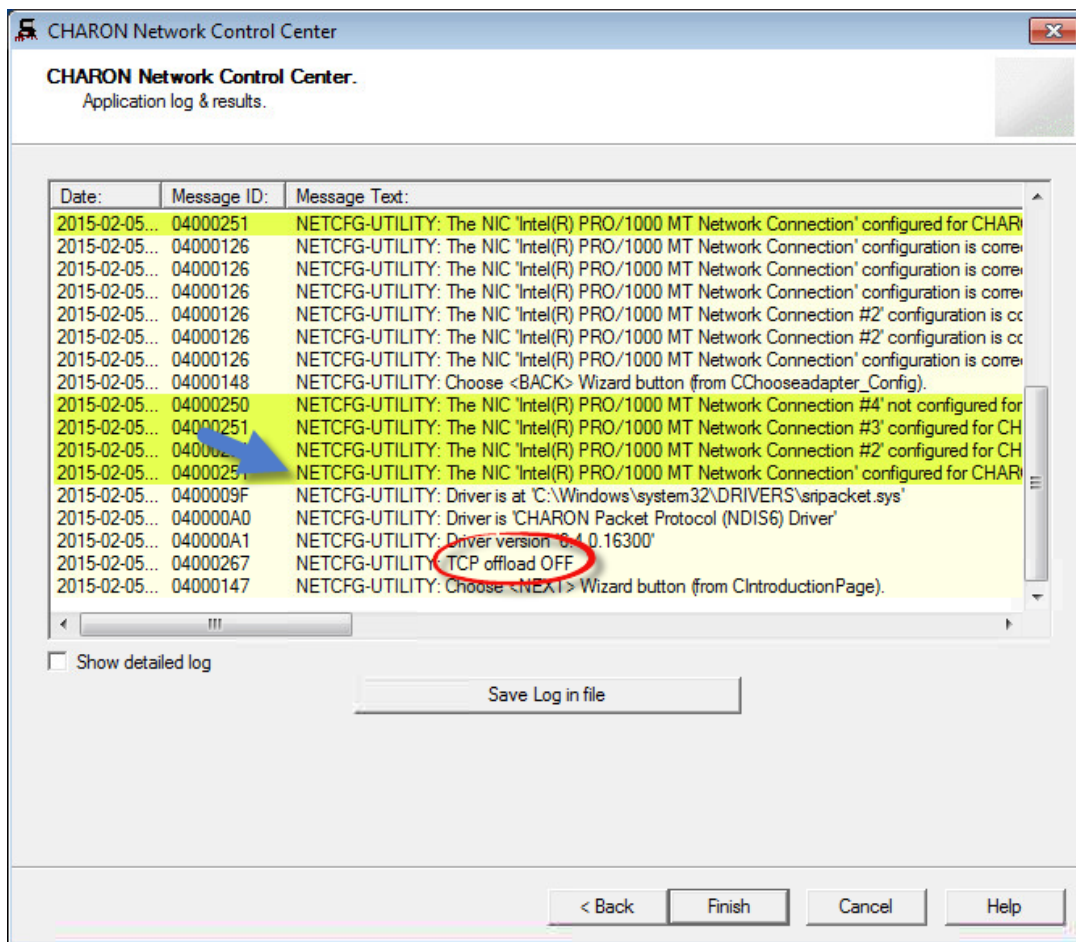
### Warning

The commands to enable and disable TCP Chimney Offload for specific applications and ports require that the Windows Firewall service and Base Filtering Engine (BFE) services are running. Before using these commands or the Network Control Center Utility, ensure that the Windows Firewall service and BFE service are running.

- Starting with CHARON-VAX V4.6 and CHARON-AXP V4.6, the Network Control Center Utility allows you to disable TCP Chimney offload:
  - Open the **Start** menu, select "**All Programs**" then "**CHARON**", select your Charon product version, "**Utilities**" then "**Network Control Center version x.xx**".
  - From the "**Network Control Center**" main page, select the "**Disable TCP chimney offload for CHARON**" option:



- Click on the "**Next**" button.
- The TCP Chimney offloading will be disabled for all CHARON emulators (.exe).



- Click on the "Finish" button to exit.

**Warning**

- Running guests will have to be powered off/on for the new settings to be taken into account.
- This operation will have to be performed after each upgrade and patch installation.

■ **For older CHARON products versions**

- Open a command line windows with Admin rights
- Locate your current version of CHARON and emulator used.
- Issue the following command for all emulators running on your server:


```
C:\> netsh interface tcp add chimneyapplication state=disabled application="<emulator full path>.exe"
```

- Example:** we assume here we are running CHARON-AXP Build 16400 and emulator executed is AlphaServer 1000

```
C:\> netsh interface tcp add chimneyapplication state=disabled application="C:\Program Files\CHARON\Build_16400\x64\as1000.exe"
```

**Warning**

- Running guests will have to be powered off/on for the new settings to be taken into account.
- This operation will have to be performed after each upgrade and patch installation.

 You can check TCP Chimney offload is disabled using the following command:

```
C:\Windows\system32>netsh int tcp show chimney
```

State	Application
disabled	C:\Program Files <x86>\CHARON\Build_14802\x64\as400.exe
disabled	C:\Program Files <x86>\CHARON\Build_16603\x64\as2100.exe
disabled	C:\Program Files <x86>\CHARON\Build_16601\x64\as4100.exe
disabled	C:\Program Files <x86>\CHARON\Build_16601\x64\ds10.exe
disabled	C:\Program Files <x86>\CHARON\Build_14404\x86\mv3k9.exe
disabled	C:\Program Files <x86>\CHARON\Build_14404\x86\vx3k6_128.exe
disabled	C:\Program Files <x86>\CHARON\Build_15804\x86\vx3k9_128.exe
disabled	C:\Program Files <x86>\CHARON\Build_16603\x64\as1200.exe
disabled	C:\Program Files <x86>\CHARON\Build_15308\x64\as4000.exe
disabled	C:\Program Files <x86>\CHARON\Build_14802\x64\ES40.exe

●●●

## For older versions of Windows

Intelligent packet processing should be switched off for the network adapters dedicated to CHARON-AXP (naming of the parameters depends on the network adapter driver):

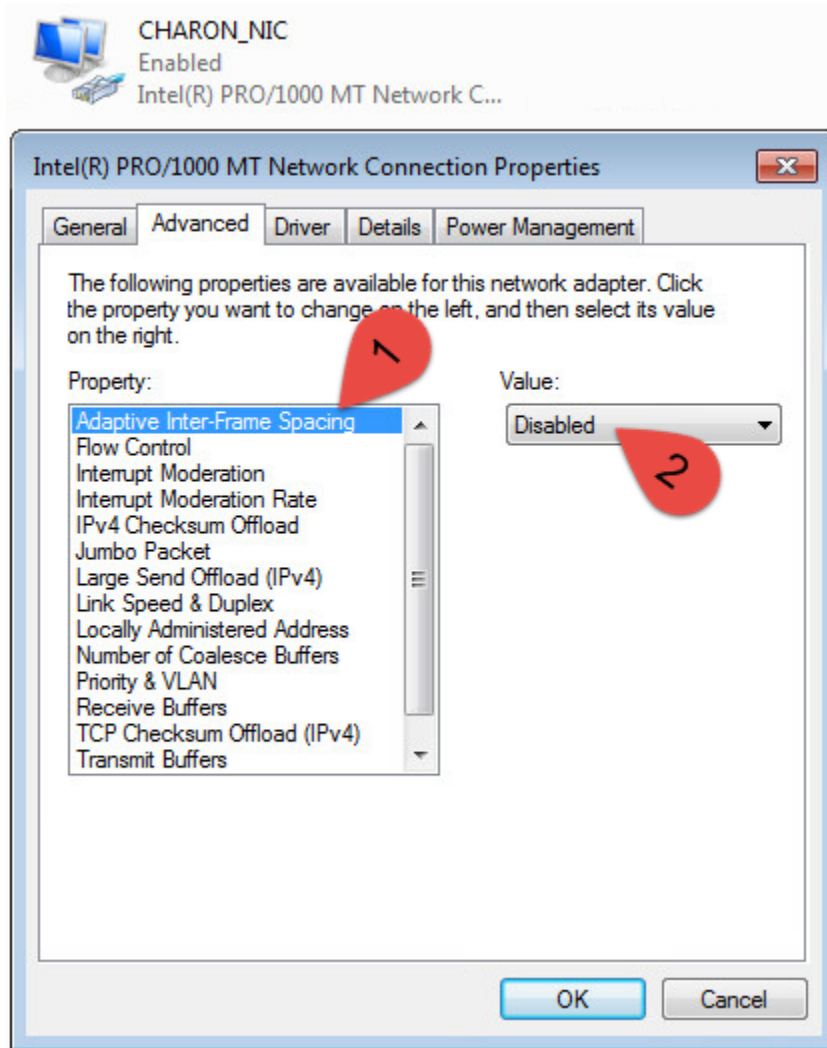
- Adaptive interframe spacing
- Flow control
- Interrupt moderation
- Interrupt moderation rate
- Ipv4 checksum offload
- Jumbo packet
- Large send offload
- Disable Priority and VLAN
- TCP checksum offload
- UDP checksum offload

To do so, open the "Control Panel", select "Network and Internet" and "Network Connections". Right click on the adapter you dedicated to CHARON, select "Properties" and click on the "Configure" button.

Select the "Advanced" tab and disable the properties mentioned in the list above.



Example:



Once all parameters have been turned off, click on the "OK" button.

**Warning**

Running guests will have to be powered off/on for the new settings to be taken into account

## After installation

If you plan to use local system administrator ("Administrator") or the CHARON-AXP user having administrative privileges no other actions are required.

If the CHARON-AXP user belongs to some domain you have to add this user to the CHARON-GRP group as it is described in [this article](#) and then reboot CHARON host.

Otherwise it is possible to use standard account (both local and domain) for running CHARON-AXP:

1. Login as the local system administrator ("Administrator") on the host system.
2. Create a special user for running CHARON-AXP. This user must have standard privileges. Please consult with your Windows User's Guide on details.
3. Add this user to the CHARON-GRP group as it is described in [this article](#) and then reboot CHARON host.
4. Login as the created user.

[Back to Table of Contents](#)

# Running CHARON-AXP for Windows

## Table of Contents

- Introduction
- Running from a Desktop shortcut
- Running from the Launcher utility
- Running as system service
  - Installation and start of CHARON-AXP service
  - Management of CHARON-AXP service
  - Stopping CHARON-AXP service
  - Modification of CHARON-AXP service
  - Removing CHARON-AXP service

## Introduction

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

In case of multiple instances, please use only absolute paths and unique names to all the files referenced in the configuration file of each CHARON-AXP instance (log, toy clock, rom files and all the other data such as disk images - all these objects to be discussed later in this document). Also, hardware devices (e.g., CD-ROM) may be used by only one instance at a time (not shared).

For example:

```
...
set session log="C:\Charon Instances\1st_es40.log"
set rom container="C:\Charon Instances\1st_es40.bin"
set toy container="C:\Charon Instances\1st_es40.dat"

load KZPBA PKA scsi_id = 7
set PKA container[0]="C:\Charon Instances\1st_es40_boot_disk.vdisk"
...
```

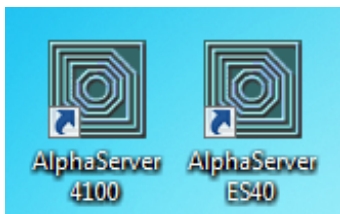
Please refer to the next chapters for more details concerning CHARON-AXP configuration details.

[Back to Table of Contents](#)

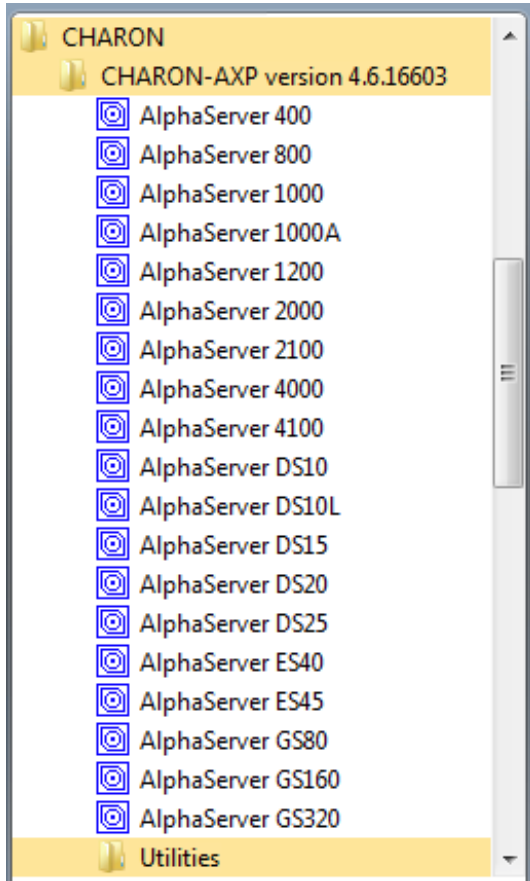
## Running from a Desktop shortcut

The CHARON installation procedure creates shortcuts on the desktop for so-called "default" HP Alpha models and shortcuts under "Start" menu for all HP Alpha models included during installation.

Desktop shortcuts:

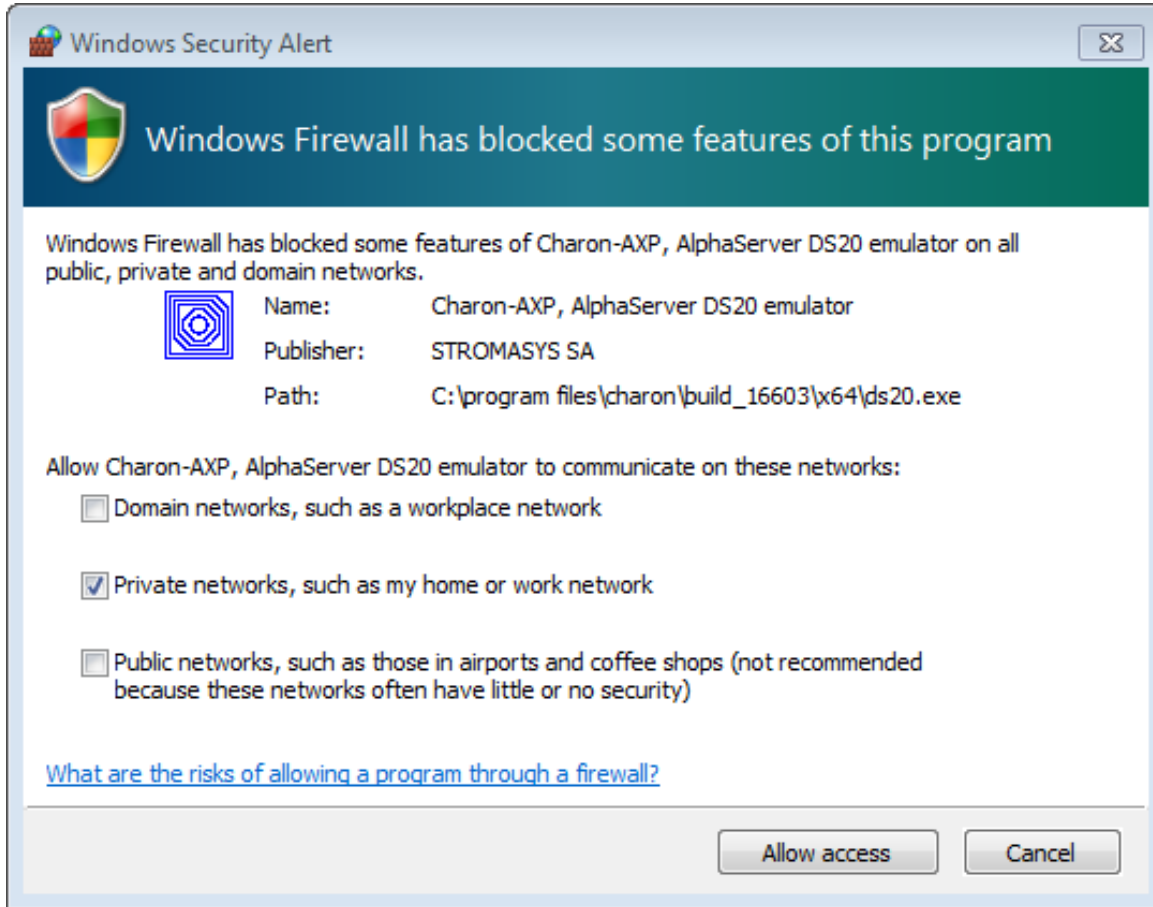


"Start" menu shortcuts:



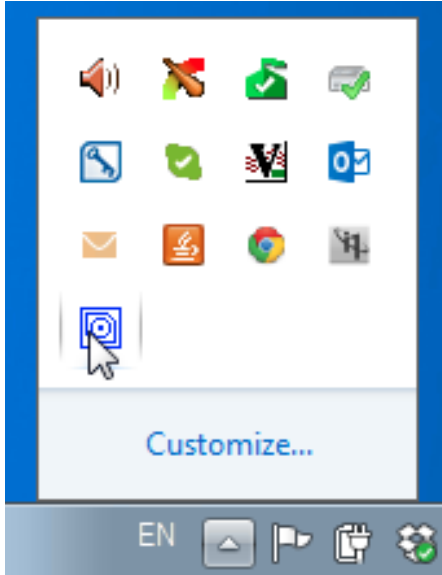
To run a certain CHARON-AXP model, double click the target model's desktop shortcut (or click on "Start" menu shortcut).

Depending on settings, Windows may display a Firewall warning dialog asking to confirm CHARON access to different networks.

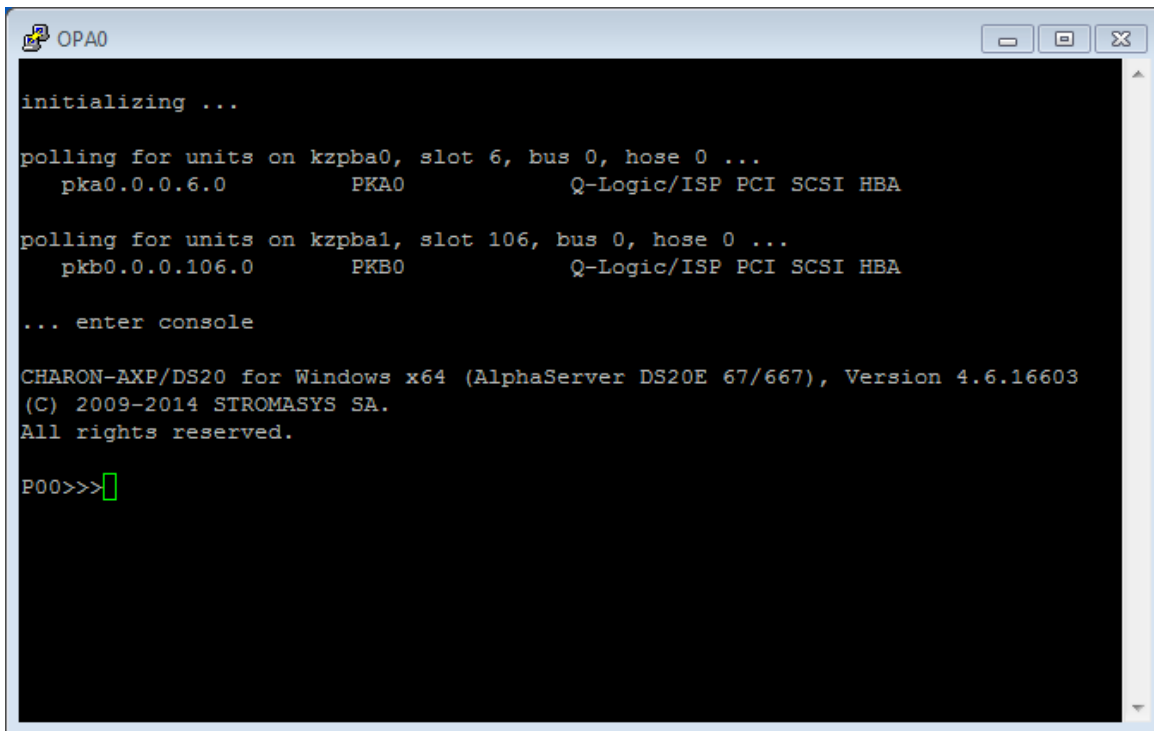


Check the desired options and press "Allow access" button.

The CHARON icon will appear in the system tray menu:



Next, the HP Alpha emulated console will appear:



If the CHARON icon in the system tray menu disappears, it means a critical error happened at run-time. In this case you have to analyze the CHARON log file, by default stored beside CHARON executables. The default location for log files is "C:\Program Files\CHARON\Build\_XXX\x64". The name of the log file depends on the model of emulated HP Alpha and the date and time, for example "AlphaServer\_1200-2015-01-20-13-40-32-000000000.log"

The easiest way to find the needed log file is to sort all the files in the CHARON executables directory by date of modification.

Generally it is not recommended to use template (sample) configuration files (used in shortcuts) for your configuration since those files are part of the installation and may be deleted by the CHARON installation procedure on modifications or re-installation. It is a good practice to copy the required configuration template from "C:\Program Files\CHARON\Build\_XXX\x64" directory to some local file and build the configuration from there.

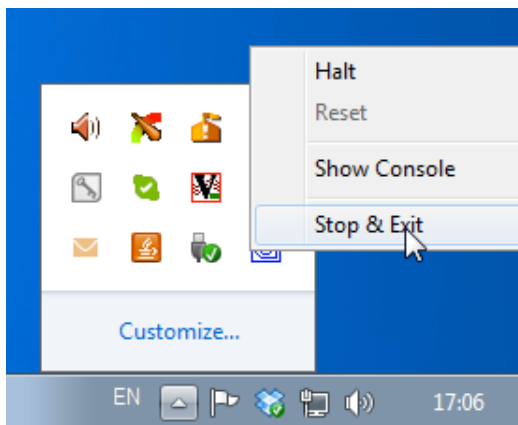
Once the sample configuration files are copied, CHARON default shortcuts can be changed to use them instead:

1. Right click on CHARON shortcut for target HP Alpha model
2. Choose "Properties"
3. In the "Target" edit box, change the default template configuration file path and name to the local one. For example "C:\My configurations\my\_es40.cfg"
4. Press "Apply" button

The next step is to install or copy a system disk for CHARON operation. This can be either installation of a new OpenVMS, Tru64 Unix, etc. system using a distributive provided by HP or a transfer of data from an existing HP Alpha system. These possibilities will be discussed in detail in the next chapters.

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

- **Configuration with no changes to the template configuration file**
  - Right click on CHARON icon in the system tray menu, choose "**Stop & Exit**":



- Type "**power off**" in CHARON-AXP SRM console
- **Configuration file updated to enable "F6" button in configuration file to trigger exit from CHARON:**

```
set OPA0 stop_on = F6
```

- To exit, press "**F6**" in the console window
- Use CHARON icon tray menu as shown above

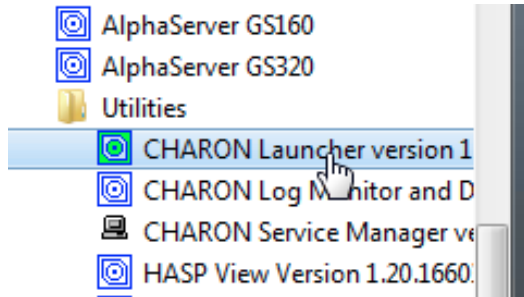
 Before stopping the CHARON-AXP service, shutdown the operating system running in CHARON-AXP.

[Back to Table of Contents](#)

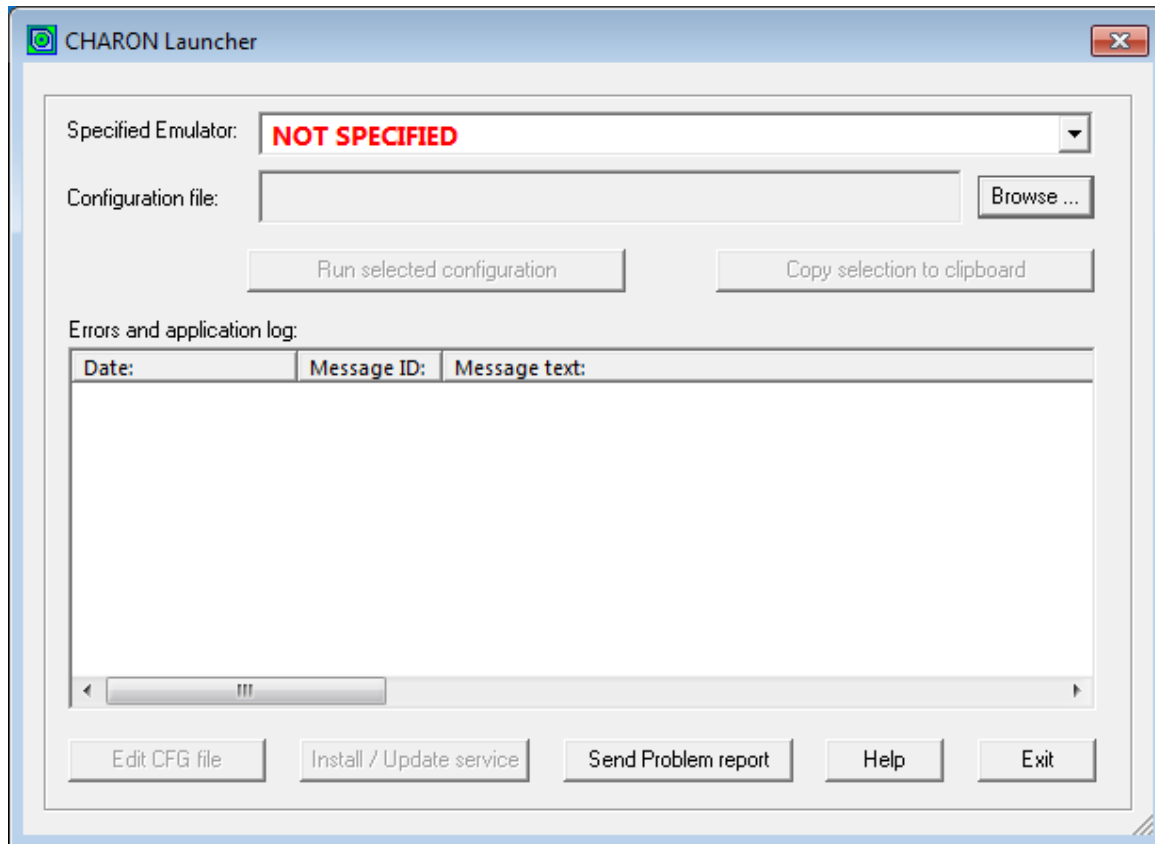
## Running from the Launcher utility

Running from the "CHARON Launcher" utility is very convenient during the debugging of your CHARON configuration. The utility shows CHARON log updates in real time. If RDP is used to connect to the CHARON host, the CHARON may not see the license; start CHARON as a Service instead. From the Launcher, CHARON runs as a User process which will terminate if the User logs off. The Service is a System process and will continue to run until stopped.

Start the Launcher:

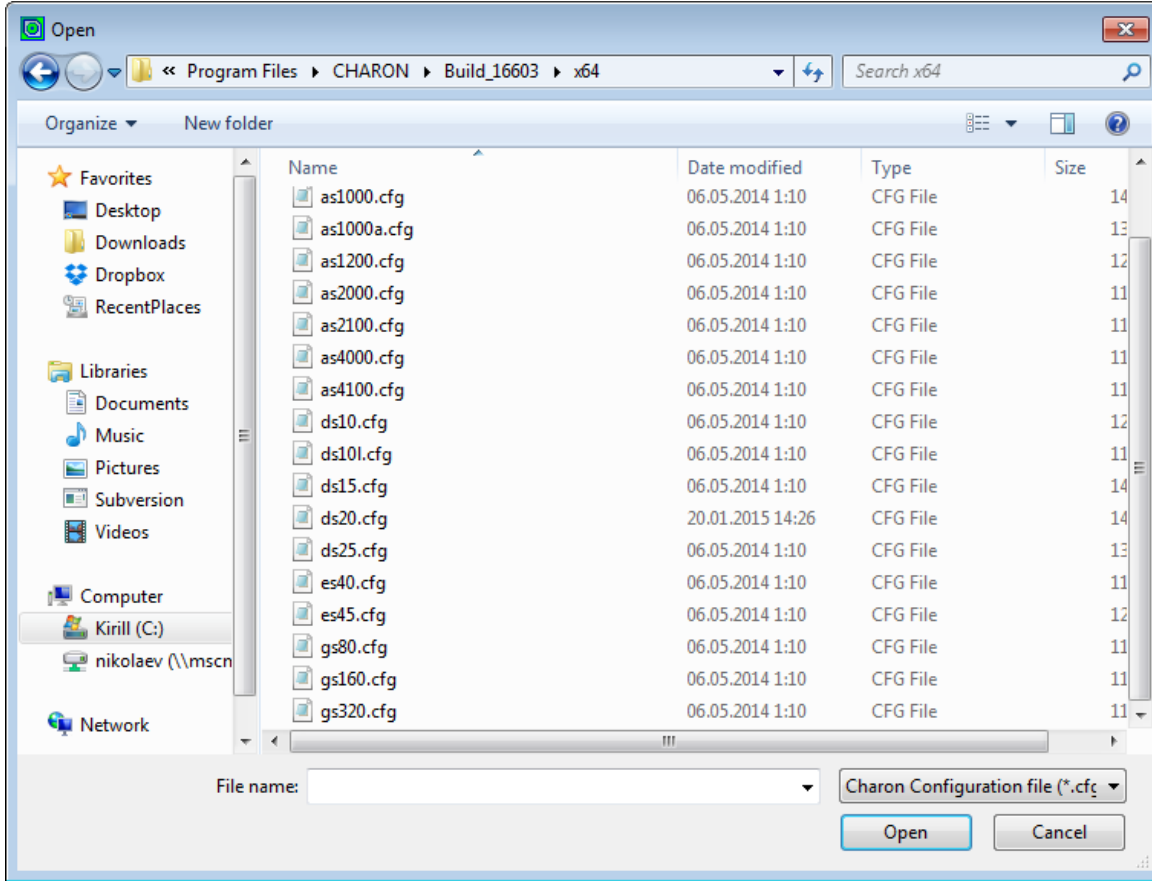


In the Launcher main interface, click the "Browse" button to select the desired CHARON configuration file:

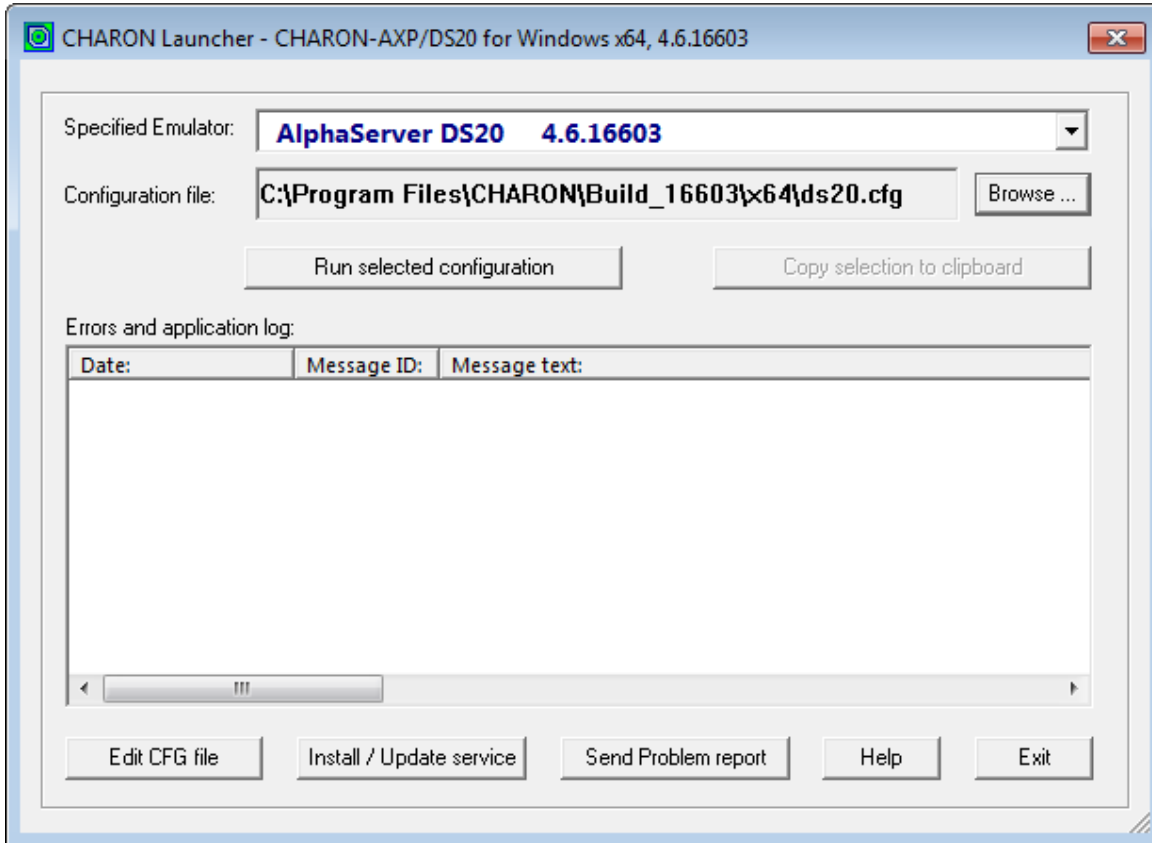




Select the desired configuration file:

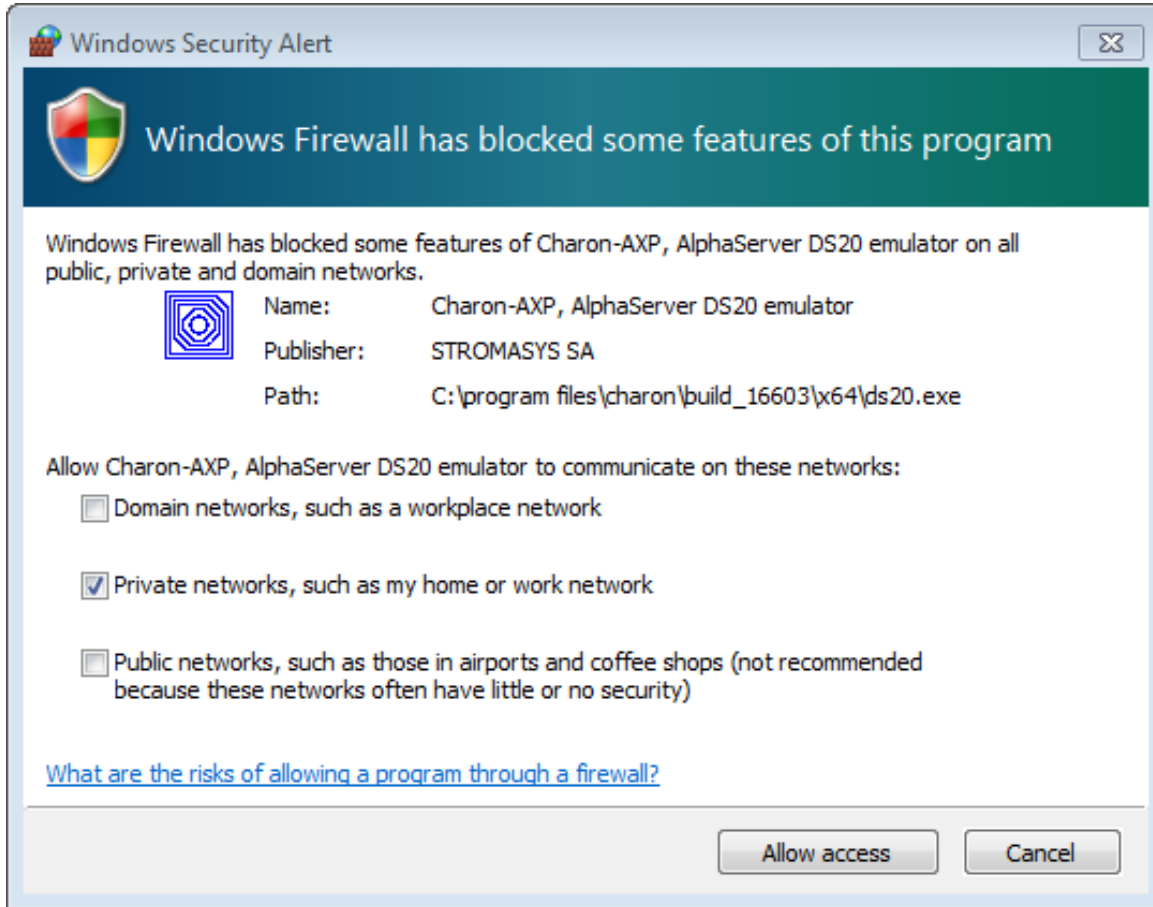


The Launcher will display the CHARON model specified in the selected configuration file:



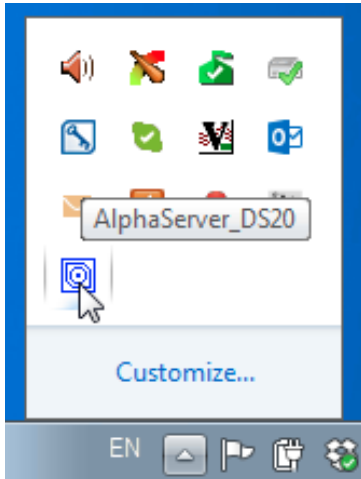
Now it is possible to run CHARON. Press "Run selected configuration" button.

Depending on system settings, Windows may display a Firewall warning dialog asking to confirm CHARON access to different networks.

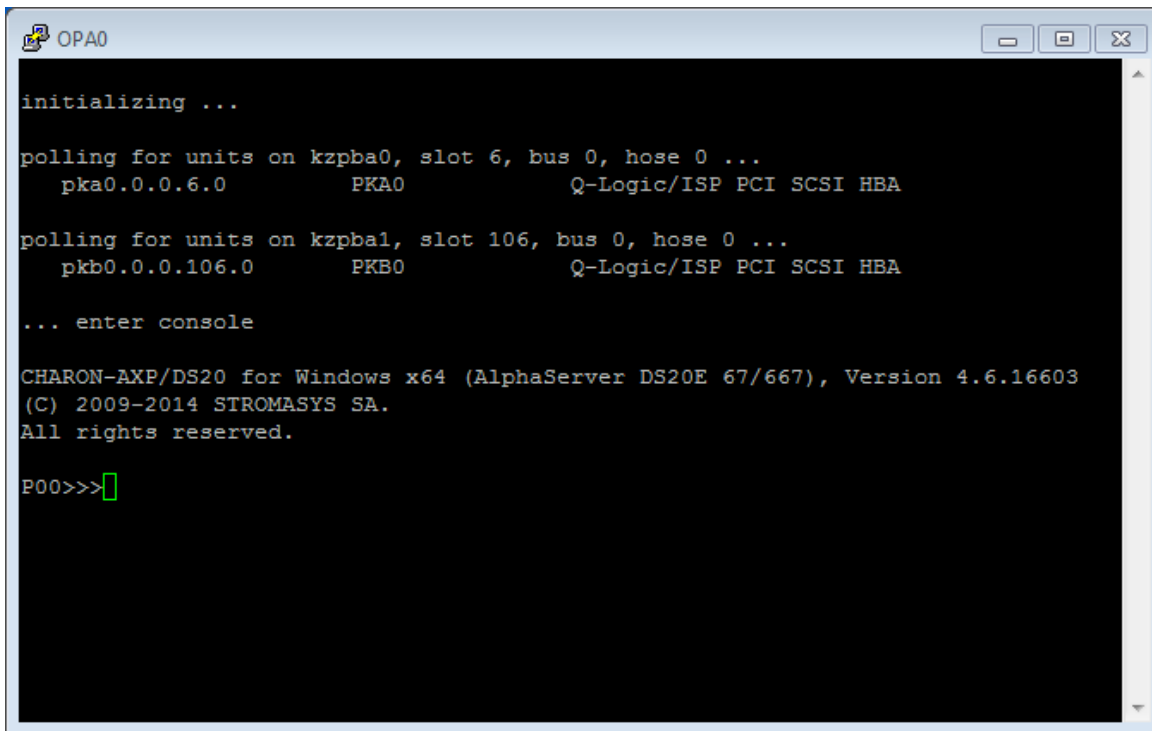


Check the desired options and press "Allow access" button.

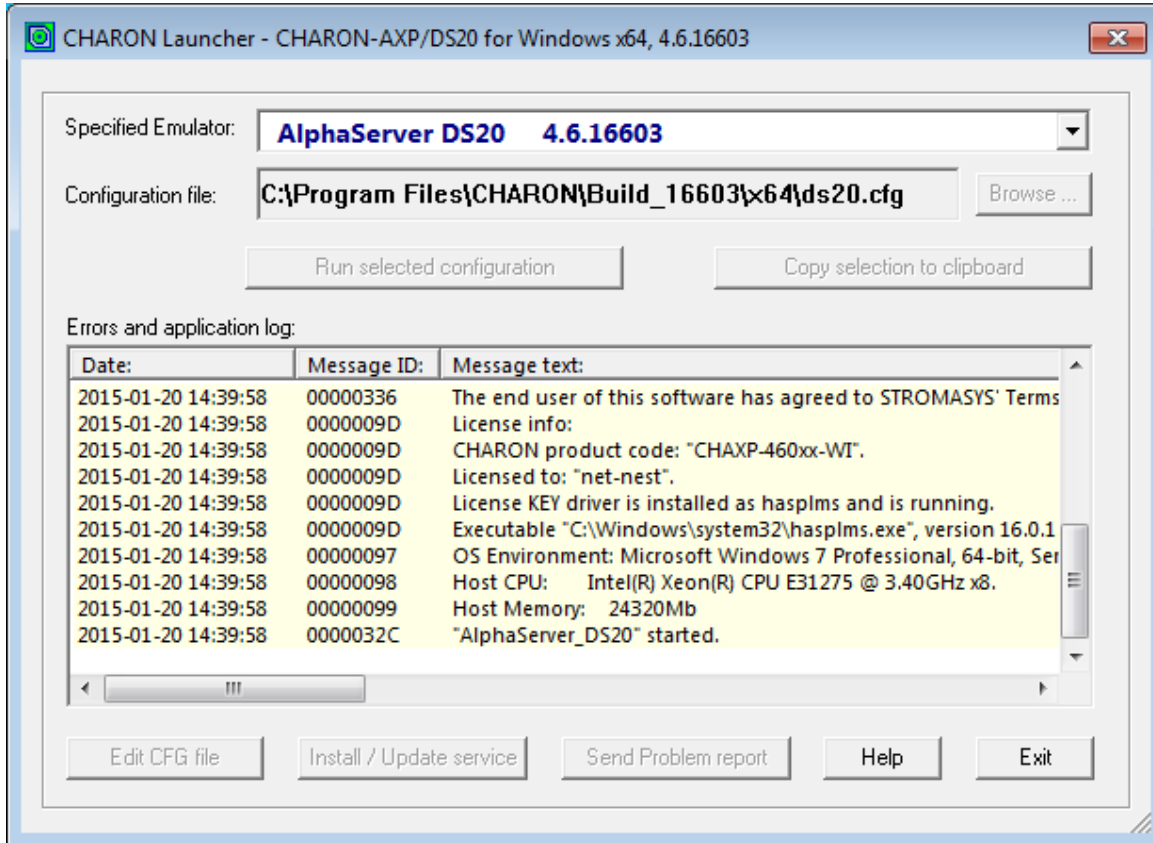
Next, the CHARON icon will appear in the system tray menu:



Right after that the HP Alpha emulated console will appear (in the default configuration. Some configurations may not assume console startup):



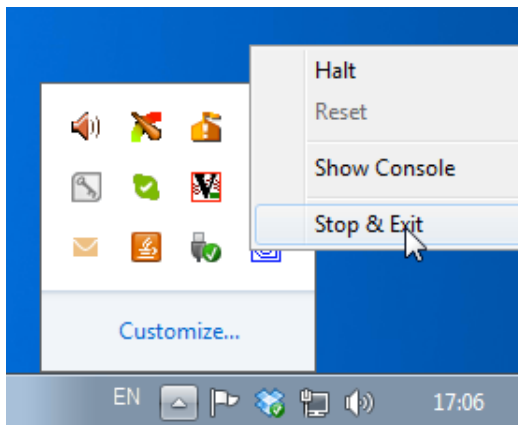
The Launcher will display CHARON log file:



It is possible to scroll the log file to analyze possible problems.

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

- **Configuration with no changes to the template configuration file**
  - Right click on CHARON icon in the system tray menu, choose "Stop & Exit":



Using this menu, it is also possible to halt the virtual HP Alpha, reset it or show its console if it has been closed

- Type "power off" in CHARON SRM console
- **Configuration file updated to enable "F6" button in configuration file to trigger exit from CHARON:**

```
set OPA0 stop_on = F6
```

- To exit, press "F6" in the console window
- Use CHARON icon tray menu as shown above

**!** Before stopping the CHARON-AXP service, shutdown the operating system running in CHARON-AXP.

After CHARON exit press "Exit" button to quit from the Launcher.

[Back to Table of Contents](#)

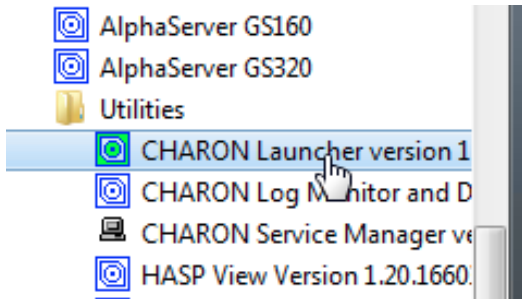
## Running as system service

It is possible to run CHARON-AXP as a Windows service.

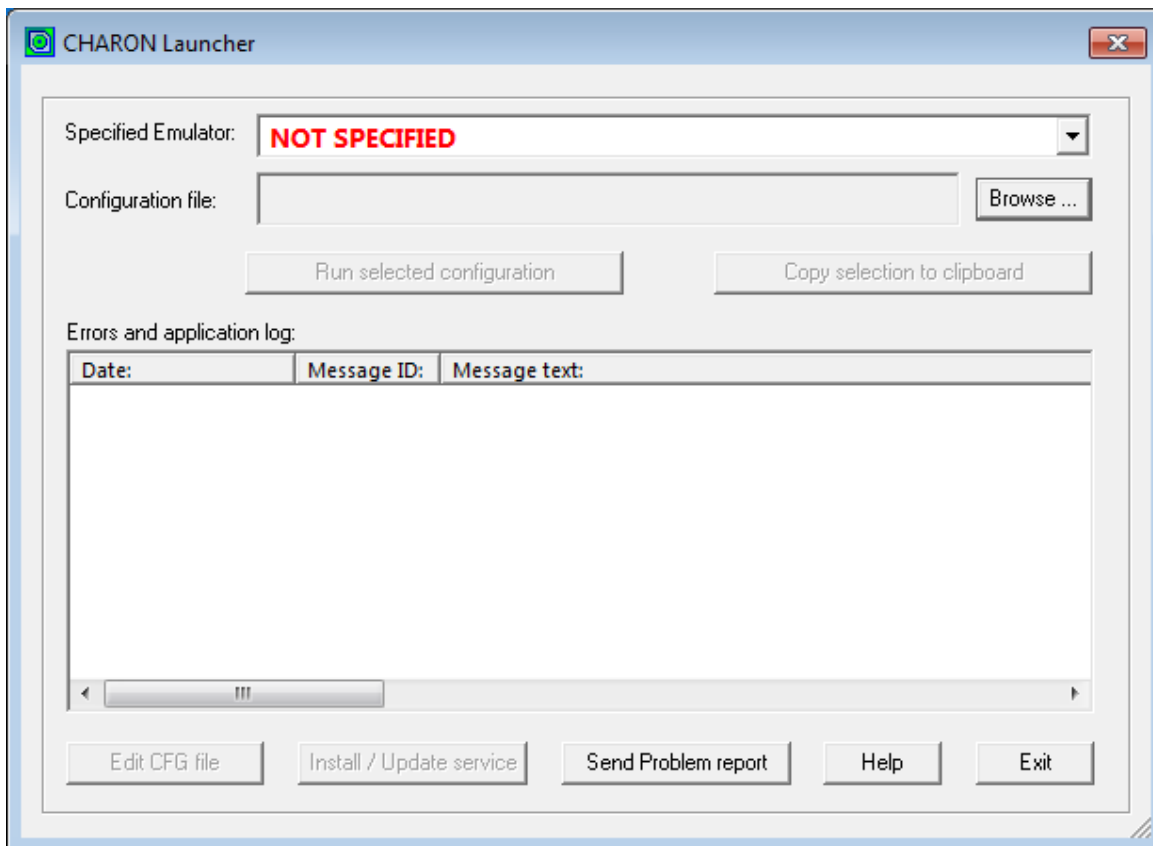
Follow the description below to establish and run CHARON-AXP as Windows service:

## Installation and start of CHARON-AXP service

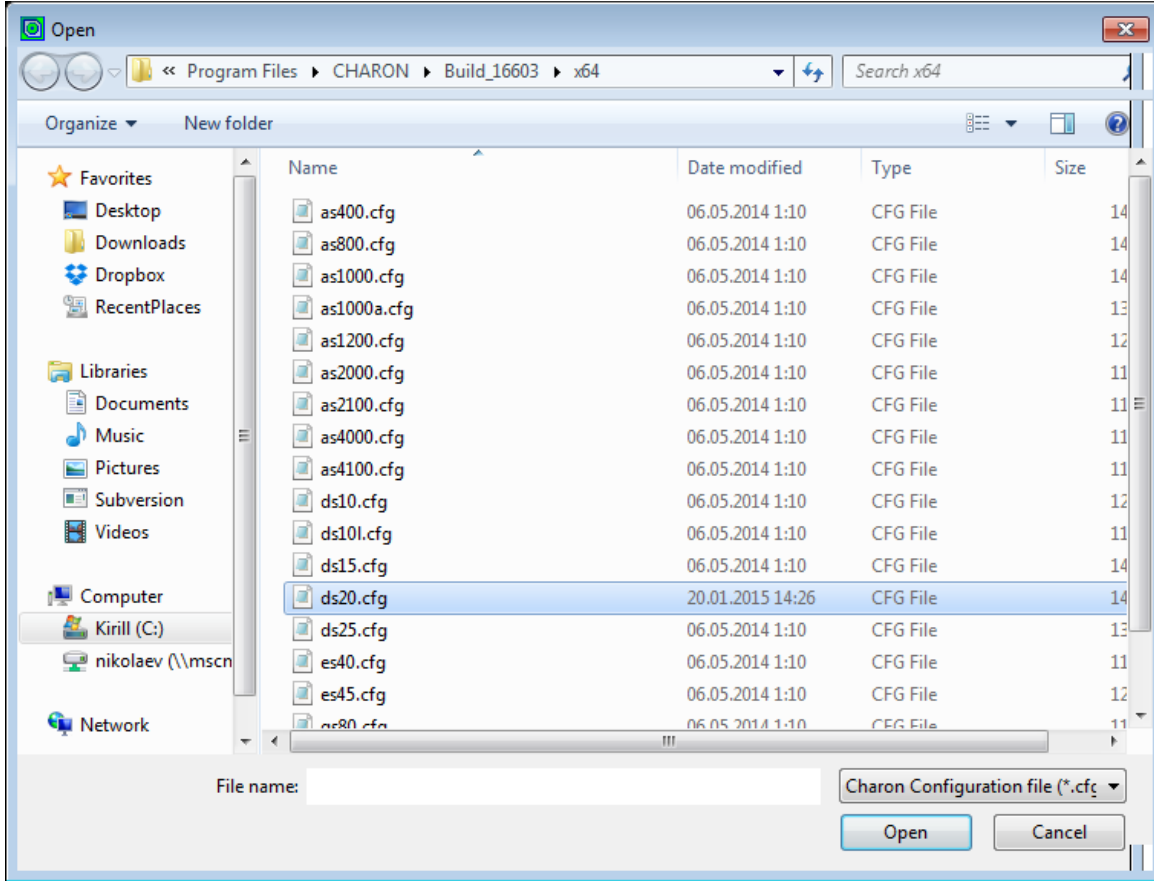
Open the "CHARON Launcher" utility from "Start" menu:



In the "CHARON Launcher" main interface press "Browse" button to select CHARON configuration to be used by CHARON running as Windows service:

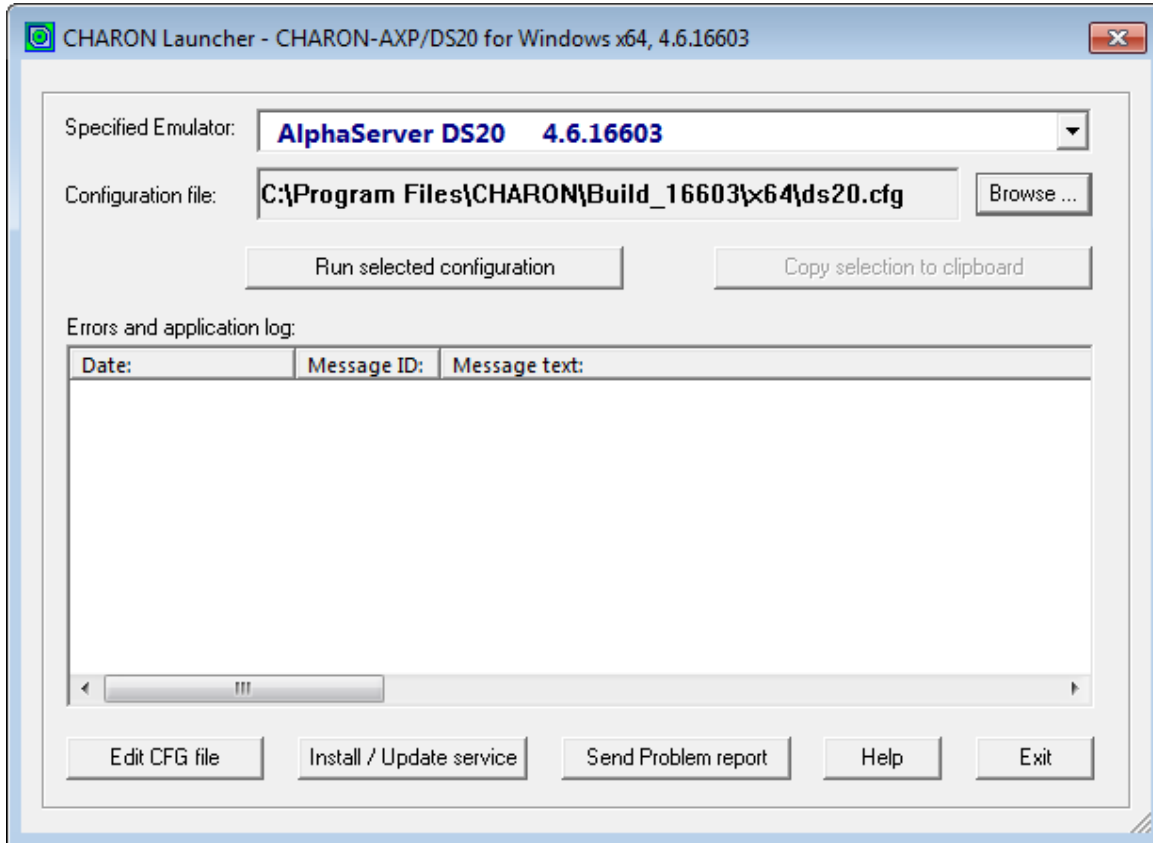


Choose the desired CHARON configuration file to be installed as service:





The Launcher will display the specific CHARON model name specified in the configuration file and enable management buttons:



Press "Edit CFG file" button and edit (with standard "Notepad" program) the chosen configuration file:

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

```
...
set session log="C:\Charon Instances\1st_ds20.log"
set rom container="C:\Charon Instances\1st_ds20.bin"
set toy container="C:\Charon Instances\1st_ds20.dat"

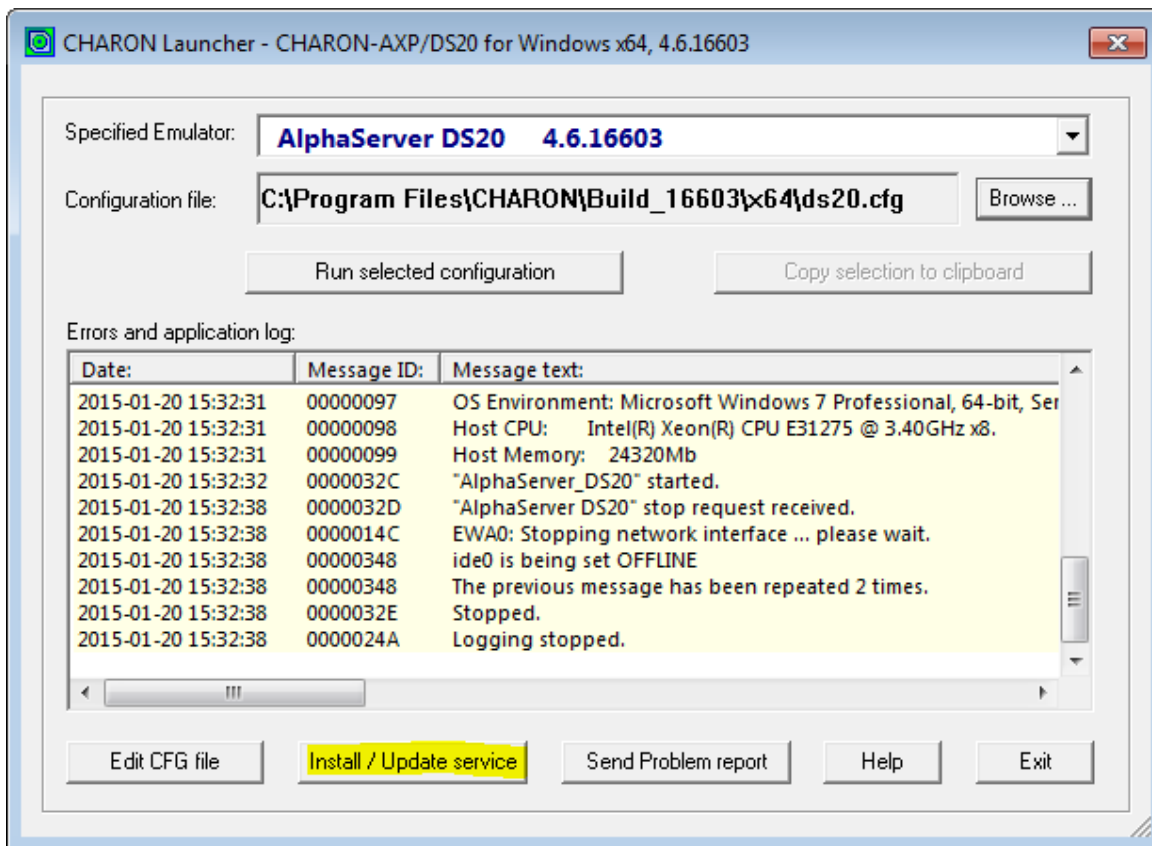
set PKA container[0]="C:\Charon Instances\1st_es40_boot_disk.vdisk"
...
```

- Make sure the same physical devices are not used by other CHARON-AXP services

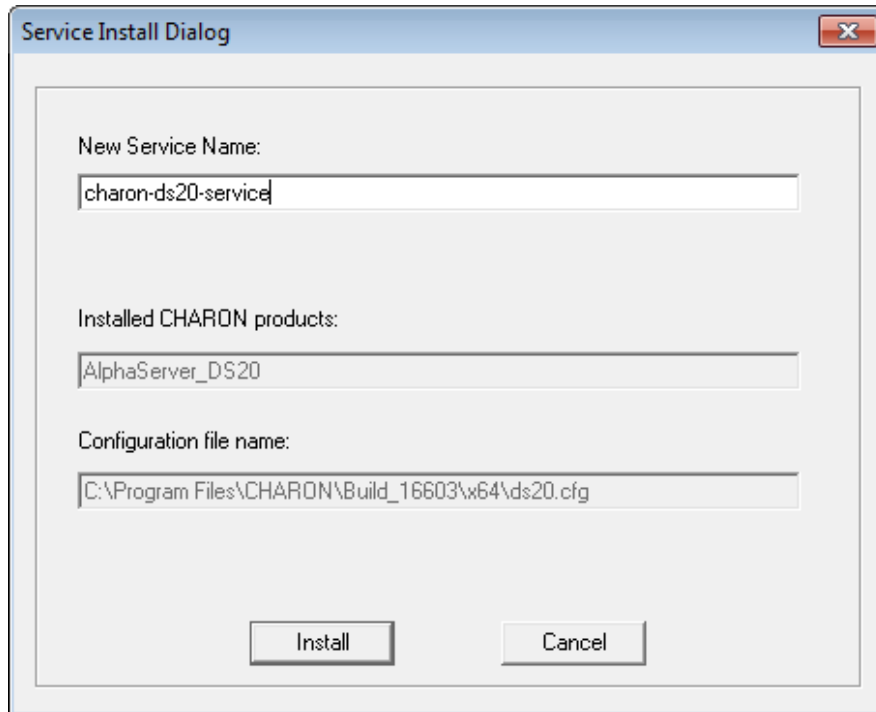
Save configuration file with "Notepad" save feature, exit from it.

Run the configuration to make sure that it is ready to be installed as Windows service; stop CHARON (see above how to do that).

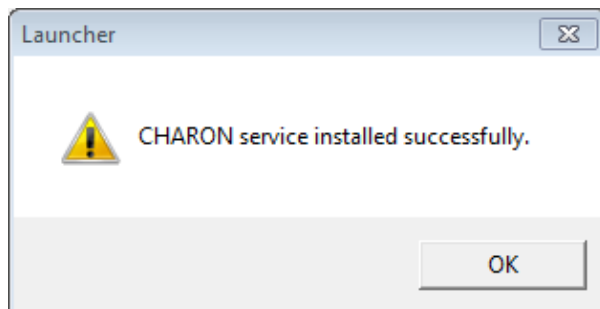
Press "Install / Update service" button in the Launcher:



A special dialog for adding new CHARON service will appear:

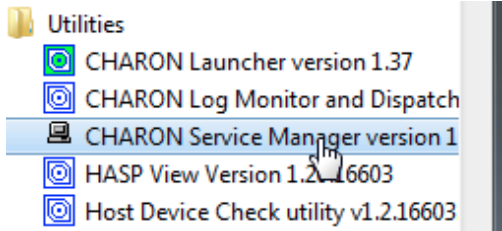


Enter a new service name (in the example above it is "charon-ds20-service") and press "Install" button. If the service has been installed correctly, a configuration dialog box will appear:

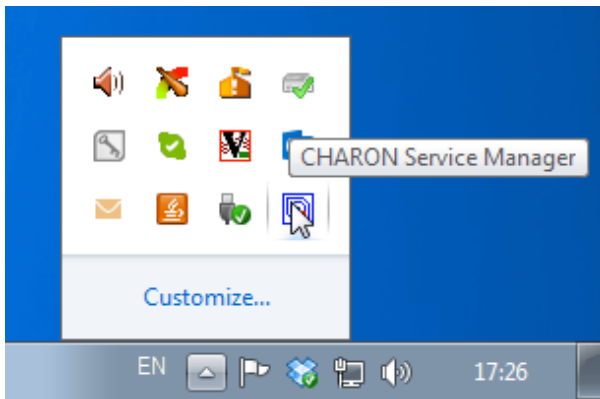


Press "Ok" button then exit from "Launcher" by pressing "Exit" button.

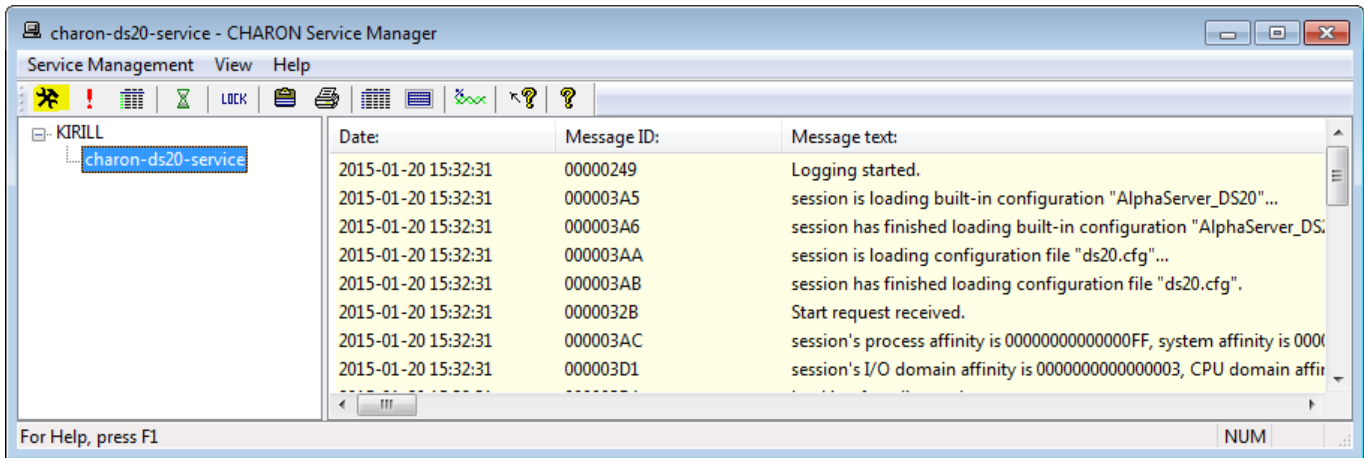
CHARON service is installed, now let's configure it. Start the "CHARON Service Manager" utility:



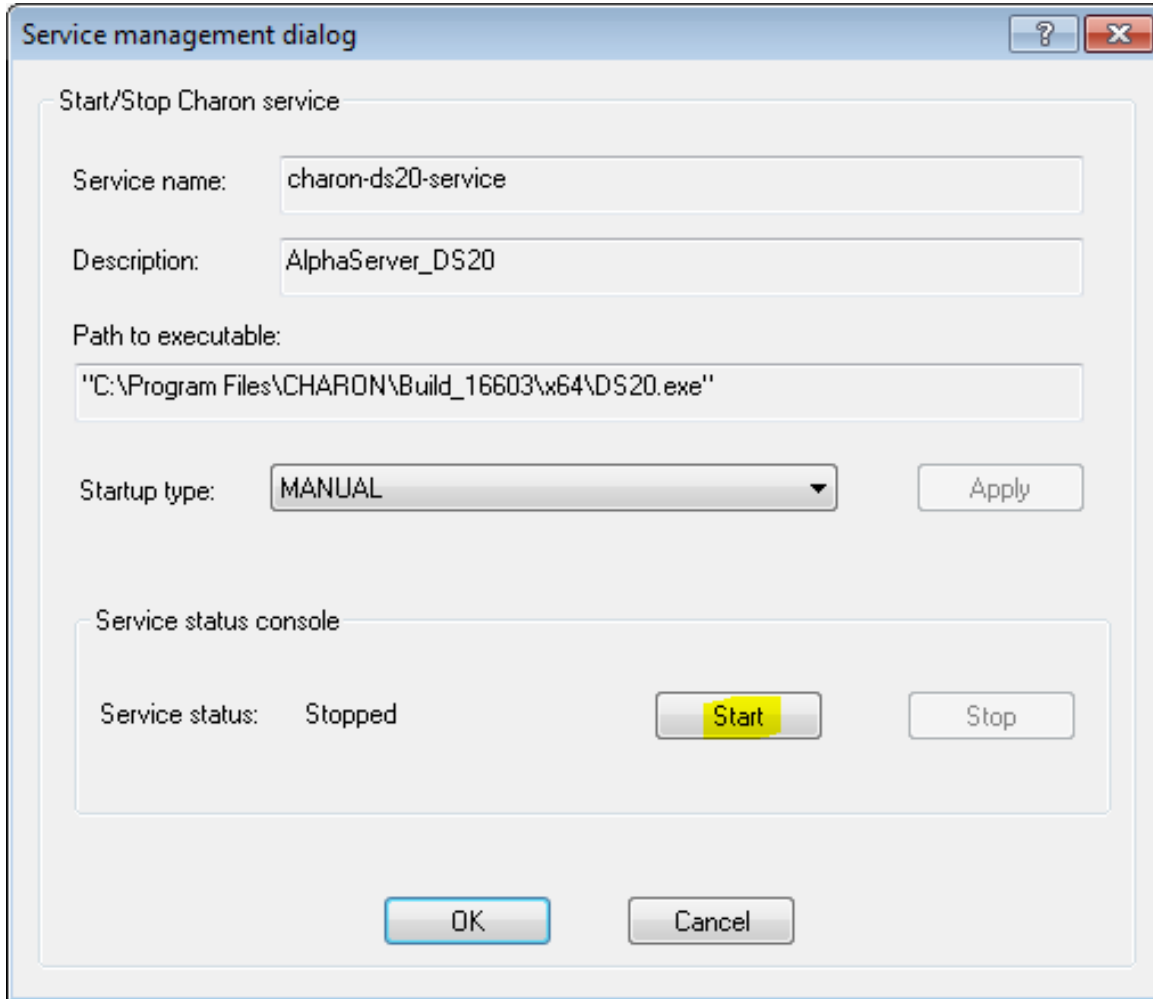
It is also possible to start the "CHARON Service Manager" utility from the system tray menu:



In the main window of the utility, select the target CHARON service ("charon-ds20-service" in our example) and press the "Service" button (labeled with yellow on the screenshot below):

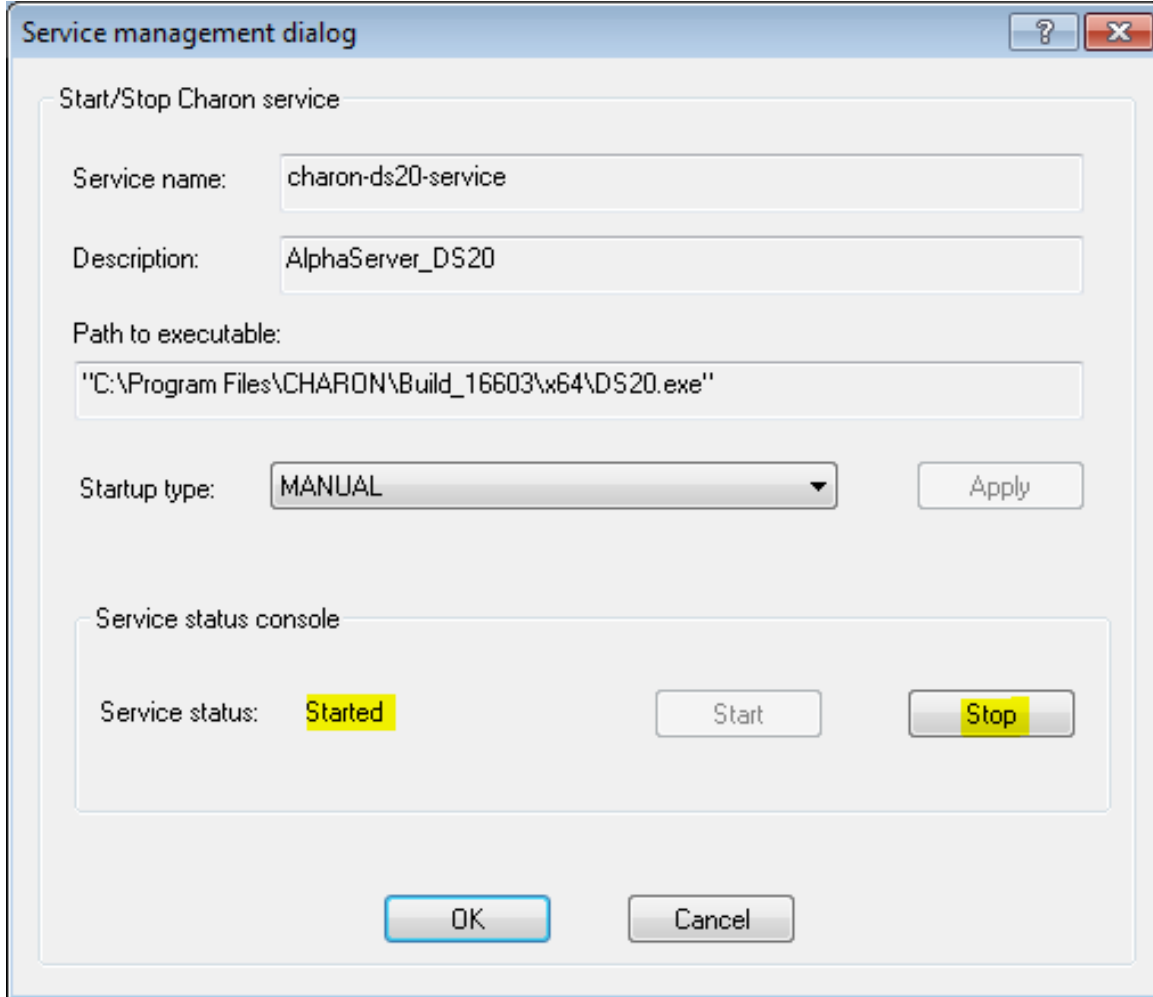


The "Service Management dialog" will appear:



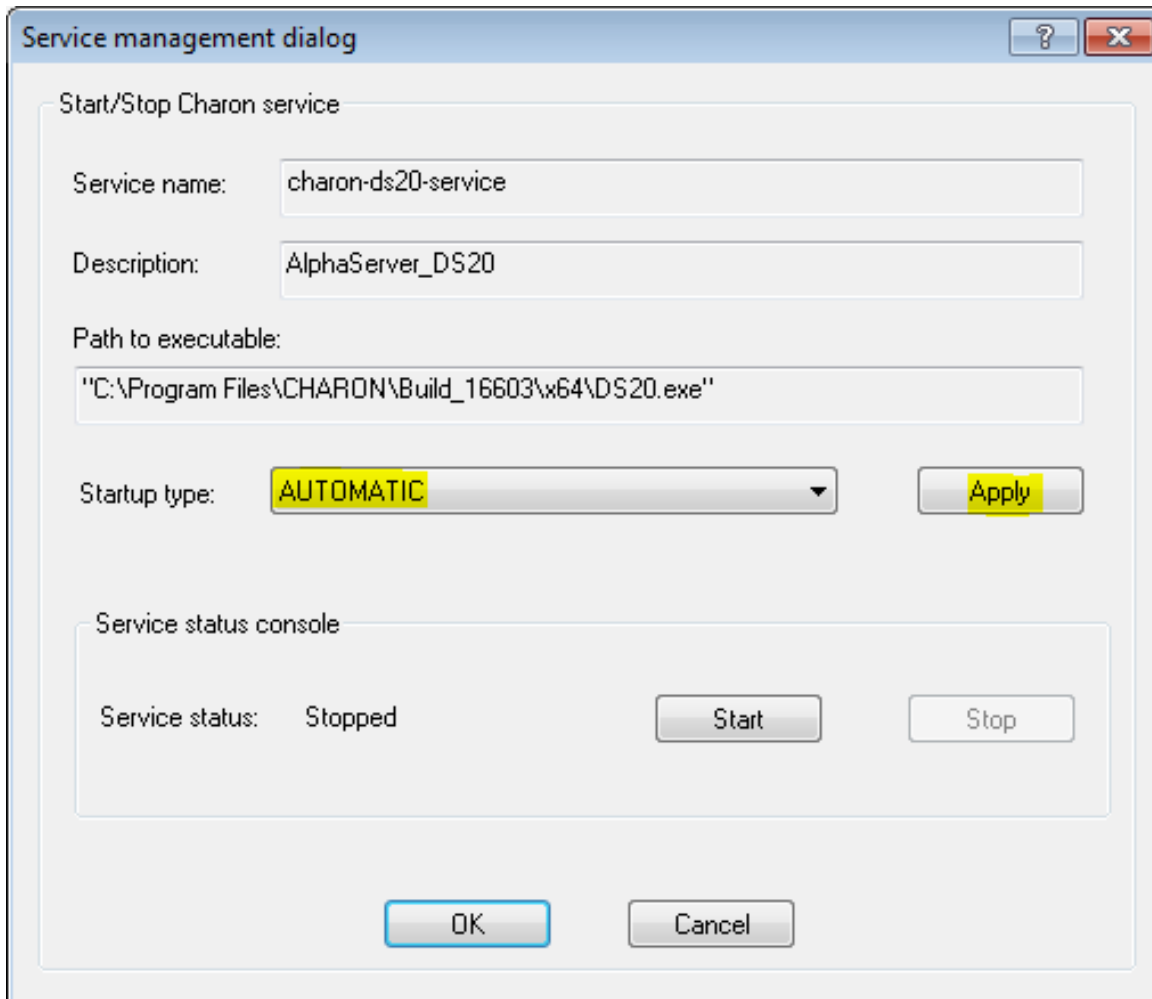
The "Service Management dialog" shows the current status of the selected CHARON service and permits changes to its start up type and to start and stop it. The example screenshot above shows that the service "charon-ds20-service" startup type is "MANUAL" (can be started / stopped only manually) and in the "Stopped" state - these are the default values.

Press the "Start" button to start the service:



CHARON will start and the Service status will be "Started". To stop the service, press the "Stop" button.

It is possible, and often desirable, to configure the selected CHARON service to start when host operating system starts. To enable this capability, select "AUTOMATIC" in the "Startup type" drop-down box and press the "Apply" button:



Note that a certain delay may occur before the Sentinel Run-time process finds the network license on CHARON-AXP host system startup. If the CHARON-AXP service is starting automatically on the host system, it may report a "License not found" error and exit.

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

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

where:

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

Example:

```
set session license_key_lookup_retry = 5
```

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

See [General Settings](#) section for more details.

[Back to Table of Contents](#)

## Management of CHARON-AXP service

To see the execution log of a CHARON service, select the service ("charon-ds20-service" in the example below) and the current log messages will be displayed in the right panel of the utility interface. You may need to scroll up to see startup/previous messages:

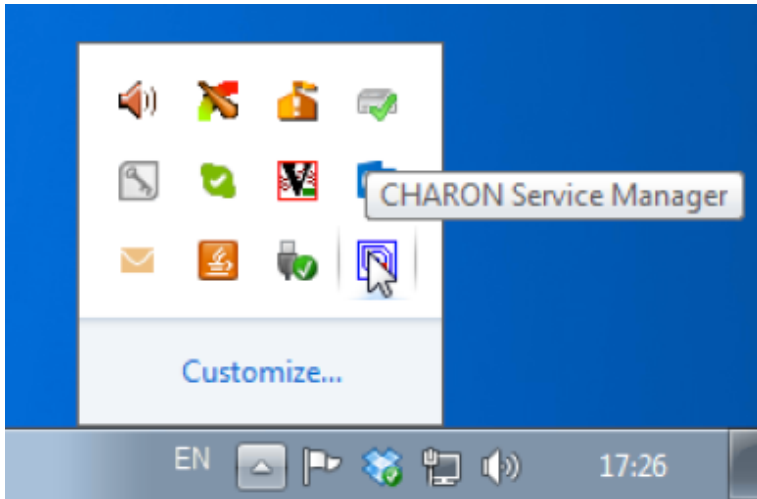
The screenshot shows the 'charon-ds20-service - CHARON Service Manager' window. The left pane shows a tree view with 'KIRILL' expanded and 'charon-ds20-service' selected. The main pane displays a log table with columns for Date, Message ID, and Message text. The log entries show the service starting at 16:20:56 and stopping at 16:22:37.

Date:	Message ID:	Message text:
2015-01-20 16:20:56	000003A9	session has verified service configuration checksum.
2015-01-20 16:20:56	0000032B	Start request received.
2015-01-20 16:20:56	000003AC	session's process affinity is 00000000000000FF, system affinity is 00000000000000FF.
2015-01-20 16:20:56	000003D1	session's I/O domain affinity is 0000000000000003, CPU domain affinity is 0000000000
2015-01-20 16:20:56	000003BA	Looking for a license key ...
2015-01-20 16:20:57	000003DC	... found license key 1422726238.
2015-01-20 16:20:57	0000024D	STROMASYS SA, (C) 2009-2014
2015-01-20 16:20:57	00000350	CHARON-AXP (AlphaServer DS20), V 4.6 B 16603, Dec 15 2014 / net-nest / 1422726238
2015-01-20 16:20:57	00000336	The end user of this software has agreed to STROMASYS' Terms and Conditions for S
2015-01-20 16:20:57	0000009D	License info:
2015-01-20 16:20:57	0000009D	CHARON product code: "CHAXP-460xx-WI".
2015-01-20 16:20:57	0000009D	Licensed to: "net-nest".
2015-01-20 16:20:57	0000009D	License KEY driver is installed as hasplms and is running.
2015-01-20 16:20:57	0000009D	Executable "C:\Windows\system32\hasplms.exe", version 16.0.1.41863.
2015-01-20 16:20:57	00000097	OS Environment: Microsoft Windows 7 Professional, 64-bit, Service Pack 1 (Build 7601
2015-01-20 16:20:57	00000098	Host CPU: Intel(R) Xeon(R) CPU E31275 @ 3.40GHz x8.
2015-01-20 16:20:57	00000099	Host Memory: 24320Mb
2015-01-20 16:20:57	0000032C	"AlphaServer_DS20" started.
2015-01-20 16:22:37	0000032D	"AlphaServer DS20" stop request received.
2015-01-20 16:22:37	0000014C	EWA0: Stopping network interface ... please wait.
2015-01-20 16:22:37	00000348	ide0 is being set OFFLINE
2015-01-20 16:22:37	00000348	The previous message has been repeated 2 times.
2015-01-20 16:22:37	0000032E	Stopped.
2015-01-20 16:22:37	0000024A	Logging stopped.

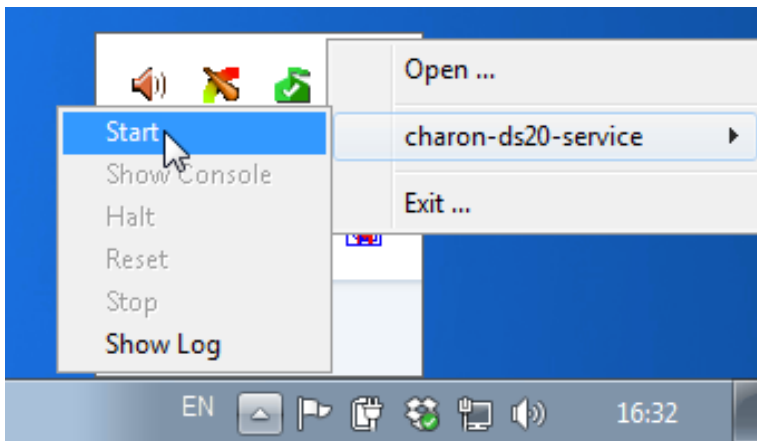
For Help, press F1



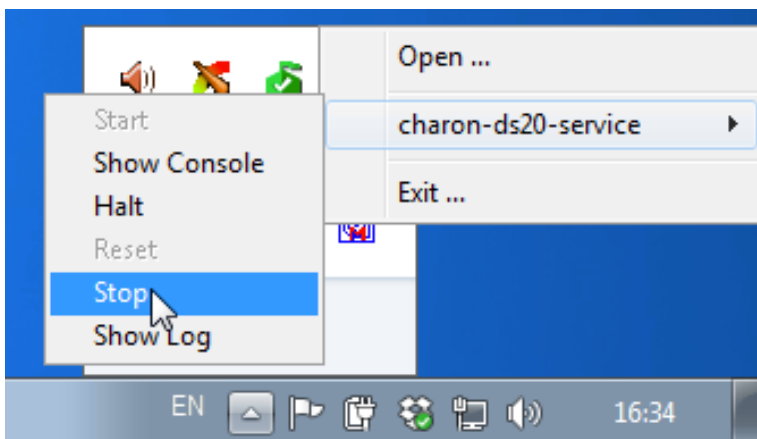
To control execution of CHARON services, open the "CHARON Service Manager" utility from the "Start" menu or use the system tray shortcut to manage the services w/o opening the main window of the utility:



Press the right button of the mouse to see a list of the installed CHARON services ("charon-ds20-service" in our example):



Select the Service to see the list of the available operations:



Explanation:

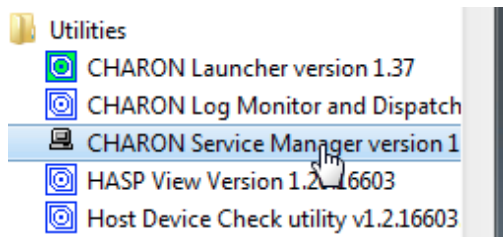
Menu Item	Function
<b>Start</b>	Start CHARON service
<b>Show Console</b>	Show CHARON service console if it has been closed or not active on the automatic service startup
<b>Halt</b>	Halt HP Alpha emulator associated with the selected CHARON service
<b>Reset</b>	Reset the virtual HP Alpha associated with the selected CHARON service
<b>Stop</b>	Stop CHARON service
<b>Show Log</b>	Show execution log of the selected CHARON service

Choose the desired action and click on it.

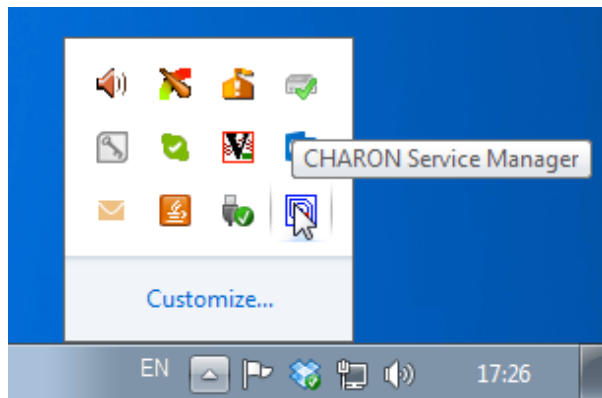
[Back to Table of Contents](#)

## Stopping CHARON-AXP service

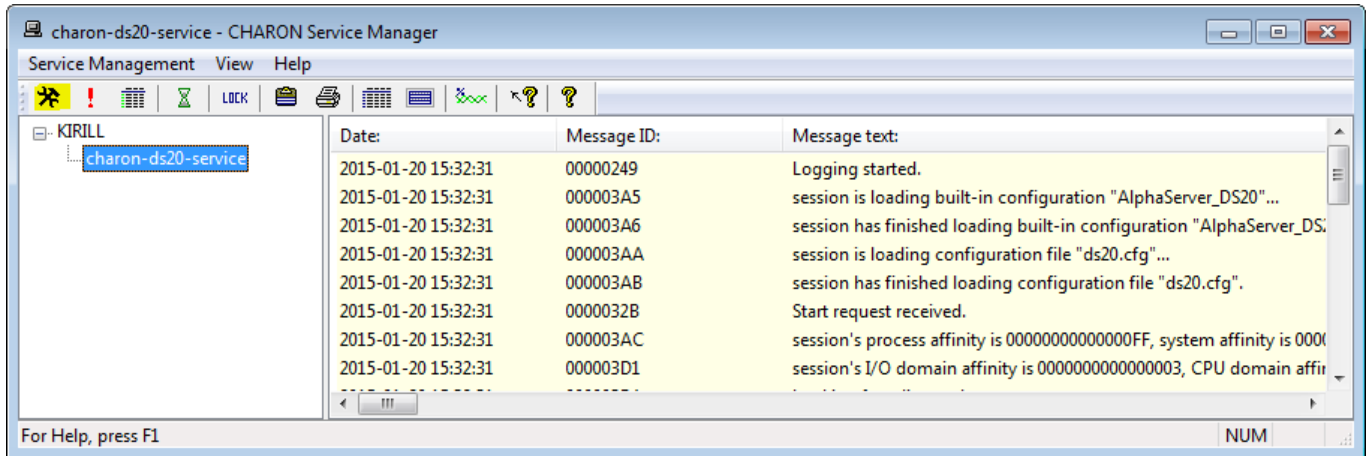
Open up "CHARON Service Manager" utility from the "Start" menu:



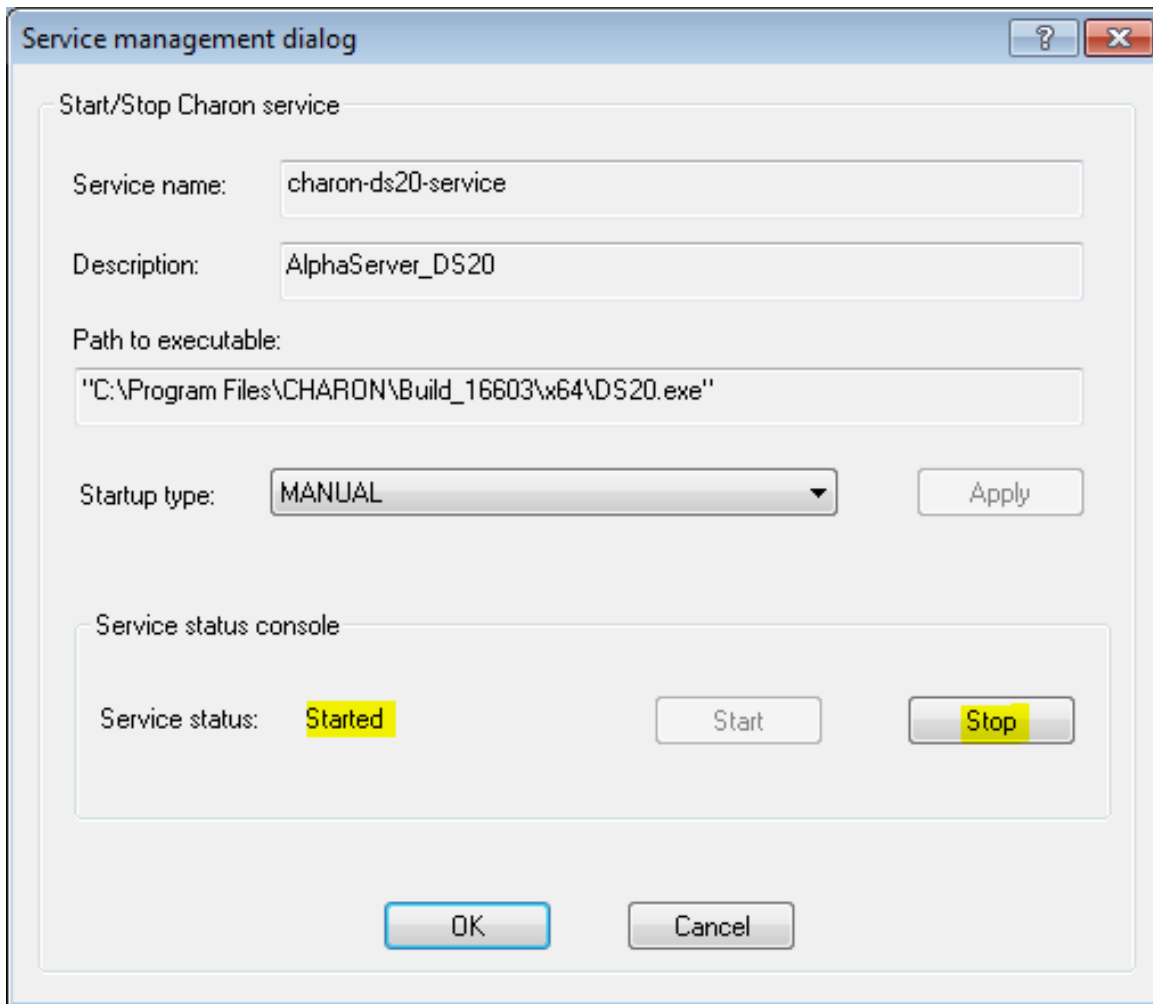
or from the system tray menu:



In the main window of the utility, select the target CHARON service ("charon-ds20-service" in our example) and press the "Service" button (labeled with yellow on the screenshot below):

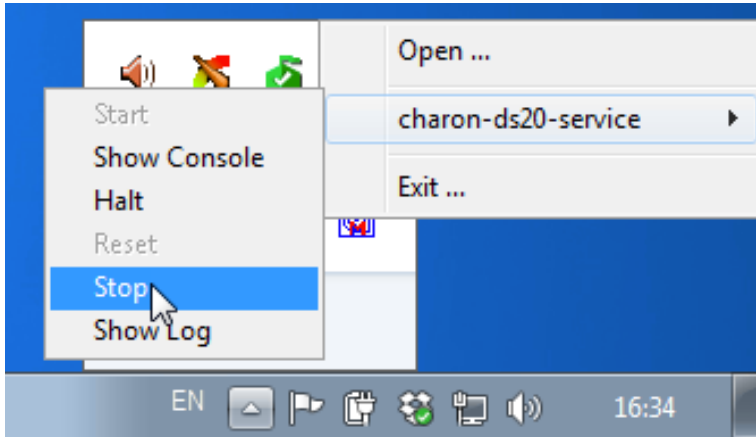


Press the "Stop" button in the resulting dialog box:



The CHARON Service can be Stopped using the shortcut on the Windows Task Bar.

Open up the "CHARON Service Manager" system tray menu, select the service you are going to stop and click on the "Stop" command:



⚠ Please note: before stopping the CHARON-AXP service, shutdown the operating system running in CHARON-AXP.

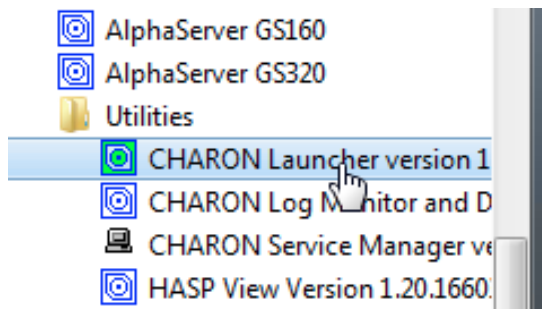
[Back to Table of Contents](#)

## Modification of CHARON-AXP service

Note that it is not enough just to modify the CHARON configuration file associated with a given CHARON service to apply the changes. You must also update the CHARON service with the "Launcher" utility.

Update the CHARON-AXP configuration file for the CHARON service.

Start up the "CHARON Launcher" utility:

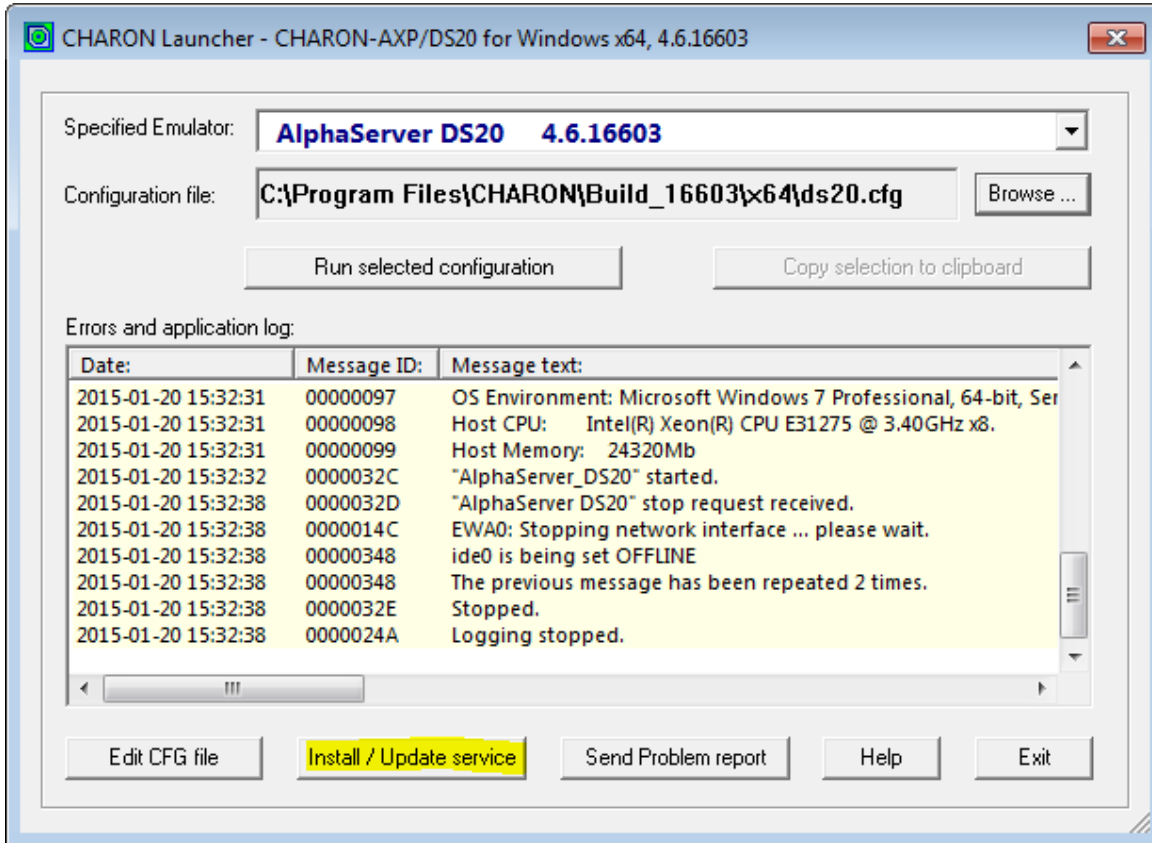


In the "CHARON Launcher" utility, select the modified configuration file and run it as described above.

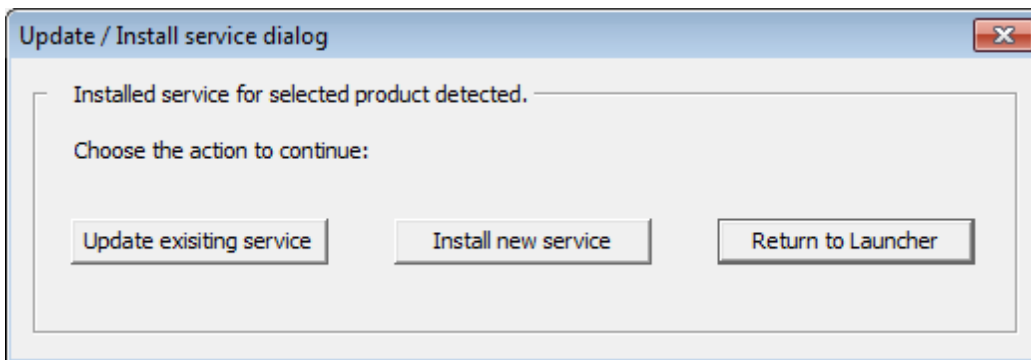
Make sure that the configuration starts without errors, then stop CHARON-AXP as described above.

If errors appear, stop CHARON-AXP, update its configuration file, start it again and make sure that the updated configuration works Ok this time, then stop CHARON-AXP.

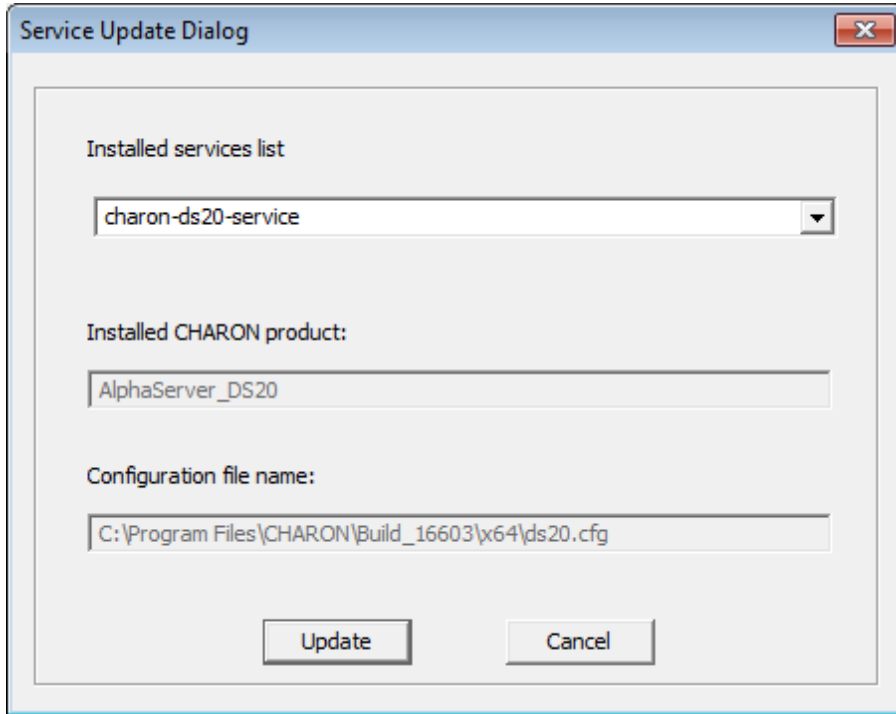
Once CHARON-AXP has been completely stopped, press the "Install / Update service" button:



The install / update dialog will appear:



Press "Update existing service" button:



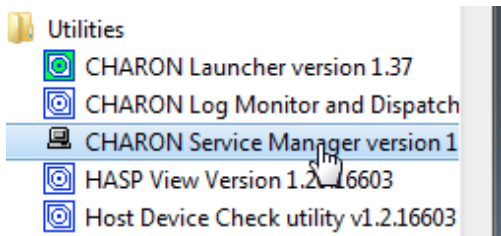
Select the service to update with the "Installed services list" drop-down box and press "Update" to apply.

⚠ Service update requires the CHARON-AXP service to be stopped.

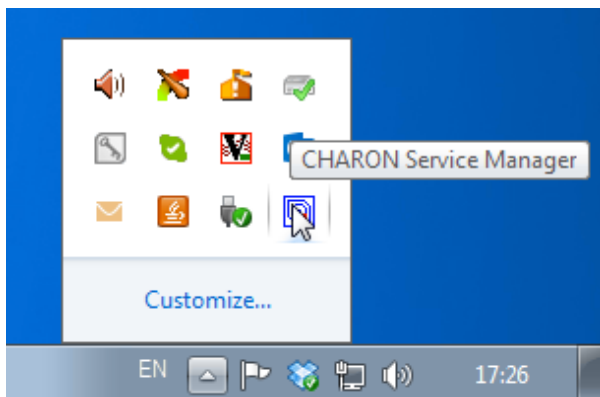
[Back to Table of Contents](#)

## Removing CHARON-AXP service

To remove the CHARON-AXP service, open the "CHARON Service Manager" utility from the "Start" menu:

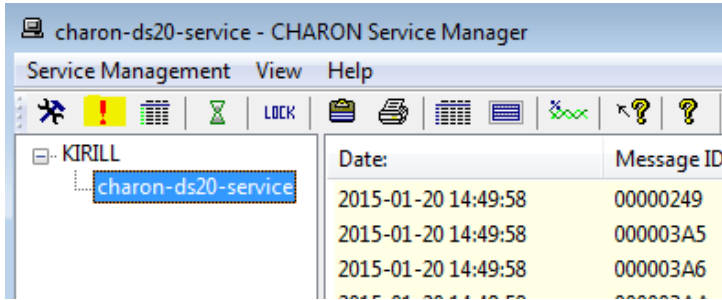


or from the system tray menu:

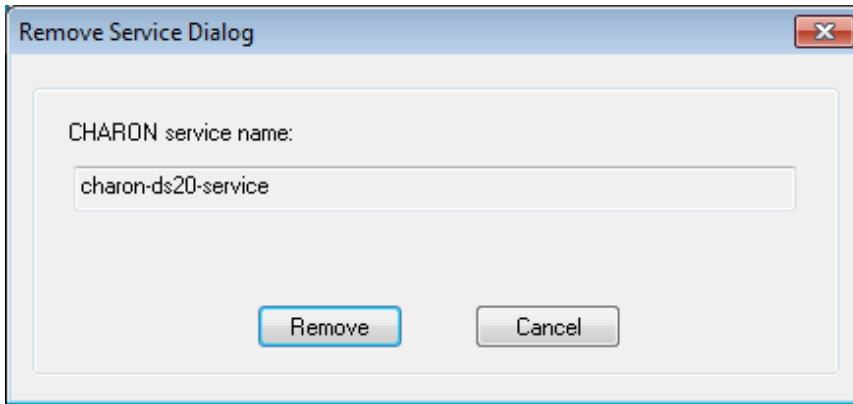


⚠ Please note: before stopping the CHARON-AXP service, shutdown the operating system running in CHARON-AXP.

In the main window of the utility, select the CHARON service to remove ("charon-ds20-service" in our example) and press the button with an exclamation sign on it:



Confirm removal of the selected CHARON service by pressing the "Remove" button:



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

[Back to Table of Contents](#)

# CHARON-AXP for Windows 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 button
- 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 mapping to a storage controller using its "presentation" mode
    - KGPSA-CA pass through mode
- FDDI support via DEFPA PCI FDDI controller in "pass through" mode
- Serial lines support via emulated PBXDA-xx family PCI controllers in "pass through" mode
- Auto boot

## Creation of your own configuration file using a template

By default, all the CHARON templates are located in the "C:\Program Files\CHARON\Build\_XXXXX\i64" folder. Copy the appropriate template configuration file(s) to your home directory (or to any directory intended for CHARON-AXP).

**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 "Notepad" editor (or use the "CHARON Launcher" utility to open the configuration file) and proceed with modification to reflect the exact features of the system you are going to emulate.

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

Note: lines 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, for example:

```
set session hw_model = AlphaServer_ES40
```

You must leave this line untouched.

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

[Back to Table of Contents](#)



## Configuration name

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

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

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

It is possible to specify a configuration name containing spaces, in this case use quotation marks:

```
set session configuration_name = "My ES40"
```

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

[Back to Table of Contents](#)

## Log file parameters

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

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

```
20150121:151204:INFO :0:00000249:ethane.cxx(9574): Logging started.
20150121:151204:INFO :0:000003A5:ethane.cxx( 744): session is loading built-in configuration
"AlphaServer_ES40"...
20150121:151204:INFO :0:000003A6:ethane.cxx( 777): session has finished loading built-in configuration
"AlphaServer_ES40".
20150121:151204:INFO :0:000003AA:ethane.cxx(1268): session is loading configuration file "es40.cfg"...
20150121:151204:INFO :0:000003AB:ethane.cxx(1310): session has finished loading configuration file
"es40.cfg".
20150121:151204:INFO :0:0000032B:ethane.cxx(2562): Start request received.
20150121:151204:INFO :0:000003AC:ethane.cxx( 688): session's process affinity is 00000000000000FF, system
affinity is 00000000000000FF.
20150121:151204:INFO :0:000003D1:ethane.cxx(1336): session's I/O domain affinity is 0000000000000003, CPU
domain affinity is 00000000000000FC
20150121:151204:INFO :0:000003BA:ll_sentine( 639): Looking for a license key ...
20150121:151205:INFO :0:000003DC:ll_sentine( 734): ... found license key 1422726238.
20150121:151205:INFO :0:0000024D:ethane.cxx(2693): STROMASYS SA, (C) 2009-2014
20150121:151205:INFO :0:00000350:ethane.cxx(2739): CHARON-AXP (AlphaServer ES40), V 4.6 B 16603, Dec 15
2014 / net-nest / 1422726238
20150121:151205:INFO :0:00000336:ethane.cxx(2766): 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
20150121:151205:INFO :0:0000009D:ethane.cxx(2843): License info:
CHARON product code: "CHAXP-460xx-WI".
Licensed to: "net-nest".
License KEY driver is installed as hasplms and is running.
Executable "C:\Windows\system32\hasplms.exe", version 16.0.1.41863.
20150121:151205:INFO :0:00000097:ethane.cxx(2852): OS Environment: Microsoft Windows 7 Professional,
64-bit, Service Pack 1 (Build 7601).
20150121:151205:INFO :0:00000098:ethane.cxx(2857): Host CPU: Intel(R) Xeon(R) CPU E31275 @ 3.40GHz x8.
20150121:151205:INFO :0:00000099:ethane.cxx(2862): Host Memory: 24320Mb
20150121:151205:INFO :0:0000032C:ethane.cxx(2598): "My_AlphaServer_ES40" started.
20150121:151221:INFO :0:0000032D:ethane.cxx(2644): "AlphaServer ES40" stop request received.
20150121:151221:INFO :0:00000348:ataunit.cx(1738): ide0 is being set OFFLINE
The previous message has been repeated 2 times.
20150121:151221:INFO :0:0000032E:ethane.cxx(2662): Stopped.
20150121:151221:INFO :0:0000024A:ethane.cxx( 328): Logging stopped.
```

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

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

## Rotating log (default)

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

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

Names of the rotating log files are composed as follows:

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

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

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

Note that "xxxxxxxx" is an increasing decimal number starting from "00000000" to separate log files with the same time of creation.

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

## Single log

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

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

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

```
set session log_method = overwrite
set session log = "C:\CHARON logs\my_ES40.log"
```

[Back to Table of Contents](#)

## CPU affinity


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

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

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

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

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

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

[Back to Table of Contents](#)

## Number of host CPUs dedicated to CHARON I/O

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

The "n\_of\_io\_cpus" overrides the default by specifying the number of I/O host CPUs explicitly

Example:

```
set session n_of_io_cpus=2
```

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

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

[Back to Table of Contents](#)

## Setting a specific HP Alpha model

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

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

```
#=====
#
# AlphaServer ES40 6/500
#
#-----
#set ace cpu_architecture = EV6
#set rom dsrdb[0] = 1816 system_name = "AlphaServer ES40 6/500"
#set rom version[1] = 1.98-4 version[2] = 1.92-5
#=====
#
# AlphaServer ES40 6/667
#
#-----
set ace cpu_architecture = EV67
set rom dsrdb[0] = 1820 system_name = "AlphaServer ES40 6/667"
```

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

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

[Back to Table of Contents](#)

## Reducing number of emulated CPUs

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

```
set session n_of_cpus=1
```

[Back to Table of Contents](#)

## Setting system serial number

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

```
set rom system_serial_number = SN01234567
```

[Back to Table of Contents](#)

## TOY and ROM containers

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

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

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

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

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

[Back to Table of Contents](#)

## Emulated memory (RAM) size

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

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

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

The following table shows all the parameters:

Hardware Model	RAM size (in MB)			
	Min	Max	Default	Increment
AlphaServer 400	64	1024	512	64
AlphaServer 800	256	8192	512	256
AlphaServer 1000	256	1024	512	256
AlphaServer 1000A	256	1024	512	256
AlphaServer 1200	256	32768	512	256
AlphaServer 2000	64	2048	512	64
AlphaServer 2100	64	2048	512	64
AlphaServer 4000	64	32768	512	64
AlphaServer 4100	64	32768	512	64
AlphaServer DS10	64	32768	512	64
AlphaServer DS15	64	32768	512	64
AlphaServer DS20	64	32768	512	64

AlphaServer DS25	64	32768	512	64
AlphaServer ES40	64	32768	512	64
AlphaServer ES45	64	32768	512	64
AlphaServer GS80	256	65536	512	256
AlphaServer GS160	512	131072	512	512
AlphaServer GS320	1024	262144	1024	1024

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

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

[Back to Table of Contents](#)

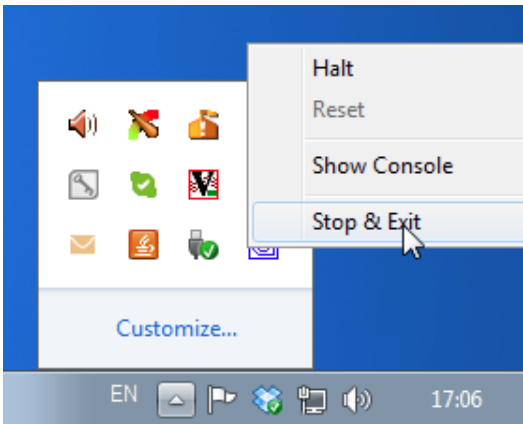
## Console

### Mapping to system resources

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

```
#load physical_serial_line OPA0 line="COM1:"  
#load virtual_serial_line OPA0 port=10003  
#load virtual_serial_line OPA0 port=10003 application="opa0.ht"  
load virtual_serial_line OPA0 port=10003 application="putty -load OPA0"  
#load virtual_serial_line OPA0 port=10003 application="c:\kea\user\opa0.ktc"
```

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	Mapping to host "\\.\COM<n>" ("COM<n>:") physical serial line.						
virtual_serial_line	<p>Mapping to an IP port of CHARON-AXP host. Using this mapping it is possible to connect to CHARON-AXP console and disconnect from it at any time. It is also possible to start up some chosen application to connect to the CHARON console IP port.</p> <table border="1"> <thead> <tr> <th>Parameter</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>port</td> <td>Specifies a unique IP port on the host to which the CHARON-AXP console will be mapped.</td> </tr> <tr> <td>application</td> <td>A string specifying a name of an application to be started together with CHARON to provide emulation of HP Alpha console (terminal emulator). It is assumed that this application has already been configured to use the IP port specified by the corresponding "port" parameter.</td> </tr> </tbody> </table> <p>Example:</p> <pre>load virtual_serial_line/chserial OPA0 port=10003 application="putty.exe -load OPA0"</pre> <p>In the example above, CHARON-AXP use the IP port "10003" for the HP Alpha console and starts the "Putty" terminal emulator (included in each CHARON distributive) with the "OPA0" saved session directing it to connect to the IP port "10003". If the terminal emulator is closed, it is always possible to reopen it with the "Show Console" command from the CHARON system tray icon:</p> 	Parameter	Description	port	Specifies a unique IP port on the host to which the CHARON-AXP console will be mapped.	application	A string specifying a name of an application to be started together with CHARON to provide emulation of HP Alpha console (terminal emulator). It is assumed that this application has already been configured to use the IP port specified by the corresponding "port" parameter.
Parameter	Description						
port	Specifies a unique IP port on the host to which the CHARON-AXP console will be mapped.						
application	A string specifying a name of an application to be started together with CHARON to provide emulation of HP Alpha console (terminal emulator). It is assumed that this application has already been configured to use the IP port specified by the corresponding "port" parameter.						

The default setting is "load virtual\_serial\_line/chserial OPA0 port=10003 application="putty.exe -load OPA0".

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="COM1:"
#load virtual_serial_line TTA0 port=10003
#load virtual_serial_line TTA0 port=10003 application="tta0.ht"
load virtual_serial_line TTA0 port=10003 application="putty -load TTA0"
#load virtual_serial_line TTA0 port=10003 application="c:\kea\user\tta0.ktc"
set COM2 line=TTA0
```

 Note there are a number of additional parameters for CHARON-AXP serial lines configuration. Follow [this link](#) for details.

## Exit on pressing F6 button

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

```
set OPA0 stop_on = F6
```

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

[Back to Table of Contents](#)

## Improve granularity of emulated timer

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

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

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

[Back to Table of Contents](#)

## ATAPI CD/DVD-ROM configuration

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

```
set ide container="\\.\CdRom0"
```

[Back to Table of Contents](#)

## Networking

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

All of them are configured in a similar way:

```
load DE500BA/dec21x4x EWA interface=EWA0
load packet_port/chnetwrk EWA0 interface="connection:Charon"
```

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

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

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

```
load DE500BA/dec21x4x EWA interface=EWA0
load packet_port/chnetwrk EWA0 interface="connection:Charon1"
load DE500BA/dec21x4x EWB interface=EWB0
load packet_port/chnetwrk EWB0 interface="connection:Charon2"
```

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 = "connection:Charon"
```

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

#set PKA container[0] = "<file-name>.vdisk"

#set PKA container[100]="\\.\PhysicalDrive0"
#set PKA container[101]="\\.\PhysicalDrive(DevID=XXX-XXXX-XXXX-XXXX-XXXX-XXXX-XXXX-XXXX)"
#set PKA container[102]="\\.\PhysicalDrive(iScsiTarget = <iSCSI target>, LUN = <LUN number>)"

#set PKA container[200]="\\.\ScsiN:X:Y:Z"

#set PKA container[300]="\\.\CdRom0"
#set PKA container[300]="\\.\CdRom<N>"

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

#set PKA container[500]="\\.\Tape0"
#set PKA container[500]="\\.\Tape<N>"

#set PKA container[600] = "<file-name>.vtape"


#set PKA container[600]="\\.\A:"
#set PKA media_type[600]="RX23"
```

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:


- **File representing a physical disk of the HP Alpha system (disk image)**
  - "<file-name>.vdisk"
 

These files can be created from scratch with "MkDisk" utility. Data and OS disks 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: "C:\My disks\my\_boot\_disk.vdisk"
- **Physical disk**
  - "\\.\PhysicalDrive<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.
- **Physical disk by its WWID**
  - "\\.\PhysicalDrive(DevID =XXXX-XXXX-XXXX-XXXX-XXXX-XXXX-XXXX-XXXX)"
 

 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.

DevID addresses the target physical disk by its WWID (hexadecimal 128-bit identifier assigned to the disk drive by its manufacturer/originator).

**Example:**

```
set PKA container[100]="\\.\PhysicalDrive(DevID= 6008-05F3-0005-2950-BF8E-0B86-A0C7-0001) "
```

▪ **iSCSI disks**

- "\\.\PhysicalDrive(iScsiTarget = <iSCSI target>, LUN = <LUN number>)"

Parameter	Description
iScsiTarget	Addresses the disk by its iSCSI target name.
LUN	Specifies LUN on connected iSCSI disk.

**Example:**

```
set PKA container[200]="\\.\PhysicalDrive(iScsiTarget= iqn.2008-04:iscsi.charon-target-test1, LUN= 1) "
```

▪ **SCSI device unknown to Windows for direct mapping, for example, a SCSI disk or tape reader**

- "\\.\Scsi<N>:<X>:<Y>:<Z>"

The values of N, X, Y and Z can be collected using special utility "Host Device Check" included in the CHARON distributive - or manually by investigation of the devices connected to CHARON host in the "Device Manager" applet.

Parameter	Description
N	A logical number assigned by host operating system (Microsoft Windows) to logical or host's physical storage resource such as physical SCSI HBA
X	An internal SCSI bus number (usually 0) on host's physical SCSI HBA
Y	A SCSI ID of physical SCSI target device attached to host's physical SCSI HBA
Z	A logical unit number inside physical SCSI target device attached to host's physical SCSI HBA

▪ **CD-ROM device**

- "\\.\CdRom<N>"

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

- "<file-name>.iso"

Mapping may also include the full path (recommended), for example: "C:\My disks\vms\_distributive.iso"

▪ **Host tape device**

- "\\.\Tape<N>"

▪ **File representing the tape (tape image)**

- "<file-name>.vtape"

These files are created automatically.

Mapping may also include a full path (recommended), for example: "C:\My tapes\backup.vtape"

▪ **Floppy drive**

- "\\.\A:"

▪ **Other type of drive, for example magneto-optical drive**

- "\\.\<N>:"


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

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

Some HP Alpha systems emulated by CHARON-AXP have already had one or two KZPBA controllers pre-loaded. 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 3 modes:

- Direct mapping to the host resources
- Usage of "presentation mode" of connected or external storage controllers
- 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:

```
load KGPSA FGA

#set FGA container[0] = "<file-name>.vdisk"

#set FGA container[100]="\\.\PhysicalDrive0"
#set FGA container[200]="\\.\PhysicalDrive(DevID=XXX-XXX-XXX-XXX-XXX-XXX-XXX-XXX)"
#set FGA container[300]="\\.\PhysicalDrive(iScsiTarget = <iSCSI target>, LUN = <LUN number>)"
```

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

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

- "<file-name>.vdisk"

These files can be created from scratch with "MkDisk" utility. Data and OS disks 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: "C:\My disks\my\_boot\_disk.vdisk"

- **Physical disk**

- "\\.\PhysicalDriveN"

⚠ 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.

- **Physical disk by its WWID**

- "\\.\PhysicalDrive(DevID =XXXX-XXXX-XXXX-XXXX-XXXX-XXXX-XXXX-XXXX)"

⚠ 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.

DevID addresses the target physical disk by its WWID (hexadecimal 128-bit identifier assigned to the disk drive by its manufacturer/originator).

Example:

```
set PKA container[100]="\\.\PhysicalDrive(DevID= 6008-05F3-0005-2950-BF8E-0B86-A0C7-0001)"
```

- **iSCSI disks**

- "\\.\PhysicalDrive(iScsiTarget = <iSCSI target>, LUN = <LUN number>)"

iScsiTarget addresses the disk by its iSCSI target name. LUN specifies LUN on connected iSCSI disk.

Example:

```
set PKA container[200]="\\.\PhysicalDrive(iScsiTarget= iqn.2008-04:iscsi.charon-target-test1, LUN= 1)"
```

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 mapping to a storage controller using its "presentation" mode

Some storage controllers allows CHARON-AXP to use their resources using so called "presentation" mode.

In this type of mapping CHARON-AXP automatically creates a set of virtual FC devices for each of the units provided by the storage controller and connects to them through its KGPSA-CA FC adapter.

The main benefit in this type of mapping is a flexible way of the virtual disks management depending on the mapped storage controller configuration. For example if an extra disk is added to the storage controller it automatically appears as a new disk unit on the corresponding KGPSA-CA virtual adapter mapped to that storage controller.

Below is an example of KGPSA-CA PCI FC disk controller, mapped to some storage controller (for example SAN) using its "presentation" mode:

```
load KGPSA FGA storage_controller_path_id = 5008-05F3-0005-2950-5008-05F3-0005-2951
```

This line loads an instance of KGPSA-CA controller and maps it to some external controller having ID "5008-05F3-0005-2950-5008-05F3-0005-2951".

Type of mapping	Description
storage_controller_path_id = <Storage controller path ID>	"Storage controller path ID" is a storage (for example SAN) controller path ID.  This ID can be obtained from the special utility " <a href="#">Host Device Check</a> ". Once specified, all the disks attached to the storage are automatically mapped as disk units to CHARON-AXP.

[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 = "PCI bus X, device Y, function Z"
#set FGB host_bus_location = "PCI bus A, device B, function C"
```

The "host\_bus\_location" parameter addresses the host EMULEX LightPulse PCI/PCI-X/PCIe FC adapter in the following way:

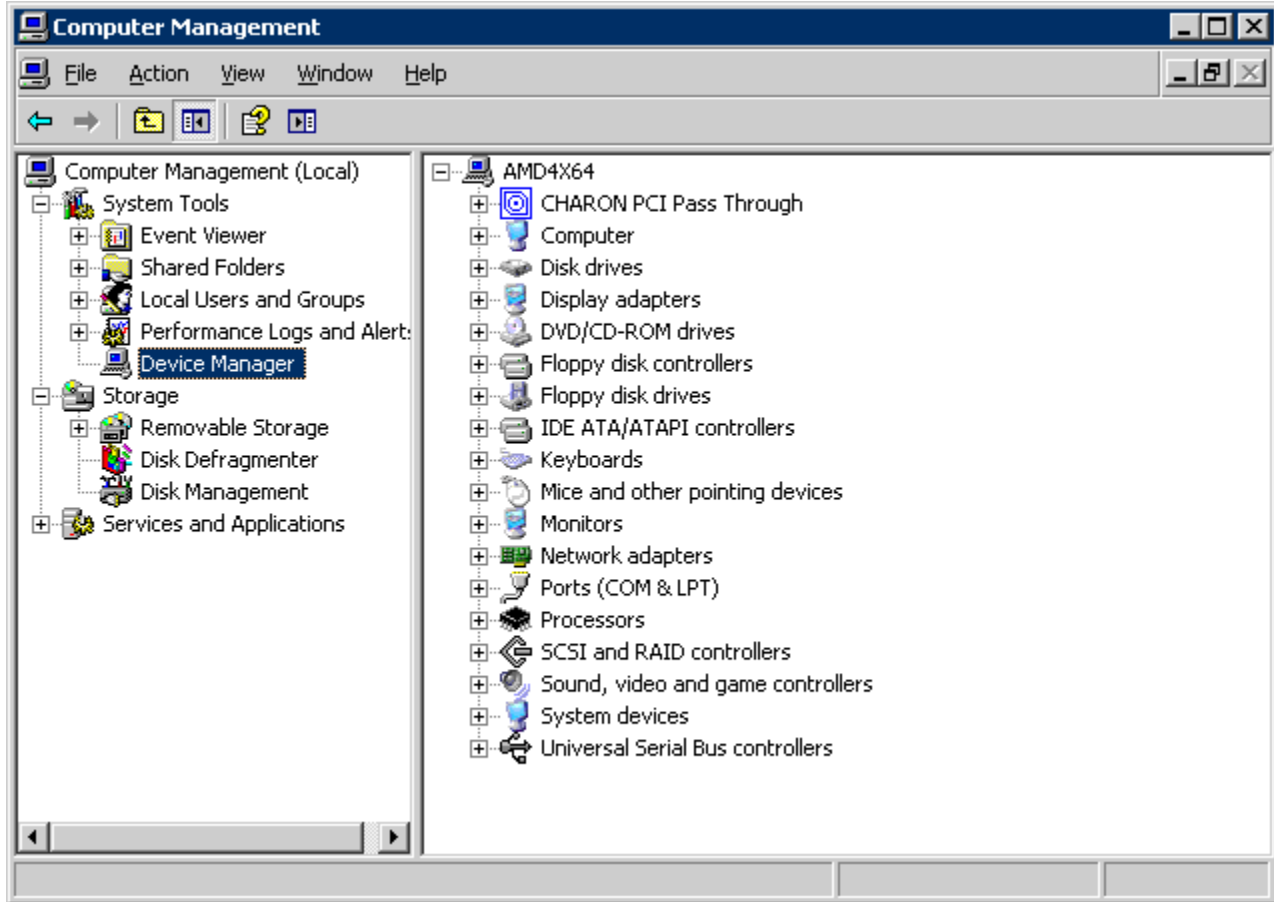
Parameters	Description
"PCI bus X"	PCI bus number of the host EMULEX LightPulse PCI/PCI-X/PCIe FC adapter
"device Y"	PCI bus device number of the host EMULEX LightPulse PCI/PCI-X/PCIe FC adapter
"function Z"	The "function" parameter of the the host EMULEX LightPulse PCI/PCI-X/PCIe FC adapter

To establish "pass through" mode do the following:

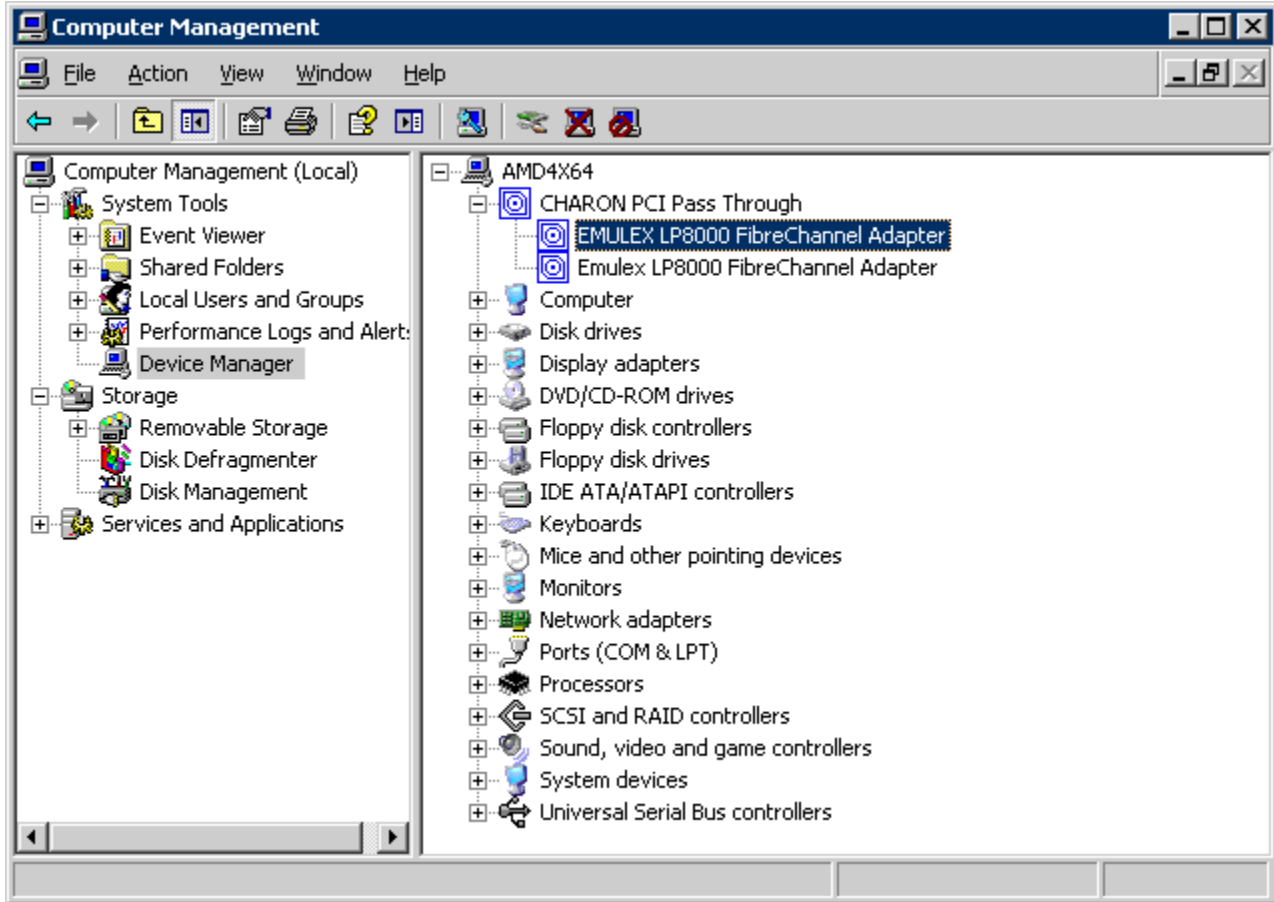
1. Install the EMULEX LightPulse PCI/PCI-X/PCIe FC adapter (see below for a list of supported models) to some spare PCI/PCI-X/PCIe slot of the host system
2. Boot a Windows operating system
3. Install the EMULEX LightPulse PCI/PCI-X/PCIe FC adapter driver from the following directory "C:\Program Files\CHARON\Drivers\EMULEX\_X.X.0.XXXXX" by choosing the "Install from a list or specific location (Advanced)" option and then selecting the "emulex\_lp\_ppt\_amd64.inf" file.
4. Reboot the host

Now it is possible to collect the parameters for CHARON-AXP mapping to the EMULEX LightPulse PCI/PCI-X/PCIe FC adapter.

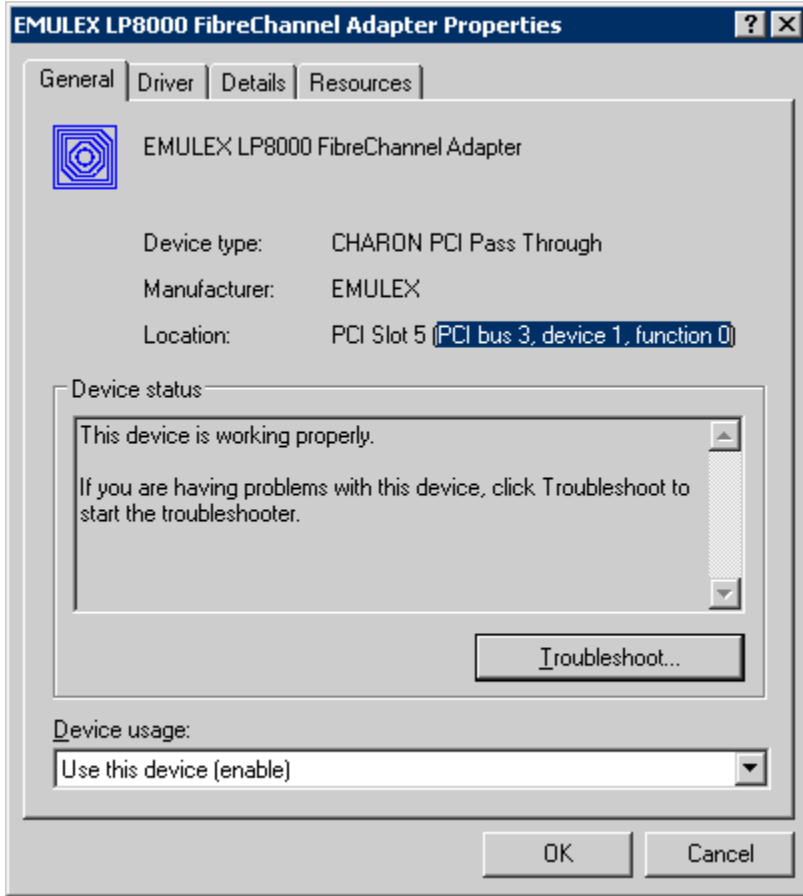
Open "Computer Management" application and select "Device Manager":



On the right panel select desired physical EMULEX LP FibreChannel adapter under "CHARON PCI Pass Through":



Open its properties sheet by double-clicking on selected adapter:



The "Location:" on the above picture gives X, Y, and Z for the "host\_bus\_location" parameter, for example:

```
set FGA host_bus_location = "PCI bus 3, device 1, function 0"
```

**!** Non-US-EN installations of Windows may present "Location:" string in local language, but "host\_bus\_location" parameter requires English notation, so the words "PCI", "bus", "device", and "function" must be specified in English.

The following is the 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	LPe1150 (FC2142SR, A8002A)	LPe11000

[Back to Table of Contents](#)

## FDDI support via DEFPA PCI FDDI controller in "pass through" mode

Optionally it is possible to configure DEFPA PCI FDDI controller in "pass through" mode, mapped to a physical DEFPA FDDI adapter installed on the host:


```
load defpa FDDI host_bus_location = "PCI bus X, device Y, function Z"
set FDDI bus=pci_1 device=1 function=0 irq=24 irq_bus=isa
```

Pay attention to proper placement of the emulated DEFPA adapter on the virtual HP Alpha PCI bus (it is controlled by "bus", "device", "function", "irq" and "irq\_bus" parameters). Refer to the [this chapter](#) of this Guide for more information.

The "host\_bus\_location" parameter addresses the host DEFPA FDDI adapter in the following way:

Parameters	Description
"PCI bus X"	PCI bus number of the host DEFPA FDDI adapter
"device Y"	PCI bus device number of the host DEFPA FDDI adapter
"function Z"	The "function" parameter of the the host DEFPA FDDI adapter

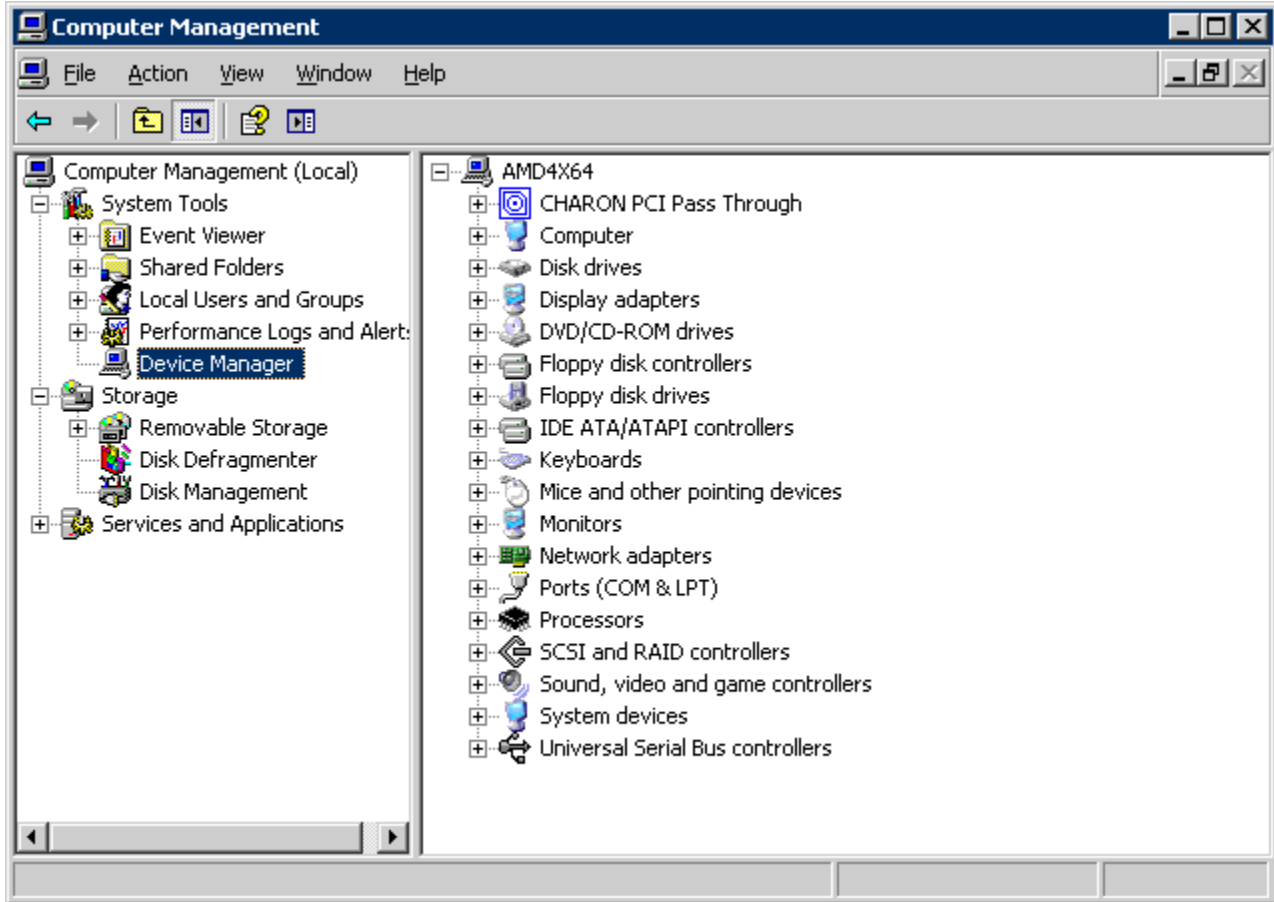
To establish the "pass through" mode do the following:

1. Install the DEFPA FDDI adapter to some spare PCI slot of the host system.  
 Note that PCIe and PCI-X are not supported by the DEFPA FDDI adapter.
2. Boot a Windows operating system
3. Install the DEFPA FDDI adapter driver from the following directory: "C:\Program Files\CHARON\Drivers\DEFPA\_X.X.X.XXXXX" by choosing the "Install from a list or specific location (Advanced)" option and then selecting the "defpa\_ppt\_amd64.inf" file.
4. Reboot the host

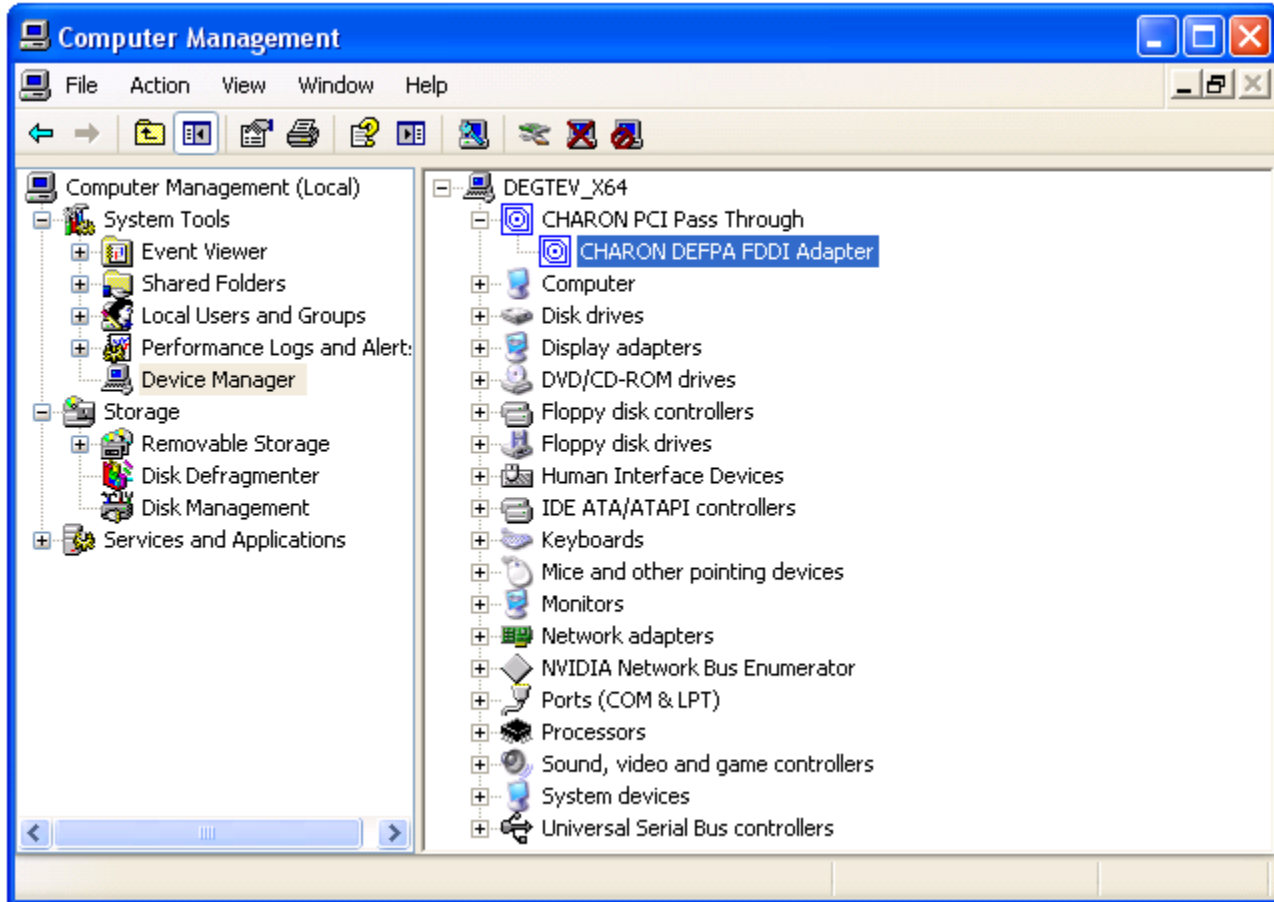
Now it is possible to collect the parameters for CHARON-AXP mapping to the DEFPA FDDI host adapter.



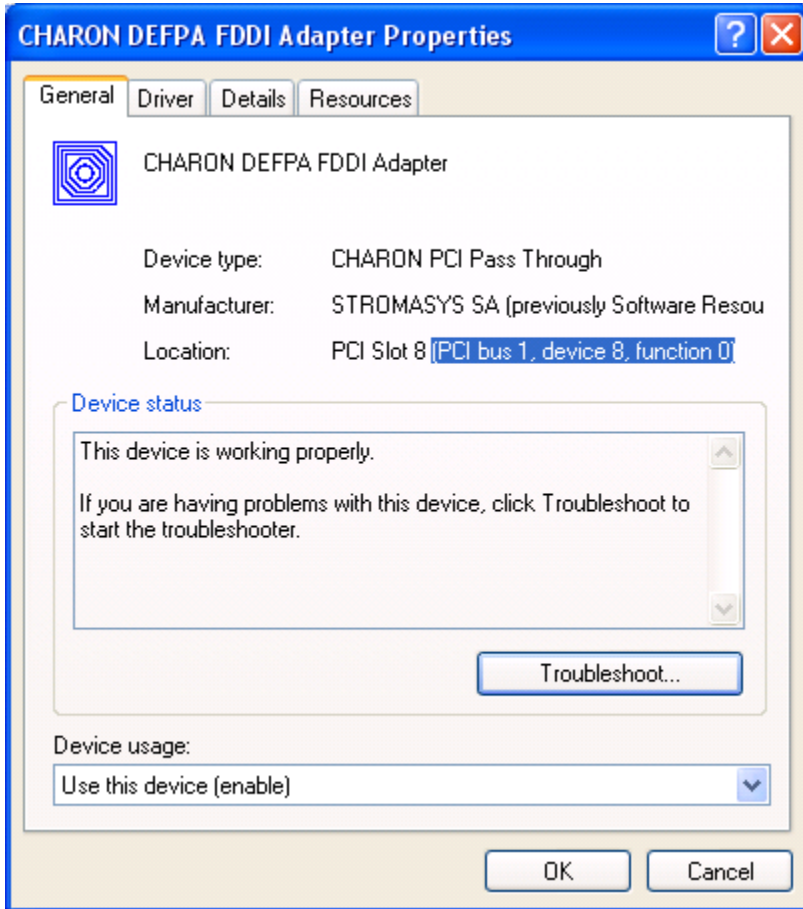
Open "Computer Management" application and select "Device Manager":



On the right panel select the installed DEFPA adapter:



Open its properties sheet by double-clicking on the selected adapter:



The "Location:" on the above picture gives X, Y, and Z for the "host\_bus\_location" parameter, for example:

```
set FDDI host_bus_location = "PCI bus 1, device 9, function 0"
```

**!** Non-US-EN installations of Windows may present "Location:" string in local language, but "host\_bus\_location" parameter requires English notation, so the words "PCI", "bus", "device", and "function" must be specified in English.

[Back to Table of Contents](#)

## Serial lines support via emulated PBXDA-xx family PCI controllers in "pass through" mode

Optionally it is possible to configure the following models of PBXDA-xx family controllers mapped to specific models of the physical DIGI serial lines adapters installed on CHARON-AXP host:

DEC PBXDA-xx adapter	Name of the device to map to	Controller	Vendor ID	Device ID
PBXDA-BA	DIGI AccelePort 4r 920	ASIC PCI	114Fh	0026h
PBXDA-BB	DIGI AccelePort 8r 920	ASIC PCI	114Fh	0027h
PBXDA-AC	DIGI AccelePort Xem	ASIC PCI	114Fh	0004h
PBXDA-AC	DIGI AccelePort Xem	ASIC PCI	114Fh	0008h

Below is an example of mapping to a physical DIGI adapter installed on the host:

```
load digi PBXDA host_bus_location="PCI bus 3,device 1,function 0"
set PBXDA bus=pci_1 device=1 function=0 irq=24 irq_bus=isa
```

Pay attention to proper placement of the emulated PBXDA-xx adapter on the virtual HP Alpha PCI bus (it is controlled by "bus", "device", "function", "irq" and "irq\_bus" parameters). Refer to the [this chapter](#) of this Guide for more information.

The "host\_bus\_location" parameter addresses the host DIGI adapter in the following way:

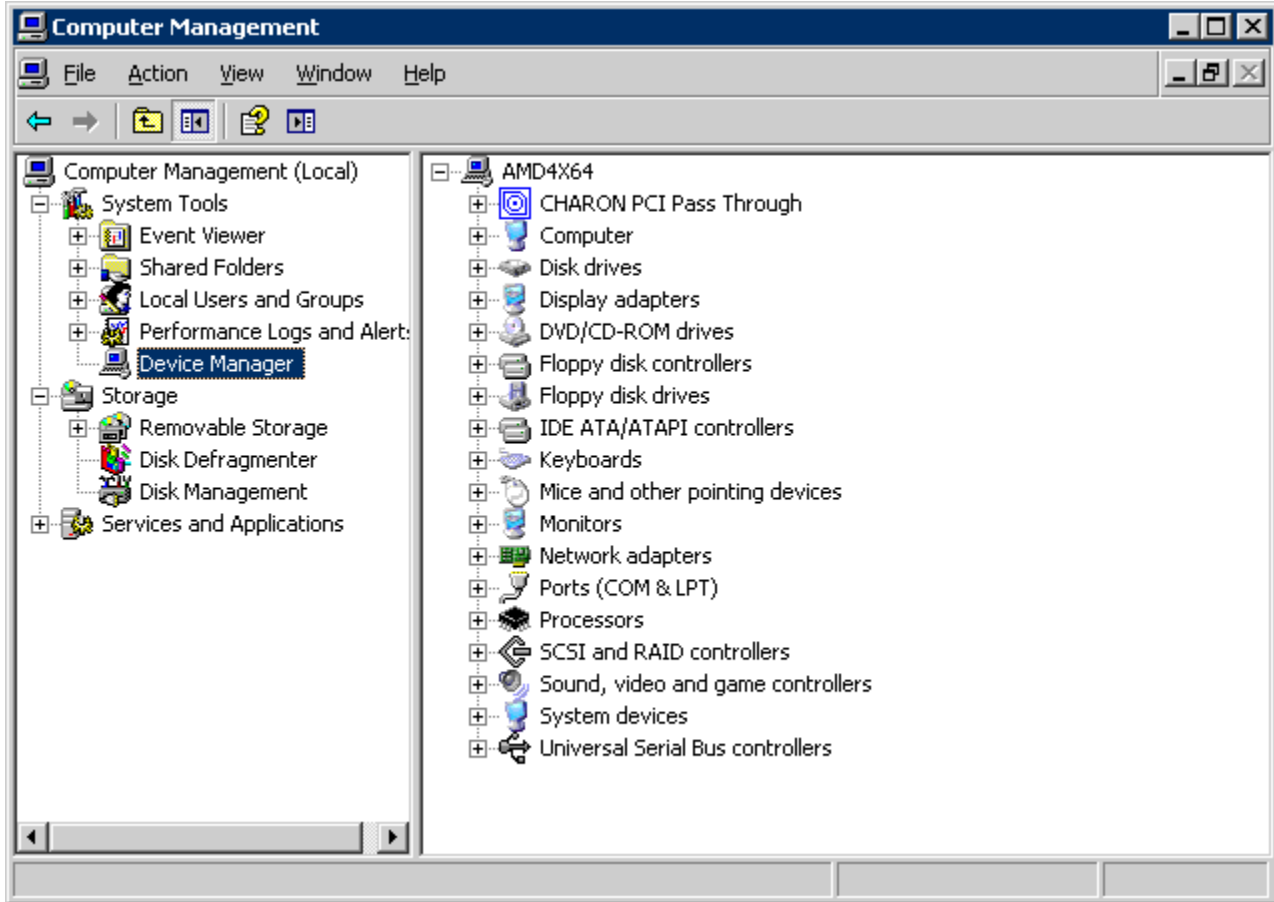
Parameters	Description
"bus X"	PCI bus number of the host DIGI adapter
"device Y"	PCI bus device number of the host DIGI adapter
"function Z"	The "function" parameter of the the host DIGI adapter

To establish the "pass through" mode do the following:

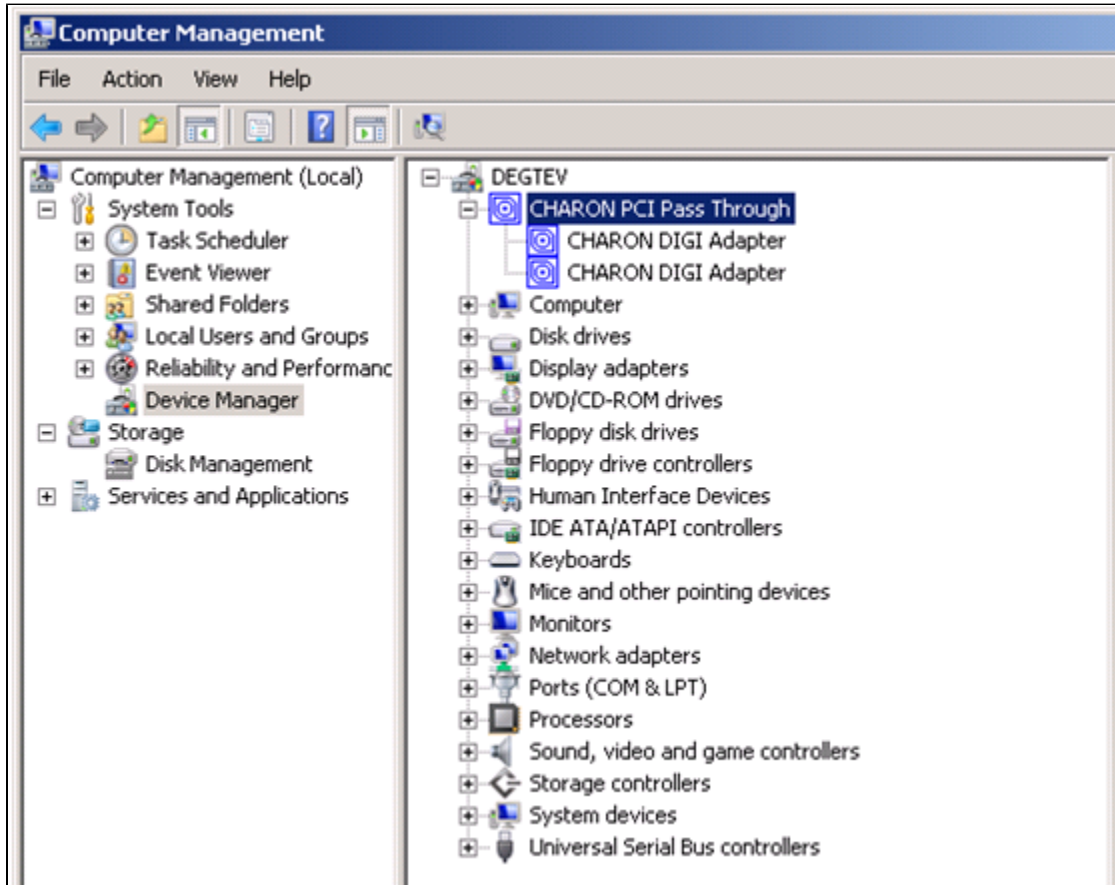
1. Install the DIGI adapter of the required type to some spare PCI/PCI-X/PCIe slot of the host system.
2. Boot a Windows operating system
3. Install the DIGI adapter driver from the following directory: "C:\Program Files\CHARON\Drivers\DIGI\_X.X.X.XXXXX" by choosing "Install from a list or specific location (Advanced)" option and then selecting the "digi\_ppt\_amd64.inf" file.
4. Reboot the host

Now it is possible to collect the parameters for CHARON-AXP mapping to the DIGI host adapter.

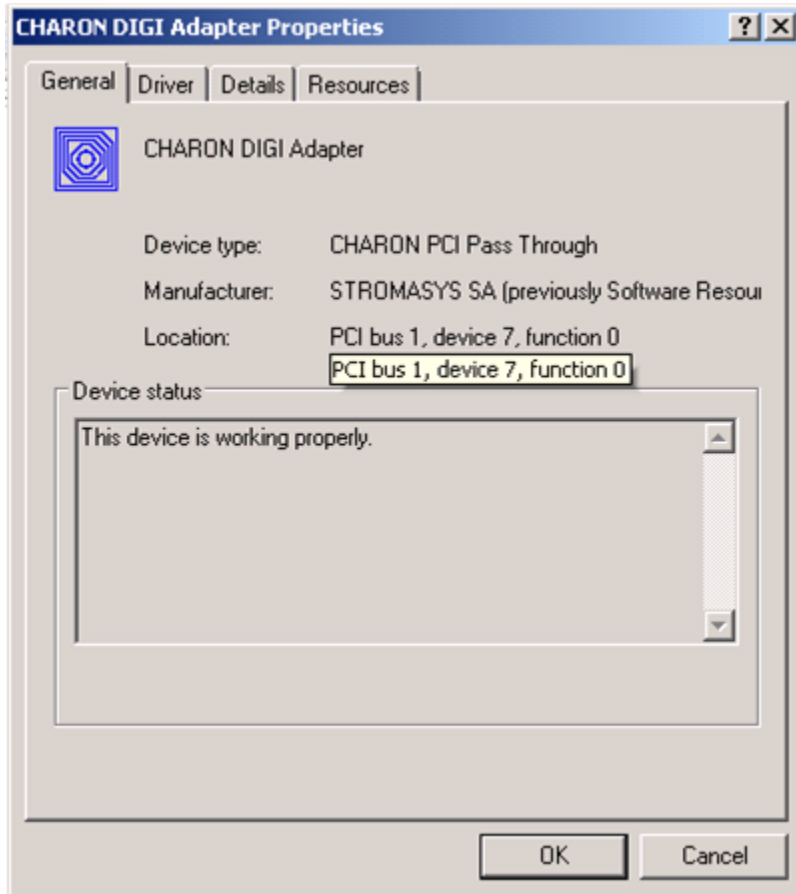
Open "Computer Management" application and select "Device Manager":



On the right panel select proper physical DIGI adapter:



Open its properties sheet by double-clicking on the selected adapter:



The "Location:" on the above picture gives X, Y, and Z for the "host\_bus\_location" parameter, for example:

```
set PBXDA host_bus_location = "PCI bus 1, device 7, function 0"
```

**!** Non-US-EN installations of Windows may present "Location:" string in local language, but "host\_bus\_location" parameter requires English notation, so the words "PCI", "bus", "device", and "function" must be specified in English.

[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 Windows

## 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 "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 "show dev /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 execlret = 3 : Enabled : Streamlined.
Config tree = None
Primary CPU = 0
HWRPB CPUs = 4
Page Size = 8192
Revision Code =
Serial Number = SN01234567
Default CPU Capabilities:
System: QUORUM RUN
Default Process Capabilities:
System: QUORUM RUN

....

>>>
```

```
>>>show mem

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

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

...

>>>
```

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

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

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

```
>>>show config

...

PCI Bus

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

Bus 00 Slot 07: Cypress PCI Peripheral Controller
Bus 00 Slot 07: Function 1: PCI IDE
Bus 00 Slot 07: Function 2: PCI IDE
Bus 00 Slot 07: PCI USB

Bus 00 Slot 08: DECchip 21052 PCI to PCI Bridge

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

>>>
```

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

- Bus number
- Slot number
- Function number

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

[Back to Table of Contents](#)

## Creation of CHARON-AXP configuration file

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

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

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

...

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

set rom system_serial_number = SN01234567

#
# Specify RAM size: 512 Mb
#

set ram size=512

#
# Map OPA0 console to "Putty" terminal emulator (included in CHARON-AXP kit)
#

load virtual_serial_line OPA0 port=10003 application="putty -load OPA0"

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

set ide container="\\.\CdRom0"

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

load DE500BA/dec21x4x EWA interface=EWA0
load packet_port/chnetwrk EWA0 interface="connection:Charon"

#
# 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] = "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[600] = "C:\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] = "C:\My disks\RZ24.vdisk"

...
```

[Back to Table of Contents](#)

## Making disk images

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

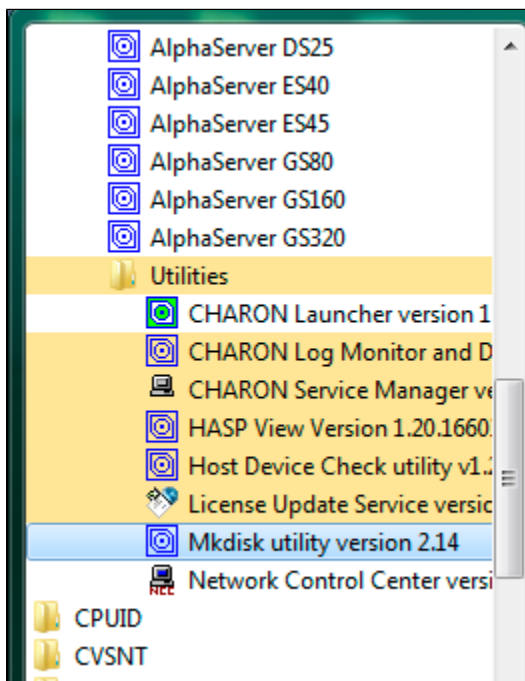
Our example creates disk images of the original physical type. In reality, this step is the best opportunity in the migration to provision bigger disks to get extra storage space.

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

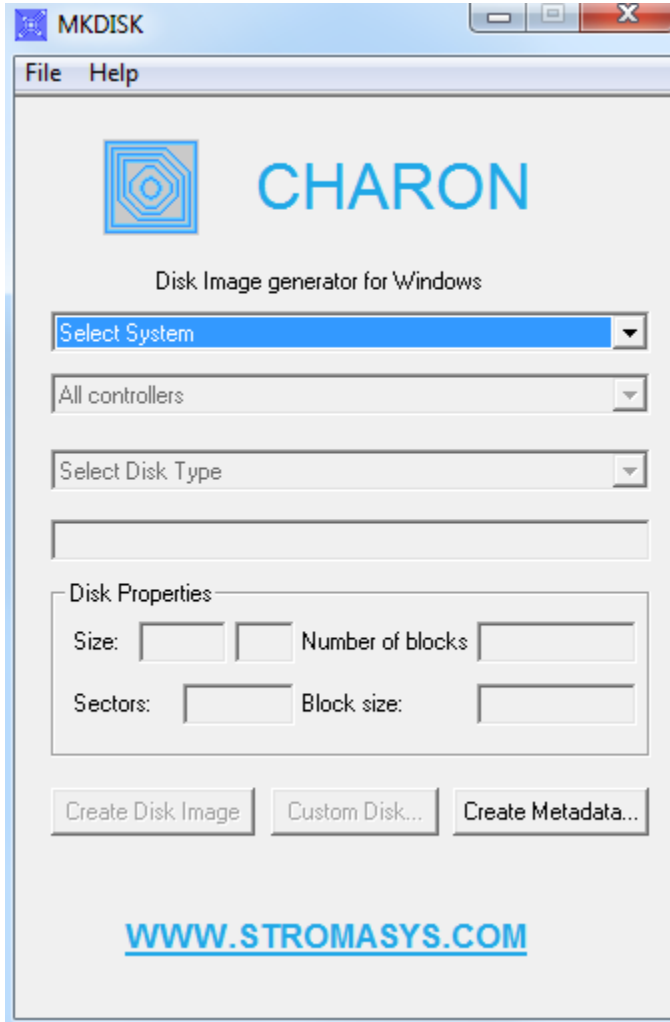
```
...> cd C:\
C:\> mkdir "My disks"
C:\> mkdir "My tapes"
```

This operation can also be done using Windows Explorer.

Start the "MkDisk" utility from the "Start" menu to create disk images:

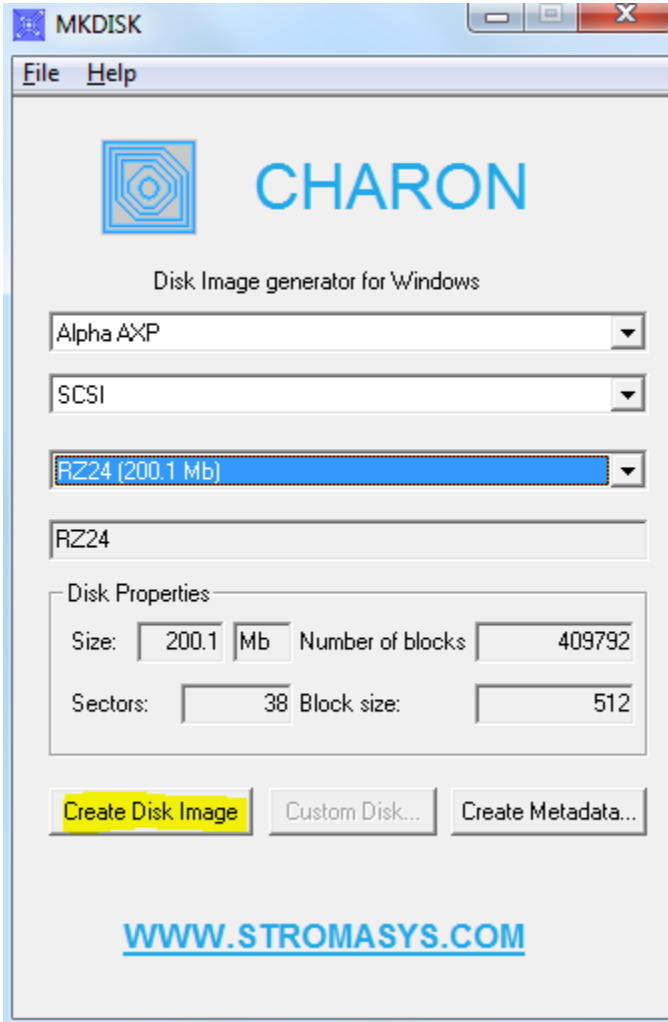


The main dialog of the utility will appear:

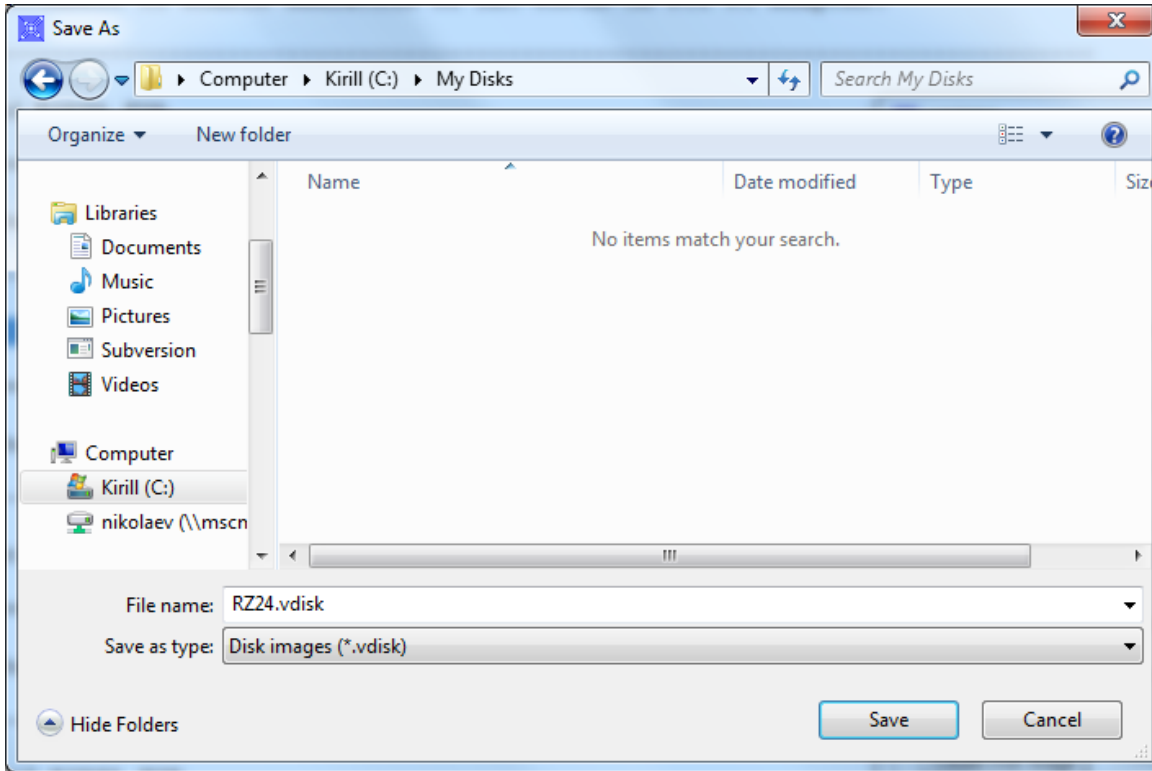


Select "Alpha AXP" in the "Select System" drop-down menu, "SCSI" in the "All Controllers" drop-down menu and "RZ24" (for example) in the "Select Disk Type" drop-down menu.

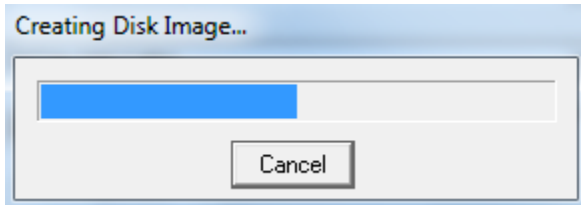
Press then the "Create Disk Image" button:



The utility will ask you to specify the output file name. Select the "C:\My disks" directory created just before (or create it directly in the dialog) and press "Save" button:



The "MkDisk" utility will create the requested disk image:



Repeat this sequence for the disks "RZ28.vdisk", "RZ22.vdisk" and "RZ23.vdisk". Place them in the same folder.

[Back to Table of Contents](#)



## Installation of HP Alpha operating system

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

The example configures the CHARON-AXP AlphaServer ES40 system for installation of HP OpenVMS from a distribution CD-ROM (usually it is "\\.\CdRom0" 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="\\.\CdRom0"
```

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

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

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

```
...>es40.exe migration.cfg

CHARON-AXP/ES40 for Windows x64 (AlphaServer ES40 6/667), Version 4.6.16603
(C) 2009-2014 STROMASYS SA.
All rights reserved.

P00>>>boot dqa0
```

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

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

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

[Back to Table of Contents](#)

## Making remote backups

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

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

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

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

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

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

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

[Back to Table of Contents](#)

## Restore backups to CHARON-AXP disks


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

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

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

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

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

The easiest way is to boot the CHARON-AXP virtualized system on the restored system disk with the network Disabled and configure new addresses, as needed.  The NIC can be disabled with a "connection:disabled" statement in the CHARON configuration file.

Then Enable the network and reboot.

[Back to Table of Contents](#)

## Alternative ways of data transfer

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

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

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

[Back to Table of Contents](#)

# CHARON-AXP for Windows licensing

## Table of Contents

- General description
- Parameters defined by CHARON-AXP license
- CHARON-AXP technical licensing models
  - Regular Sentinel HASP keys
  - Network Sentinel HASP keys
  - Software licenses
- Multiple licenses configuration and backup license
- License installation
  - Installation from scratch
  - Replacement of currently installed Sentinel run-time
  - Installation and update of CHARON-AXP Software License or HASP dongle License
  - Sentinel HL "driverless" mode
    - Now to define mode of Sentinel HL license key
    - Switching Sentinel HL to "driverless" mode
    - Using Sentinel HL in "driverless" mode
- License management
  - Sentinel Admin Control Center
    - General Description
    - Disable remote keys access
  - License management utility
- Transferring and removing CHARON-AXP software licenses
  - Software Licenses Transfer
  - Software License Removal
  - Cloned Software License Removal
- License Deinstallation
- Special "backup" license keys

## General description

**The CHARON-AXP products are protected by licenses issued on a customer basis by STROMASYS. The CHARON-AXP license defines all the specifics of a particular CHARON-AXP distribution and its usage.**

The license is implemented in the form of a hardware dongle (a Sentinel HASP key) or a software license. Please be careful with your license key, in case of loss or damage, the CHARON-AXP instances will not run or start until the license key is replaced. For redundancy, STROMASYS recommends to use 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 the main license dongle is no longer available.

The CHARON-AXP license being checked upon the start of each instance of CHARON-AXP and at a specified interval (defined by the license) during the emulated system execution (default is 1 hour). If CHARON-AXP detects the absence (or malfunction) of the license key or software license, CHARON-AXP will try to use a backup license (if specified in the configuration file). If the license is not available or not specified, CHARON-AXP displays a warning message in the log file requesting the license key reconnection or software license reactivation. If the license is not reconnected or reactivated within a given period of time (the check interval), CHARON-AXP exits.

**i** The CHARON-AXP main license is time restricted or unlimited, the backup license is limited by the number of executions (1 execution = 1 interval check)

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

The present CHARON-AXP implementation requires that the expired license be removed to allow the running CHARON-AXP instance to switch to a valid one.

The CHARON-AXP software license is not distributed for Proof-of-Concept and evaluation installations. Only hardware dongles could be used.

It is important to keep the HASP license keys connected to a computer powered on, even if CHARON-AXP is not running, because the keys contain a built-in battery that needs to be charged. If the battery is completely discharged, the license key can be irreparably damaged.

Update of the CHARON-AXP license can be performed on the fly without stopping CHARON-AXP. The expiration date and execution counter can be updated, however the CHARON-AXP virtual hardware configuration should remain unchanged. At the next license check, CHARON-AXP will use the updated license.

CHARON supports the "driverless" mode of the new generation Sentinel protection keys ("Sentinel HL"). Note that installation of Sentinel drivers is still required if the new generation HASP network key is going to be used to serve CHARON instances running on other hosts.

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 the 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 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 CHARON-AXP (might be required for synchronization with legacy hardware systems)</li> <li>Parameter that enables the product to support additional serial lines through an option board from a company such as DIGI</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 technical licensing models

CHARON-AXP licensing models are divided into 3 groups:

### Regular Sentinel HASP keys

This is the 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 using CHARON-AXP 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 part of the CHARON-AXP kit. Please see the "License installation" section of this chapter for details.

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

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

## Software licenses

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

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

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

Software Licenses are highly dependent on hardware configuration of CHARON host. Do not change hardware configuration since it leads to disabling of installed Software License!

If CHARON host has to be upgraded use the following procedure:

1. [Transfer Software License](#) to some other host.
2. Upgrade CHARON host.
3. [Transfer Software License](#) back to CHARON host.

[Back to Table of Contents](#)

## Multiple licenses configuration and backup license

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

The "HASP View" utility displays a range of available licenses, but note that by default CHARON-AXP may use any of them as the "active" one - unless it is directly specified by the "license\_key\_id" parameter (see below). The utility provides the license numbers and ID / IP addresses of the hosts where the licenses are installed.

CHARON-AXP **cannot**:

- check all the available license keys / software licenses,
- choose one,
- automatically switch from one key to another (except the cases if a 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><u>Example:</u></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><u>Example:</u></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>

This parameter also specifies the license which will be used as backup if the active license expires.

[Back to Table of Contents](#)

## License installation

### Installation from scratch

**Before installation of license make sure that the Sentinel run-time (driver) to be installed is collected from your CHARON kit or provided by STROMASYS on request for your particular product. Do not update the Sentinel run-time (driver) from online or any other sources, including the Sentinel web site.**

**In case of several CHARON kits containing different versions of Sentinel run-time (driver) the last one (having more recent version) must be installed. CHARON installation performs this operation automatically.**

At the moment CHARON supports Sentinel HASP keys, Sentinel HL and Sentinel Software Licenses (SL).

Installation of Sentinel run-time (driver) described below is needed for Sentinel HASP keys (both regular and network), Sentinel HL keys (both regular and network) without "driverless" mode enabled, Sentinel HL Network keys (red dongles) - if they are going to be used network wide - and Sentinel Software Licenses (SL).

Please skip installation of Sentinel run-time (driver) if you use locally connected Sentinel HL keys in "driverless" mode or going to use Sentinel HL Network key in "driverless" mode from some other host on local network. Just connect the license key to the host and proceed with collecting "\*.c2v" file (see below) if the license inside your Sentinel HL key must be updated.

Installation of a CHARON-AXP regular or network licenses consists of:

1. Installation of the Sentinel run-time environment on the CHARON-AXP host (regular and network keys) or on the host that will distribute CHARON-AXP licenses over a local network segment (network key only). The Sentinel software is installed automatically by CHARON-AXP for Windows.
2. Physical connection of the HASP license dongle to the CHARON-AXP host or to the host distributing the CHARON-AXP license over the local network segment.
3. Collecting system fingerprint (\*.c2v file), sending it to STROMASYS and applying update (\*.v2c file) in case of software license (and in case of updating HASP dongle license). See the details below.

When manual installation of Sentinel run-time is required (in the case of the network license server that does not have CHARON-AXP installed):

1. Login as Administrator
2. Open the CHARON-AXP kit folder
3. Switch to the "hasp\_install" subfolder
4. Unzip the archive located in this folder
5. Open "cmd.exe" and switch to the folder where the files were unzipped
6. Issue:

```
...> haspdinst.exe -fr -kp -nomsg
...> haspdinst.exe -install -cm
```

Sentinel HL keys can be provided both in "driverless" and regular modes. To switch the Sentinel HL key to "driverless" mode it is required to install Sentinel run-time first as described above and apply a special \*.v2c file that switches the key to "driverless" mode. Once it is done the Sentinel run-time can be deinstalled.

Note that the following operations have to be performed on installation phase for network licenses:

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

Please consult with your Windows 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 "Sentinel HASP License Manager" (hasplms.exe) service (it will be installed by CHARON-AXP).

[Back to Table of Contents](#)

## Replacement of currently installed Sentinel run-time

Replacement of currently installed Sentinel Run-time can be needed in case of installation of specific run-time provided by STROMASYS. Below please find step-by-step instructions on the run-time replacement:

Remove the current version of Sentinel run-time:

- Login as Administrator
- Extract content of the only ZIP file resided in "hasp\_install" directory of the CHARON distribution to the same folder.
- Run "cmd.exe" from "Start" menu of host computer
- "cd" to the "hasp\_install" directory
- Issue the following command:

```
...> haspdinst.exe -fr -kp -nomsg
```

Install other Sentinel run-time:

- "cd" to the directory whether the target run-time resides
- Issue the following command:

```
...> haspdinst.exe -install -cm
```

You do not need to perform this procedure for Sentinel HL keys - local and network ones (red dongle) used as local.

[Back to Table of Contents](#)

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

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

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



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





**Options**

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

Configuration for Sentinel License Manager on KIRILL

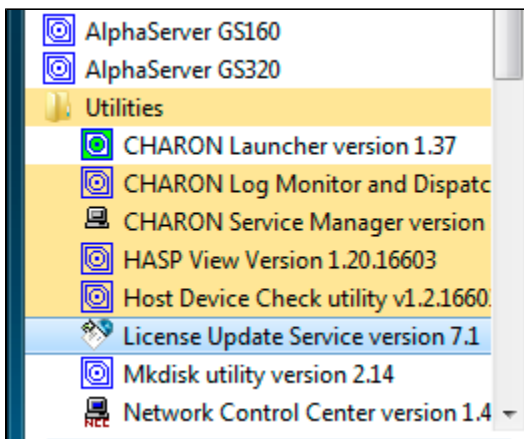
**Basic Settings** | Users | Access to Remote License Managers | Access from Remote Clients | Detachable Licenses | Network

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

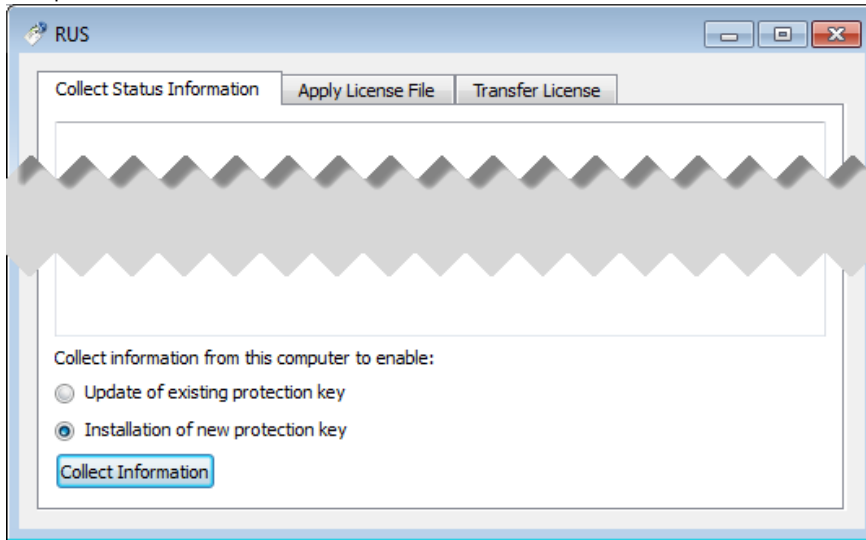
Open "License Update Service" utility:



In the utility dialog popup open the "Collect Status Information" tab, then select:

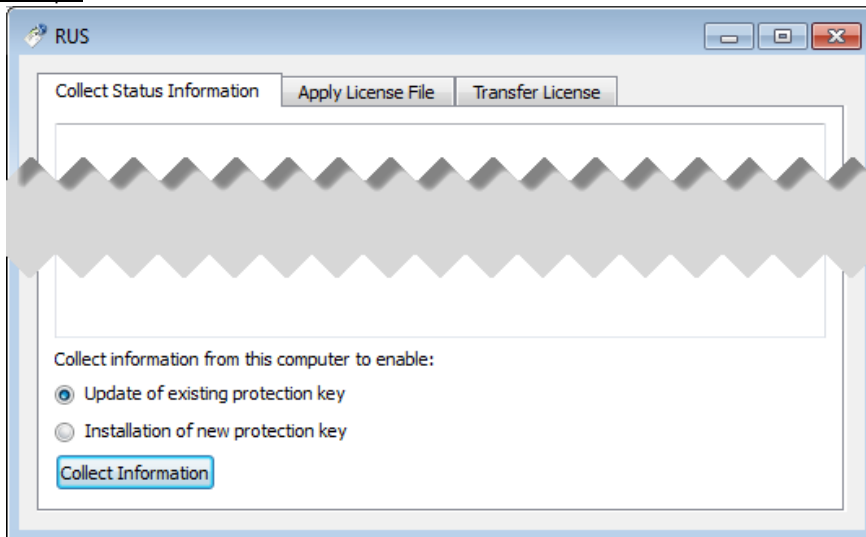
- "Installation of new protection key" (in case of Software License if no Software License has been installed on the host):

Example:



- "Update of existing protection key" (in case of HL/HASP dongle or Software License that has been already installed on the host and needs updating):

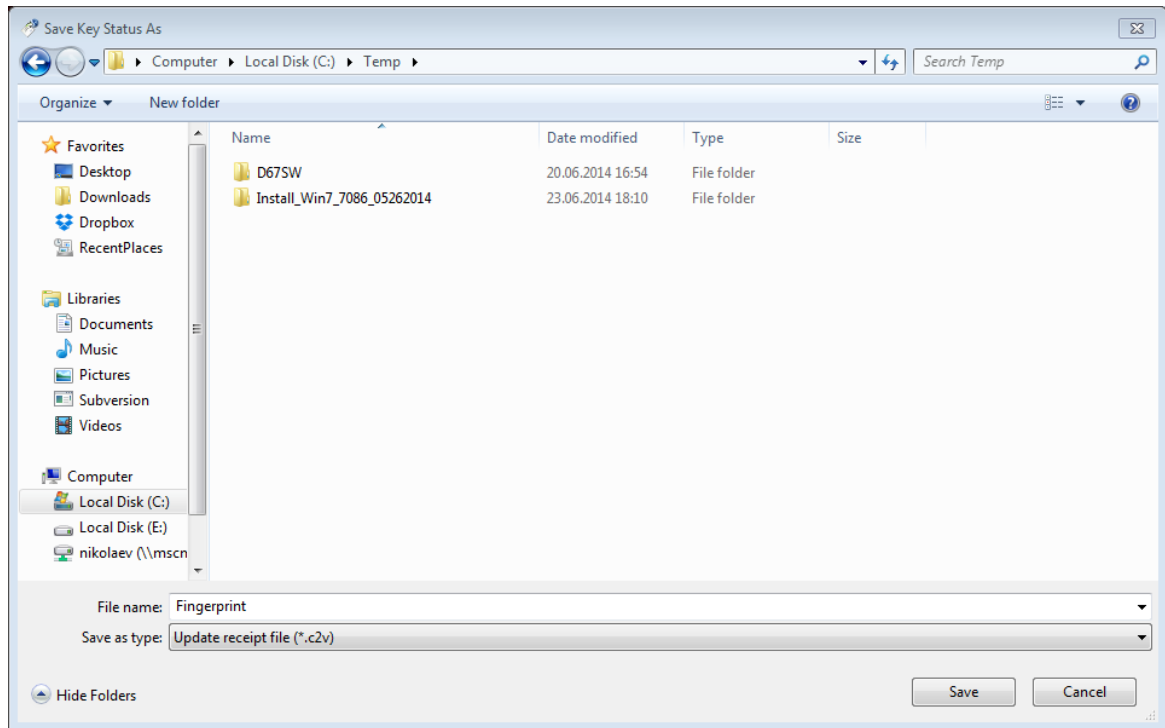
Example:



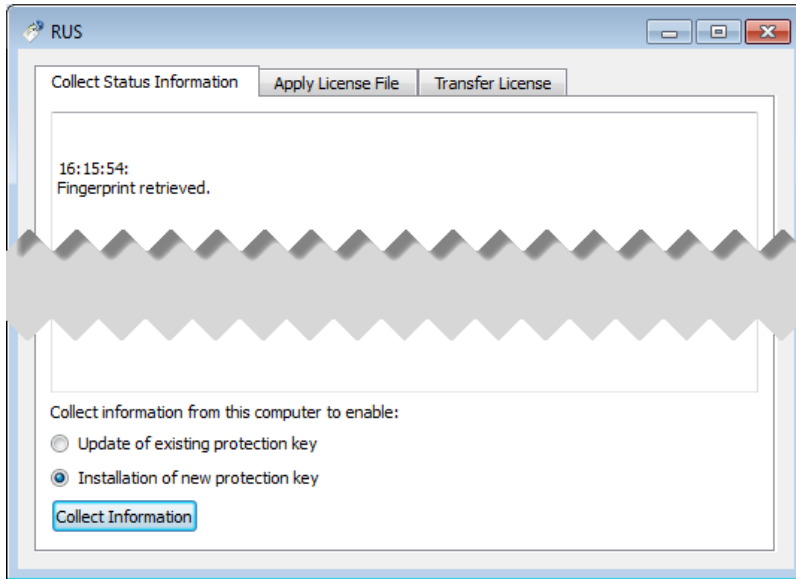
In case of HASP dongles use only "Update of existing protection key" option!

In case of Software License use "Installation of new protection key" option if host does not have Software License installed and "Update of existing protection key" option if already installed Software License has to be updated.

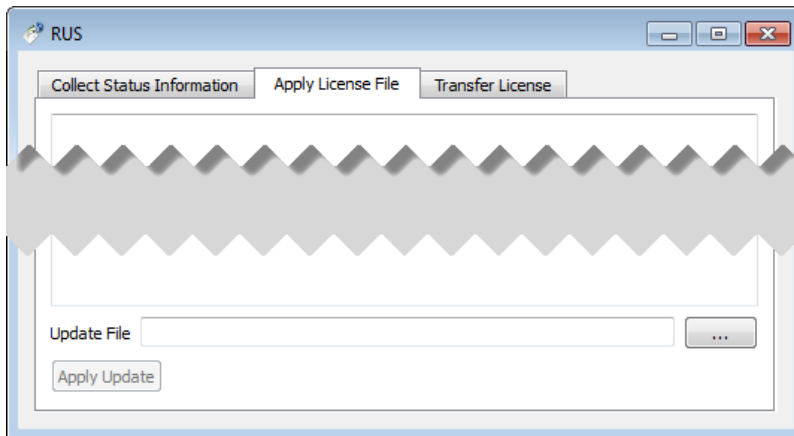
- Press "Collect Information". In the popup dialog choose the place to store the "Fingerprint.c2v" file and press "Save":



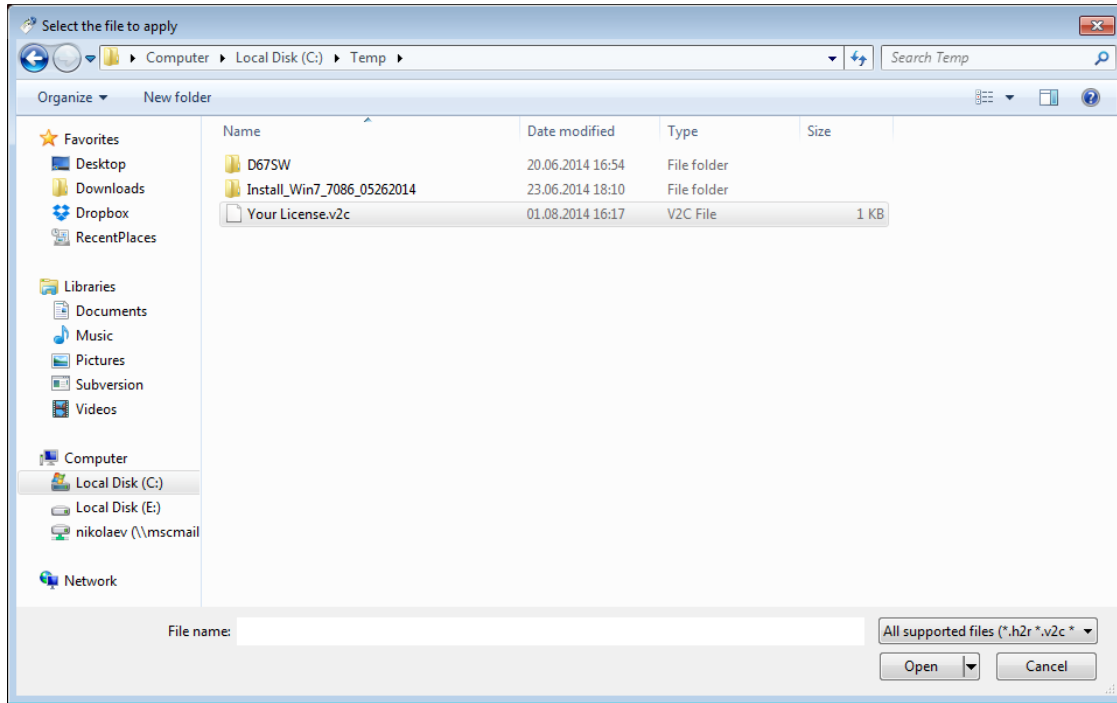
- A message should appear similar to this example, confirming the fingerprint has been collected successfully.



- Send the ".c2v" file ("*Fingerprint.c2v*" in the example above) to STOMASYS
- STOMASYS will send you a ".v2c" file in return. Put it somewhere on the CHARON-AXP host.
- Open up the "License Update Service" utility the way described above and open the "Apply License File" tab:



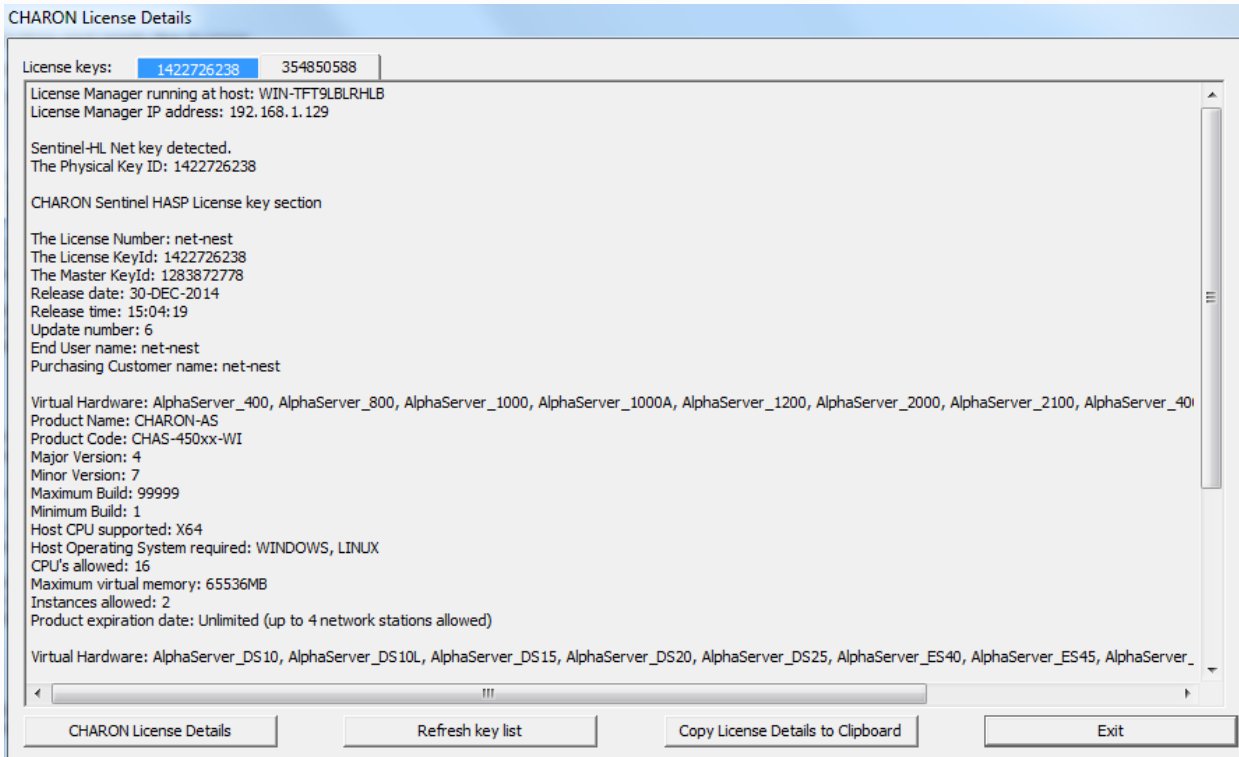
Press "..." button beside the "Update File" edit-box. In the popup select the license file received from STROMASYS:



Press the "Open" button and apply the license.

- Start any web browser on this system and go to <http://localhost:1947> to access the "Sentinel HASP Admin Control Center" (ACC).
- Ensure that the license appears in the "Sentinel Keys" menu.

Content of the installed software or HL/HASP license is not shown by the Sentinel HASP Admin Control Center. To see it run the "HASP View" utility from the CHARON-AXP "Utilities" menu (see above):



Note that the following operations have to be performed on installation phase for network-wide software licenses:

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

Please consult with your Windows 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 "Sentinel HASP License Manager" (hasplms.exe) service (it will be installed by CHARON-AXP ).

[Back to Table of Contents](#)

## Sentinel HL "driverless" mode

New license keys from Sentinel, so called "Sentinel HL" can work in 2 modes:

1. **Regular.** The key behaves as normal Sentinel HASP key.
2. **Driverless.** Sentinel HL keys do not require installation of any specific run-time or drivers since they utilize HID drivers, which are an integral part of the Windows operating system.

Sentinel HL license key in "driverless" mode has the following benefits:

- Support of a higher number of features.
- Larger on-key memory space.
- No specific drivers are needed.
- No additional support software is needed for all operations with the license keys.

There is a disadvantage of using Sentinel HL in "driverless" mode: inability to use "Sentinel Admin Control Center" (see below for details) for detailed configuration of the Sentinel HL license key attached to the host.

Sentinel HL Network key (red dongle) always requires installation of Sentinel run-time (driver) on the host it is connected to. See the sections above for details.

## Now to define mode of Sentinel HL license key

STROMASYS may provide Sentinel HL keys in regular or "driverless" mode. To define the current mode:

1. Install Sentinel run-time (drivers) as described above in "Installation from scratch" section.
2. Connect the key to the host
3. Run "HASP View" utility. It will report whether the key has "driverless" mode enabled or not.

## Switching Sentinel HL to "driverless" mode

To switch Sentinel HL from regular to "driverless" mode:

1. Install Sentinel run-time (drivers) as described above in the "Installation from scratch" section.
2. Apply a specific "\*.v2c" file provided by Stromasys as described above in the "Installation of CHARON Software License" section.
3. Remove Sentinel run-time (drivers). See the detailed description below.

## Using Sentinel HL in "driverless" mode

To use Sentinel HL key in "driverless" mode:

1. Install CHARON kit. By default it installs Sentinel run-time.
2. Reboot your host as the CHARON installation suggests.
3. Login as Administrator
4. Extract content of the only ZIP file resided in "hasp\_install" directory of the CHARON distribution to the same folder.
5. Run "cmd.exe" from "Start" menu of host computer.
6. "cd" to the "hasp\_install" directory.
7. Issue the following command: `haspdinst.exe -fr -kp`
8. Reboot your host.
9. Connect Sentinel HL key to the host. It will be correctly recognized by operating system. Once it is done the Sentinel HL key is ready to be used.

[Back to Table of Contents](#)

## License management

CHARON-AXP license management is performed by the Sentinel Admin Control Center and specific utilities.

### Sentinel Admin Control Center

#### General Description

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

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

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

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

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

For a more detailed description of Sentinel Admin Control Center, please refer to its "Help" section.

[Back to Table of Contents](#)

### Disable remote keys access

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

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



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

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



- 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
<p>Allow Access from Remote Clients <input type="checkbox"/> You may experience a delay of a few minutes before your changes will take effect.</p> <p>Access Restrictions</p> <div style="border: 1px solid gray; padding: 5px;"> <p>allow=all</p> </div> <p><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</p> <p style="text-align: right;"> <input type="button" value="Submit"/> <input type="button" value="Cancel"/> <input type="button" value="Set Defaults"/> </p>				

[Back to Table of Contents](#)

## License management utility

CHARON-AXP for Windows provides two specific utilities for license management:

- "HASP View". This utility is used to display CHARON-AXP license content.
- "License Update Service". This utility is used to collect key status information and host fingerprint (C2V) files, apply updates (".v2c" files) and perform the license transfer/removal.

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

[Back to Table of Contents](#)



## Transferring and removing CHARON-AXP software licenses

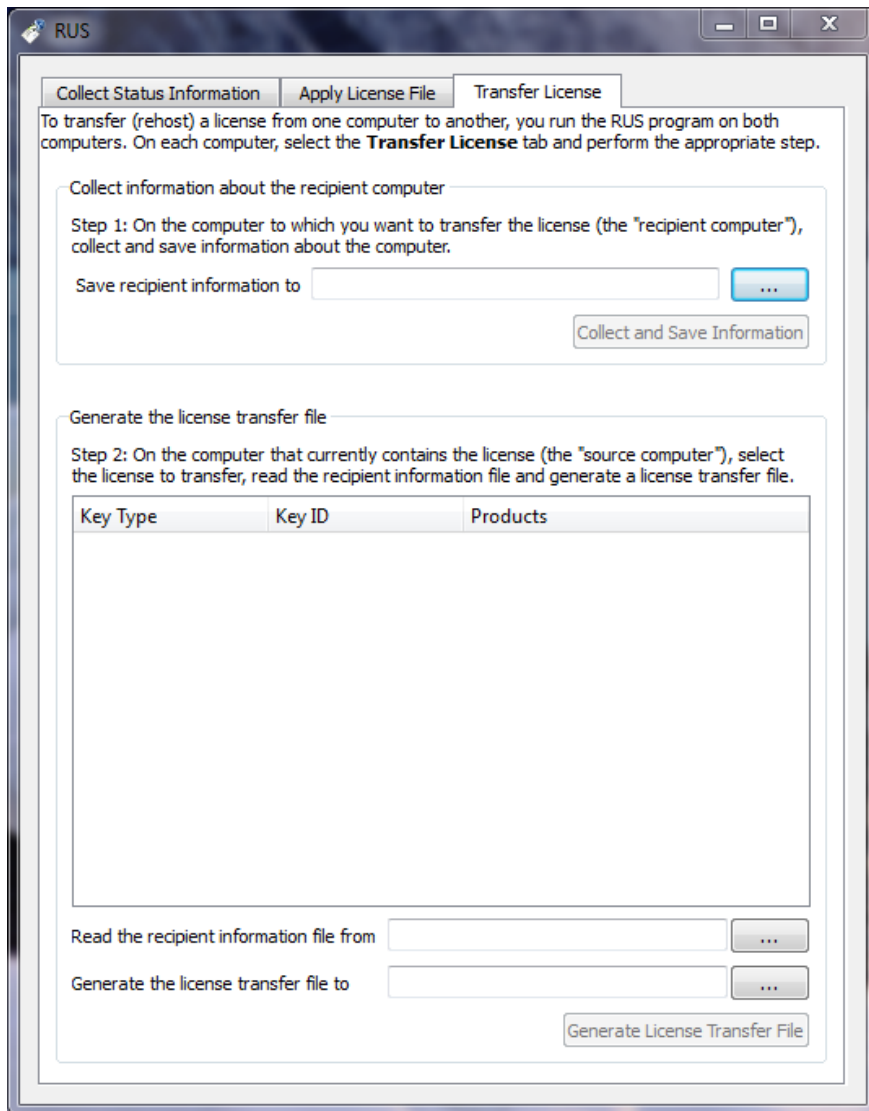
### Software Licenses Transfer

Software Licenses (SL) can be transferred from one host to another using the "License Update Service" utility.

The following example demonstrates the transfer procedure.

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

1. Collect the specific information about the "RecipientHost" to issue a transfer license: run the "License Update Service" utility on the "RecipientHost" (see above), choose the "Transfer License" tab and press the "..." button adjacent to the "Save Recipient Information" edit-box:



In the popup choose a directory and file name for the recipient information. Press the "Collect and Save Information" button to create the recipient information file.

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

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

- On "SourceHost", run "License Update Service" utility, select "Transfer License" tab and the particular license to transfer in the big edit-box. Press the "..." button adjacent to the "Read the recipient information from file" edit-box, choose the just transferred recipient file, press the "Open" button.  
Press the "..." button adjacent to the "Generate the license transfer file to" edit-box, choose the target directory and enter the desired name of the transfer file (\*.h2h) then press the "Save" button.  
Press the "Generate License Transfer File" button to create the license transfer file in the specified folder.

- Copy the resulting "\*.h2h" file to the "RecipientHost".

"\*.h2h" file is an ASCII file, so use the "ascii" option in case of FTP transfer.

- On the "RecipientHost", apply the license transfer file (\*.h2h) the same way as a regular software license (see above).
- Start any web browser on the "RecipientHost" and go to <http://localhost:1947> to access the "Sentinel HASP Admin Control Center" (ACC).
- Ensure that the license appears in the "Sentinel Keys" menu.

[Back to Table of Contents](#)

## Software License Removal

It is also possible to remove a Software License completely from a host, the license will then be stored in a specific transfer license file "\*.h2h" so it can be re-applied if needed. To remove a Software License completely from a host, follow the license transfer procedure described above. It is possible to use the fingerprint of the "SourceHost" (instead of the one from the "RecipientHost") for the transfer procedure.

[Back to Table of Contents](#)

## Cloned Software License Removal

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

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

**SafeNet Sentinel Admin Control Center**

Options: Sentinel Keys, Products, Features

Sentinel Keys Available on charontest.msc.masq

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

- Note the name of the correspondent certificate and path to the certificates base in the "Certificates" section:

**SafeNet Sentinel Admin Control Center**

Options: Sentinel Keys, Products, Features, Sessions

Certificates Available on ADANILOV: Key 387285561437702475 (Vendor: 68704)

#	Update	Status	Certificate Name	Creation Date
1	Base	OK	387285561437702475_base.v2c	Jun 9, 2015

Certificates Base Directory: C:\Program Files (x86)\Common Files\SafeNet Sentinel\Sentinel LDK\installed

- Remove the certificate file. In the example above the file to remove is "C:\Program Files (x86)\Common Files\SafeNet Sentinel\Sentinel LDK\installed\68704\387285561437702475\_base.v2c".
- Reboot CHARON host.
- Start "Sentinel HASP Admin Control Center" (ACC) again to ensure that the SL has been removed.

[Back to Table of Contents](#)

## License Deinstallation

Deinstallation is needed only in case if Sentinel run-time has been installed. If Sentinel HL in "driverless" mode has been used (see above) no specific deinstallation is required. Just disconnect the key from CHARON host.

To completely remove a CHARON-AXP license from a host, it is enough to remove the Sentinel run-time driver using the following procedure:

- Login as Administrator
- Open the CHARON-AXP kit folder
- Switch to the "hasp\_install" subfolder
- Unzip the archive located in this folder
- Open "cmd.exe" and switch to the folder where the files were unzipped
- Issue:

```
...> haspdinst.exe -fr -kp -nomsg
```

Remove the license dongle. Note that the Sentinel run-time driver is deinstalled automatically on CHARON-AXP with a complete deinstallation.

[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 is the time allotted to get a replacement 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 Windows utilities

## Table of Contents

- General description
- Details of CHARON-AXP Utilities

## General description

CHARON-AXP provides the following set of utilities:

Utility	Description
CHARON Launcher	Used to start CHARON and install it as a Windows service.
CHARON Service Manager	Used to manage CHARON services.
Host Device Check	Used to review system resources that can be mapped to CHARON.
Network Control Center	Used to configure a CHARON network.
MkDisk	GUI-based utility used to create custom or standard CHARON virtual disk containers.
mkdiskcmd	Command line utility used to create custom or standard CHARON virtual disk containers. This utility also may be used to transfer virtual disks of one type to virtual disks of another type.
HASP View	Used to display the CHARON license content.
License Update Service	Used to manage CHARON licenses, collect the host system fingerprint and to transfer software licenses from one host to another.
mtd	Used to create CHARON tape images from physical tapes and to write tape images back to physical tapes.
CHARON Log Monitor and Dispatcher	Used to trace CHARON log files and run a specific program on certain conditions.
IDLE utility	Used to improve and manage CHARON performance.
HOSTPrint	Used to print CHARON output to Windows printers.

[Back to Table of Contents](#)

## Details of CHARON-AXP Utilities

- CHARON Launcher
- CHARON Service Manager
- Host Device Check
- Network Control Center
- MkDisk
- MkDskCmd
- HASP View
- License Update Service
- mtd
- CHARON Log Monitor and Dispatcher
- IDLE Utility
- HOSTPrint

# CHARON Launcher

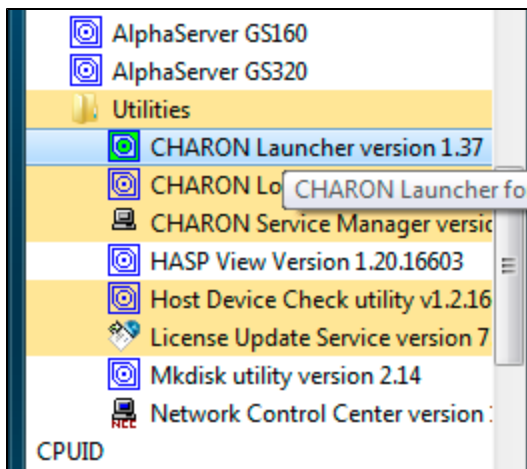
## Table of Contents

- General Description
- Running CHARON-AXP
- Installation of CHARON-AXP as a service
- Modification of CHARON-AXP service
- Sending problem reports

[Back to Table of Contents](#)

## General Description

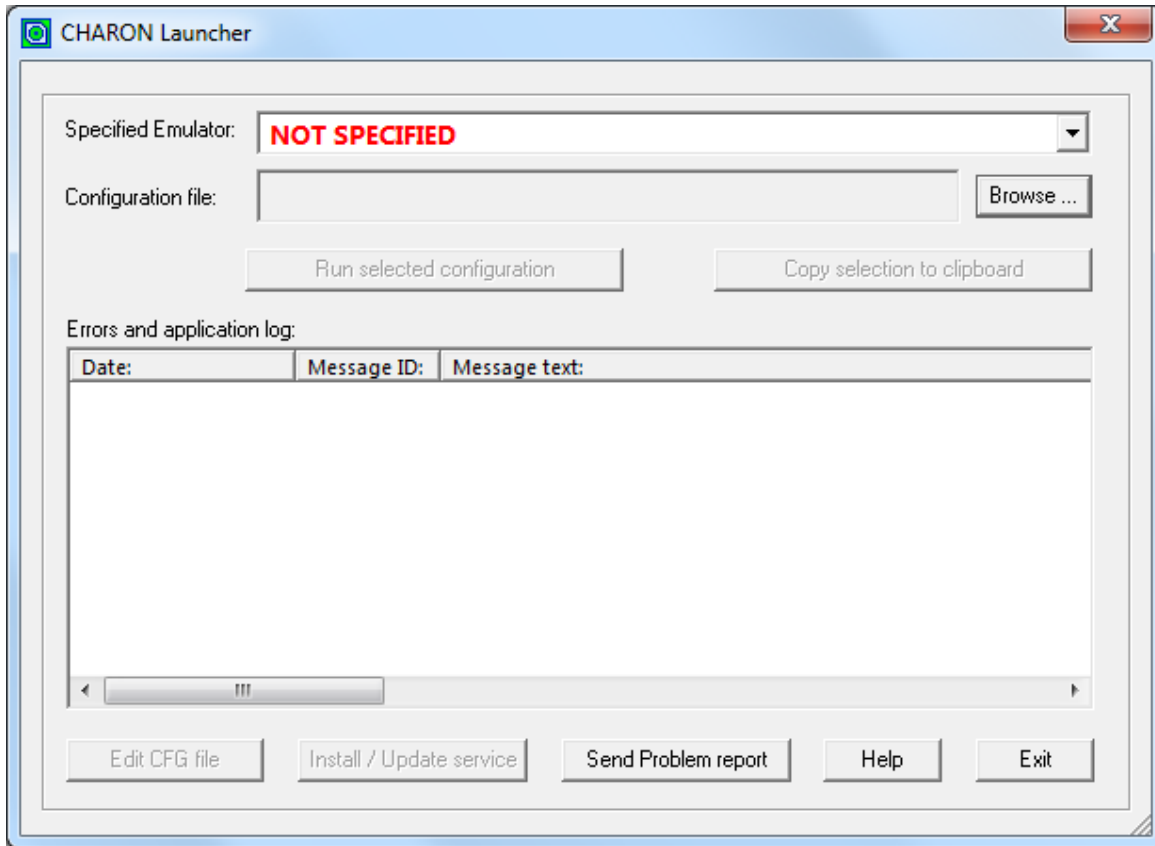
The CHARON Launcher is used to start CHARON-AXP and install it as a Windows service. It provides a more convenient way to run CHARON-AXP to debug your specific configuration, because it is able to display CHARON-AXP log updates in real time. To start the CHARON Launcher click on its icon in the "Start" menu:



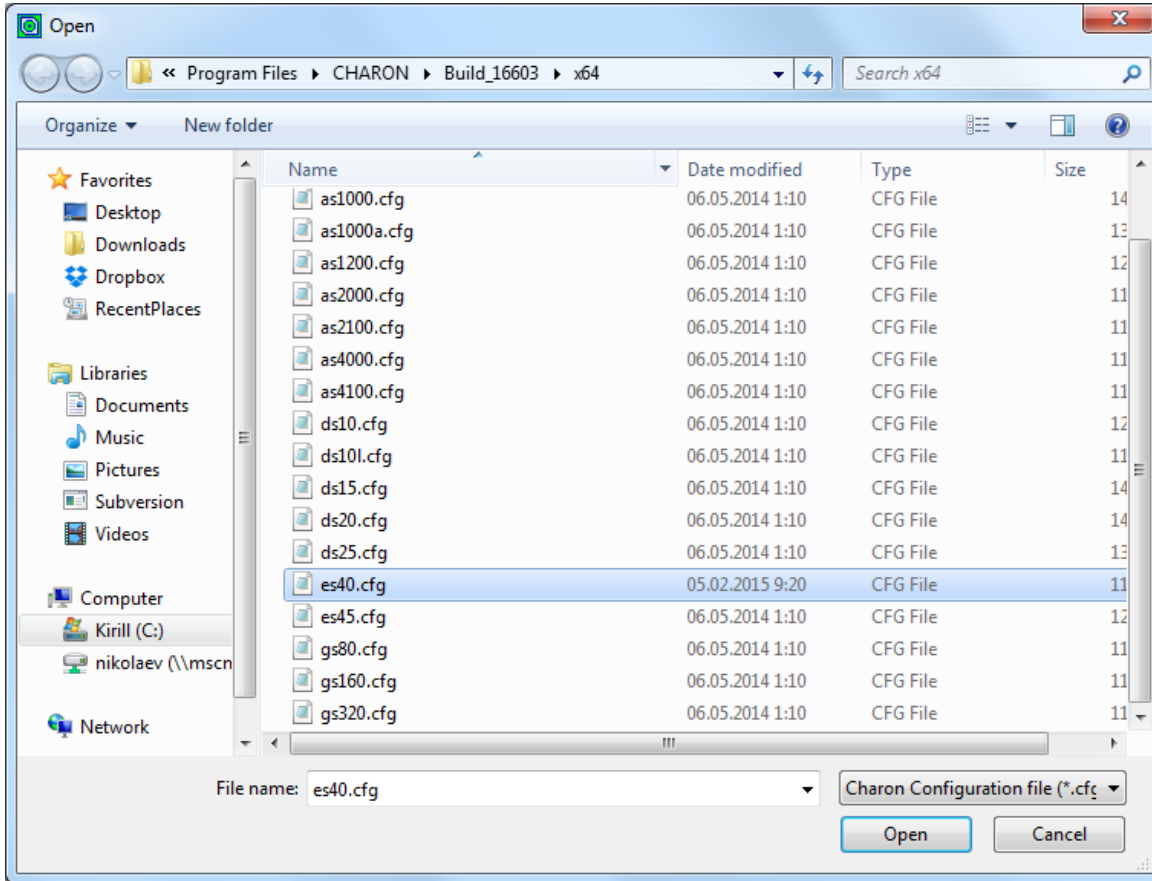
[Back to Table of Contents](#)

## Running CHARON-AXP

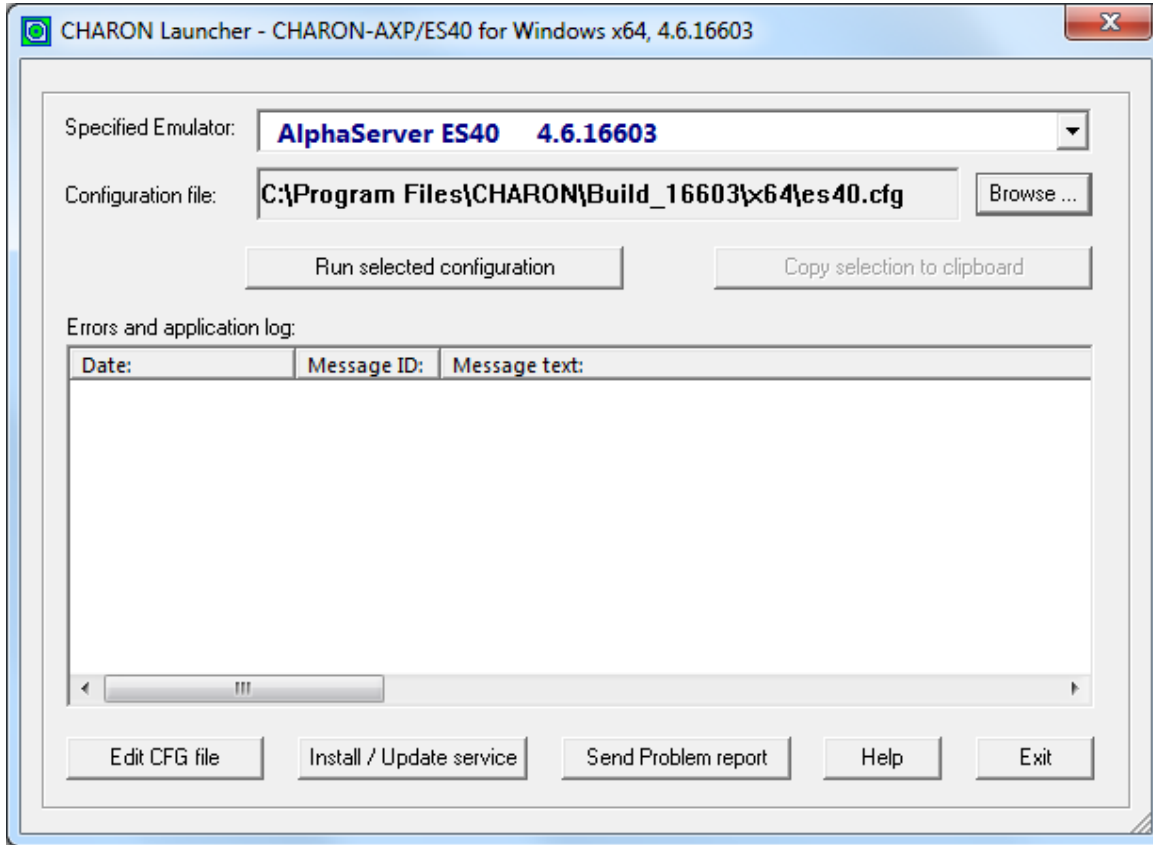
In the Launcher main interface press the "Browse" button to select a CHARON-AXP configuration:



Select the desired configuration file:



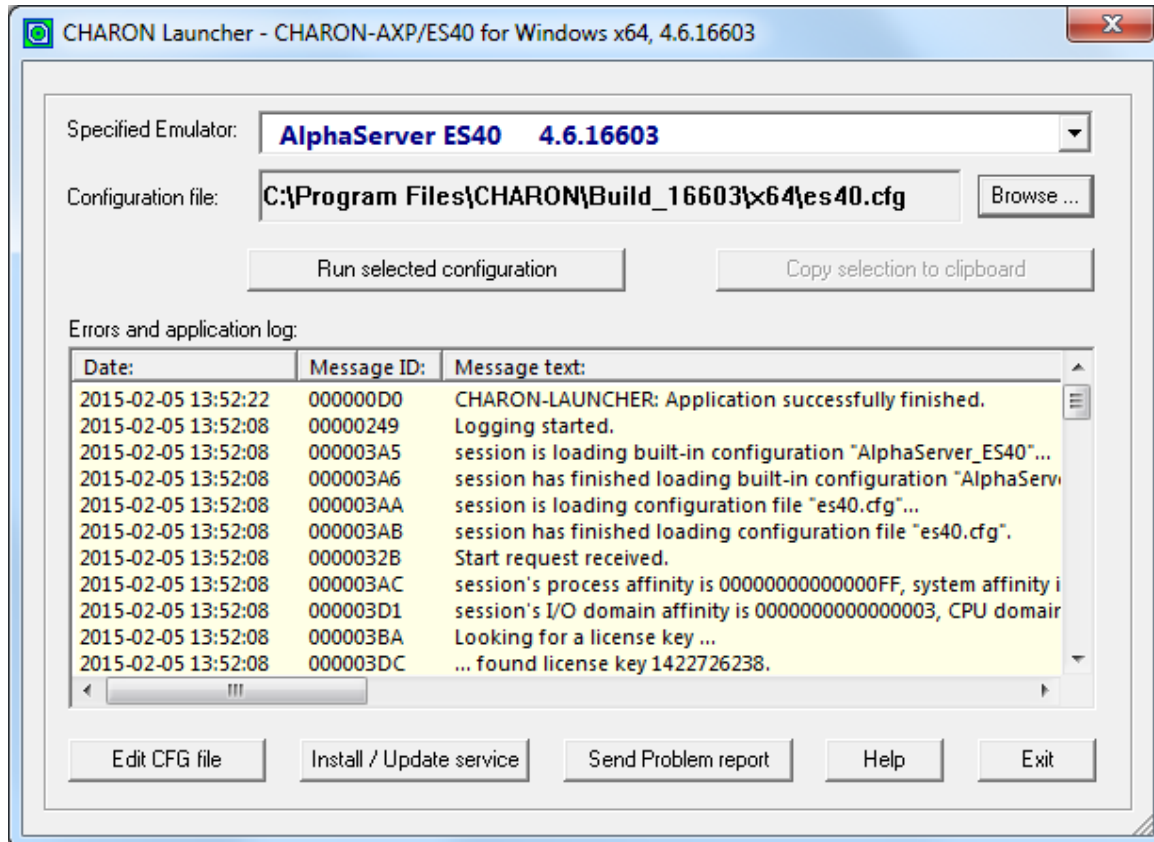
The Launcher will display the CHARON-AXP model specified in the selected configuration file:



Now it is possible to run CHARON-AXP. Press the "Run selected configuration" button.



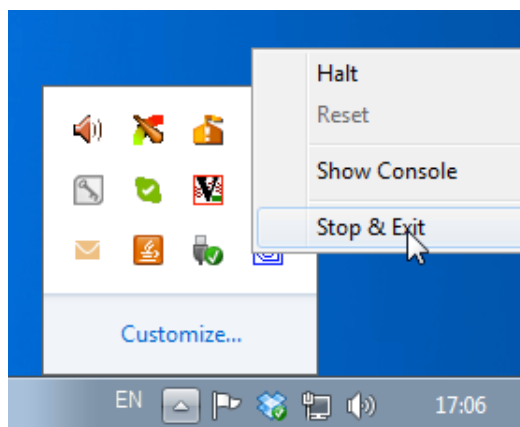
When CHARON-AXP is executed the Launcher will display CHARON-AXP log file:



It is possible to scroll the log file to analyze possible problems.

Note that the Launcher is not able to stop CHARON-AXP execution. To do that use the following methods:

- **Configuration with no changes to the template configuration file**
  - Right click on CHARON-AXP icon in the system tray menu, choose "Stop & Exit":




- Using this menu, it is also possible to halt, reset or show the virtual HP Alpha console, if it has been closed.
- **Configuration file updated to enable "F6" button in configuration file to trigger exit from CHARON:**

```
set OPA0 stop_on = F6
```

- To exit, press "F6" in the console window
- Use CHARON-AXP icon tray menu as shown above
- **"power off" command in CHARON-AXP ">>>" console**

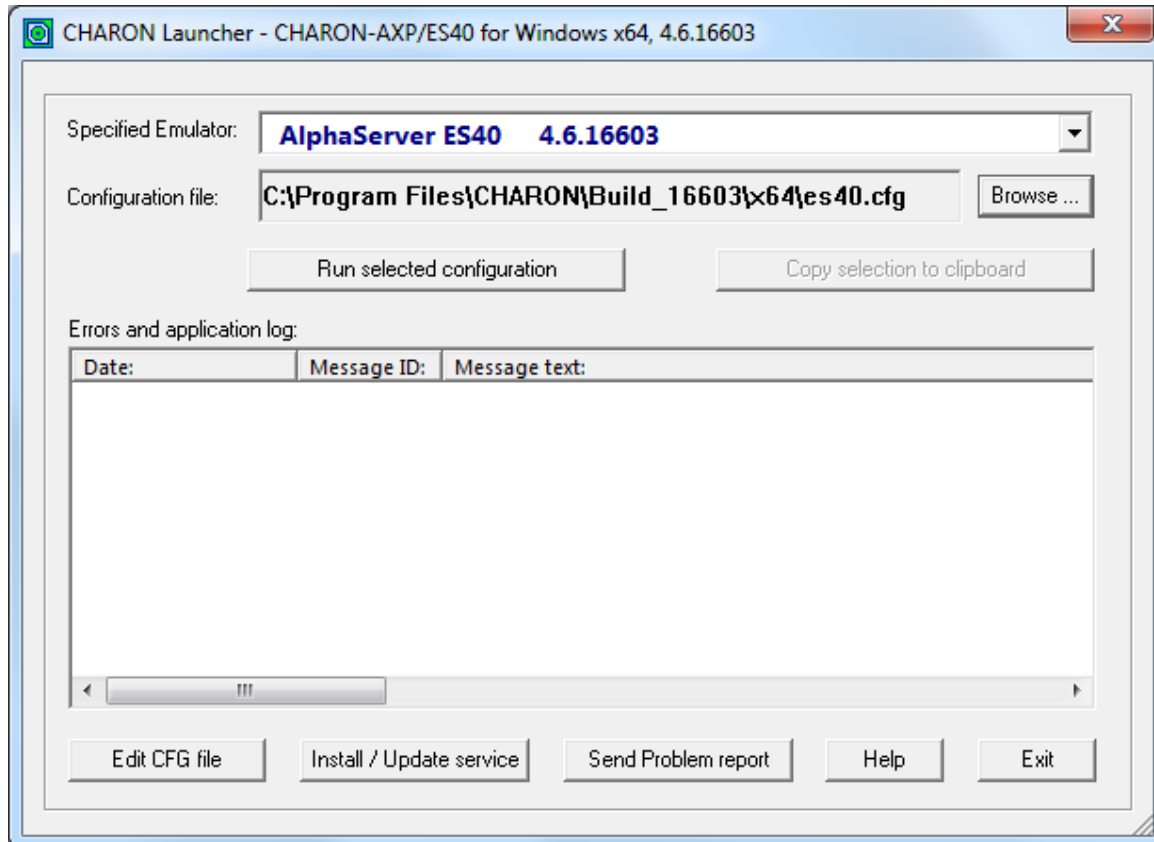
```
>>> power off
```

 Before stopping the CHARON-AXP instance, shutdown the operating system running in CHARON-AXP.

[Back to Table of Contents](#)

## Installation of CHARON-AXP as a service

Load the CHARON-AXP configuration file as it is described above:



Press the "Edit CFG file" button and edit (with a standard "Notepad" program) the chosen configuration file as described previously and make sure that the following pre-requisites are met:

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

```
...
set session log="C:\My Services\my_es40.log"
set toy container="C:\My Services\my_es40.dat"

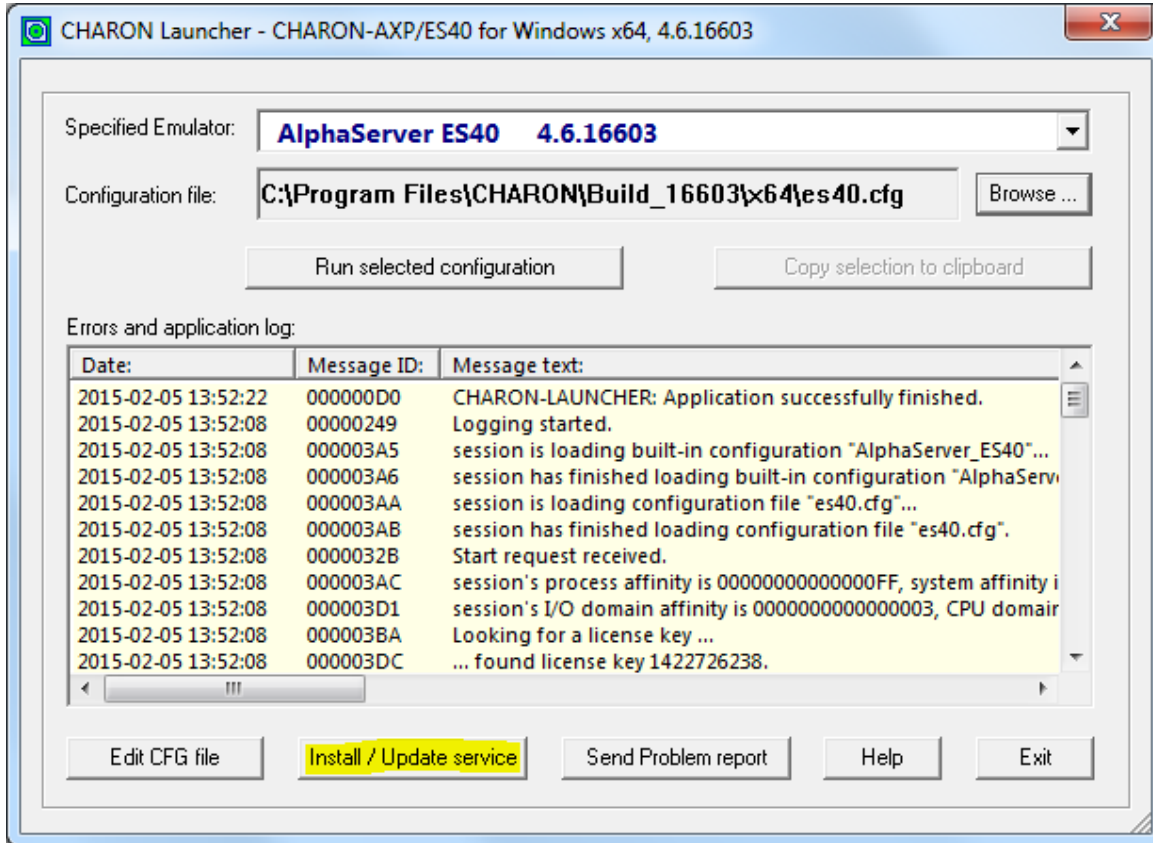
load KZPBA DKA scsi_id=7
set DKA container[0]="C:\My Services\es40_daemon_boot_disk.vdisk"
...
```

- Make sure the same physical devices are not used by other CHARON-AXP services

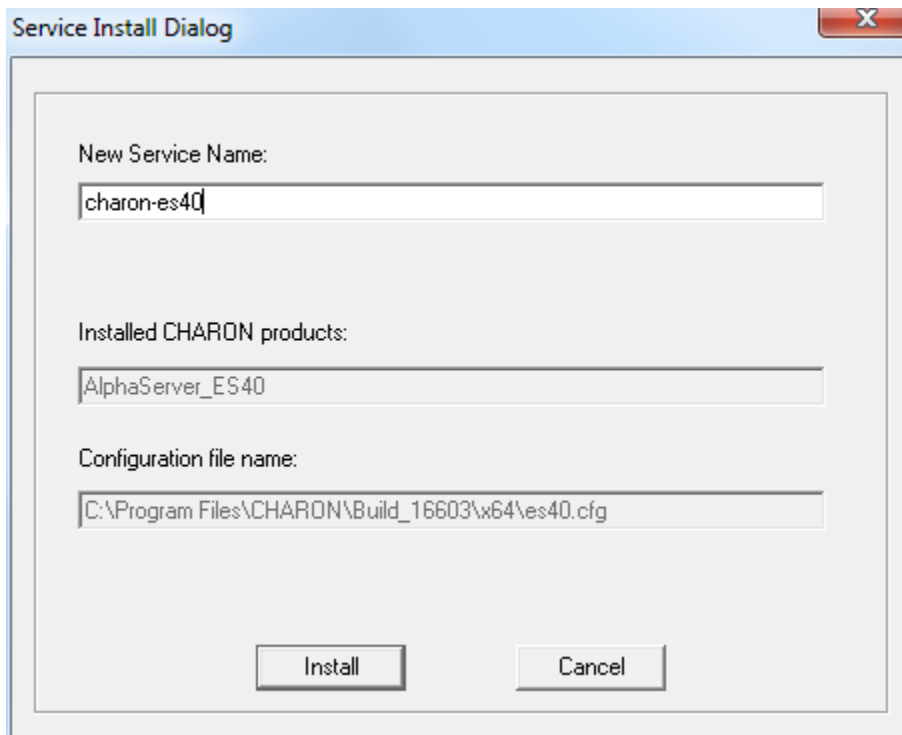
Save the configuration file with the "Notepad" save feature and then exit from it.

Run the configuration to confirm it is ready to be installed as a Windows service; stop CHARON-AXP (see above).

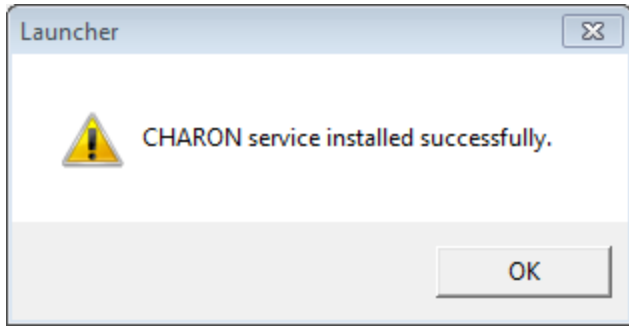
Press the "Install / Update service" button in the Launcher:



A special dialog for adding new CHARON-AXP services will appear:



Enter a new service name (in the example above it is "charon-es40") and press the "Install" button. If the service has been installed correctly, a configuration dialog box will appear:

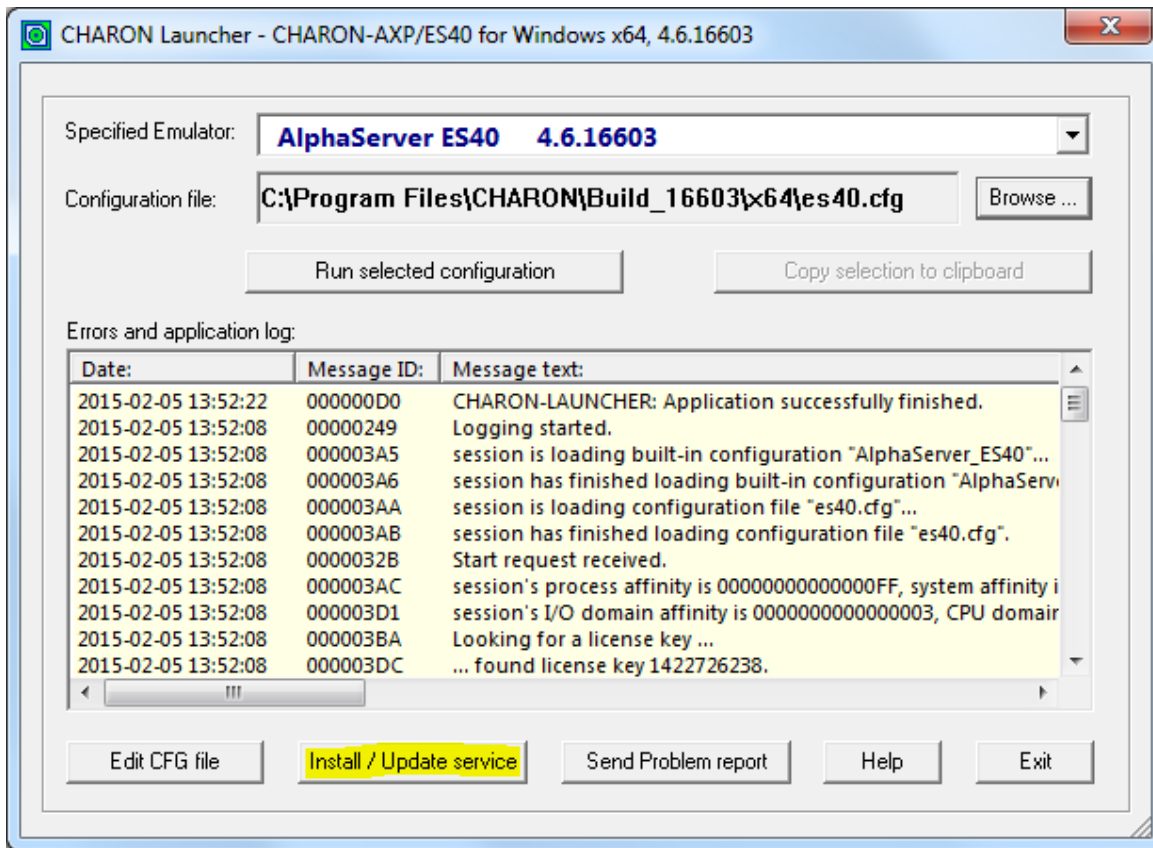


[Back to Table of Contents](#)

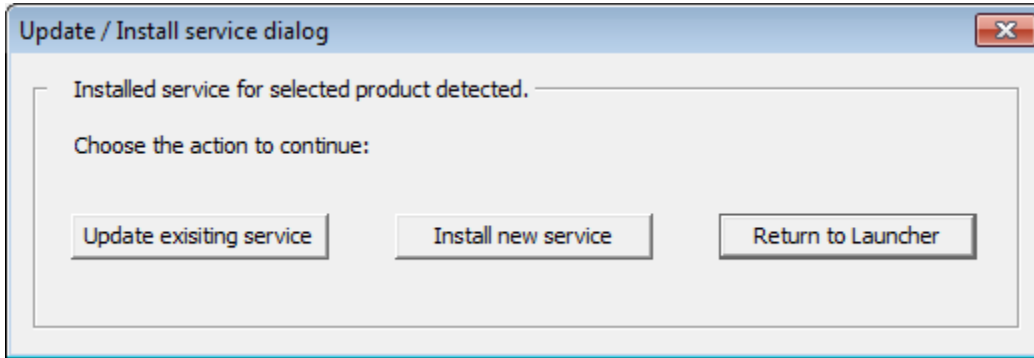
## Modification of CHARON-AXP service

Note that it is not enough just to modify the CHARON-AXP configuration file associated with a given CHARON-AXP service to apply the changes. In advance you need to update the CHARON-AXP service with the "Launcher" utility.

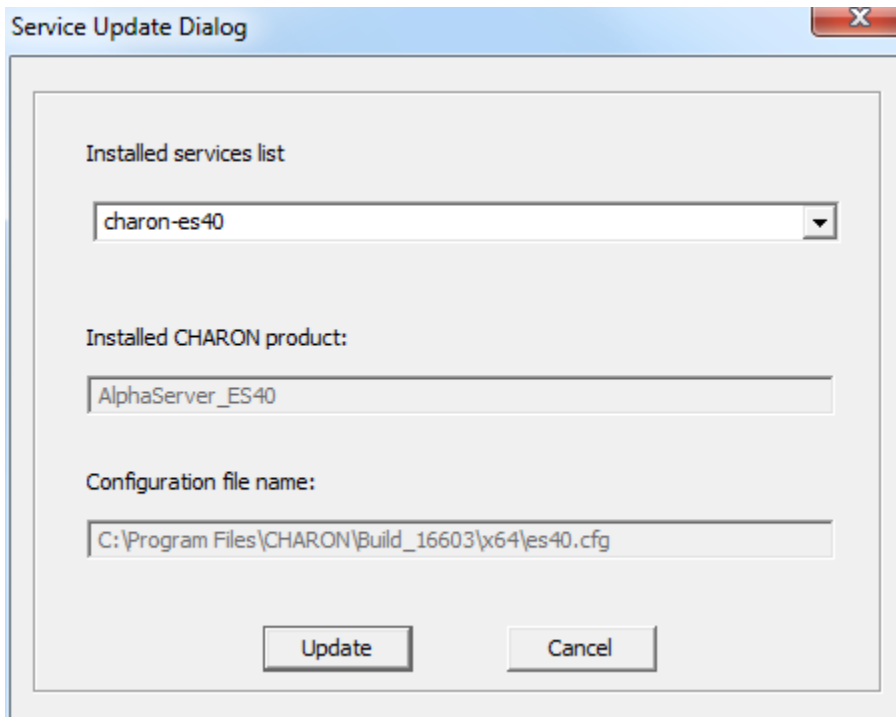
Press the "Install / Update service" button:



The install / update dialog will appear:



Press "Update existing service" button:



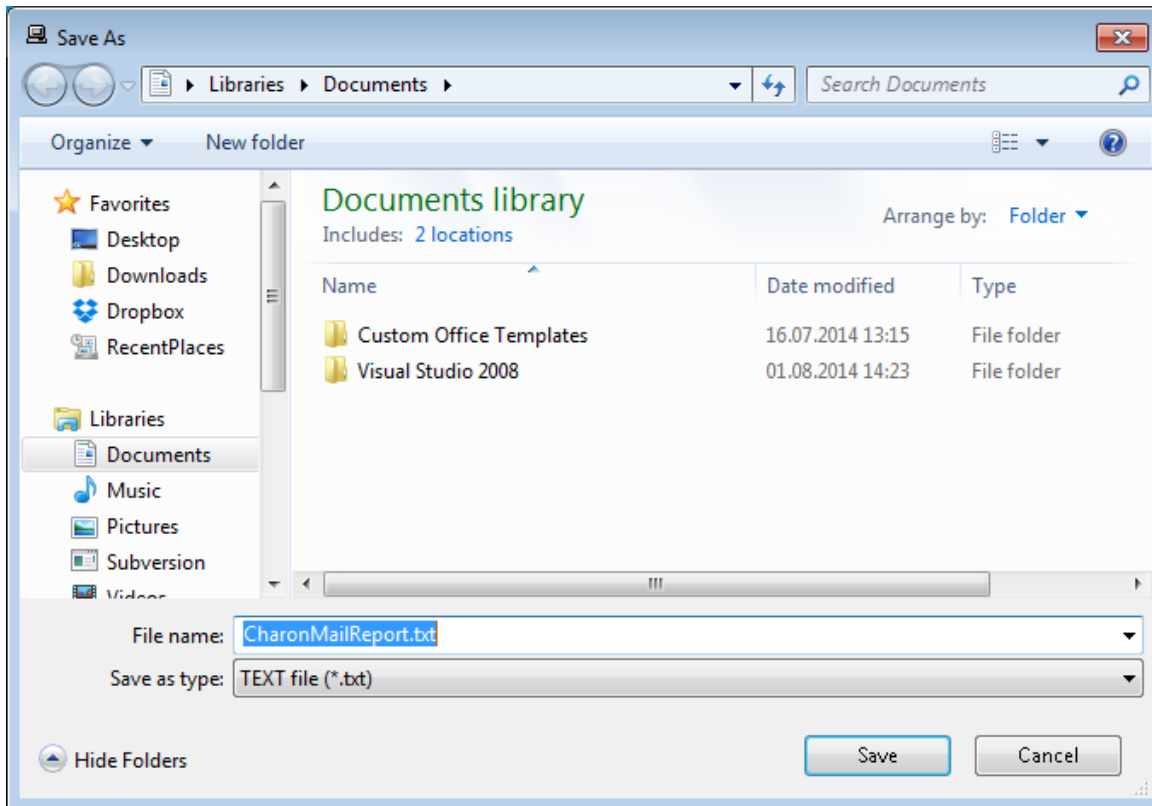
Select the service to update with the "Installed services list" drop-down box and press the "Update" button to apply.

 Please note that before updating a CHARON-AXP service, it must first be stopped

[Back to Table of Contents](#)

## Sending problem reports

The CHARON-AXP Launcher allows fragments of the log file to be copied to the clipboard. Select the fragment to be copied and press the "Copy selection to clipboard" button. Then select the folder and filename where it is to be saved:



To collect information for a problem report press the "Send Problem Report" button and save the information for customer support to a specific file. Then send the file along with detailed description of the problem to STROMASYS.

[Back to Table of Contents](#)

# CHARON Service Manager

## Table of Contents

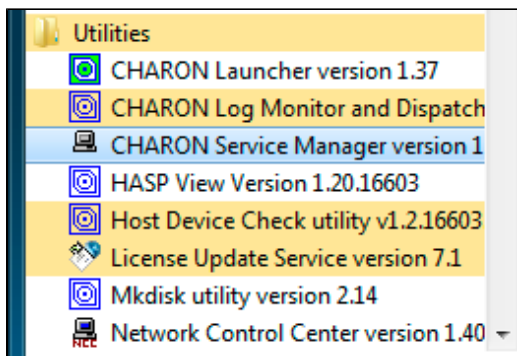
- General Description
- Starting CHARON service
- Managing CHARON service
- Stopping CHARON service
- Removing CHARON service
- Scheduling CHARON service
- Security settings
- Sending problem report
- Printing the CHARON service log file
- Reviewing the CHARON service configuration
- Tracing system resources

[Back to Table of Contents](#)

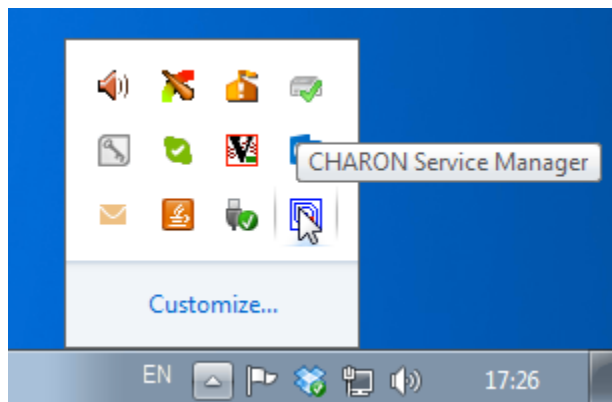
## General Description

Used to manage the CHARON services, including their stopping, starting, removing, logs tracing and specifics of execution.

To start the "CHARON Service Manager" use the "Start" menu:



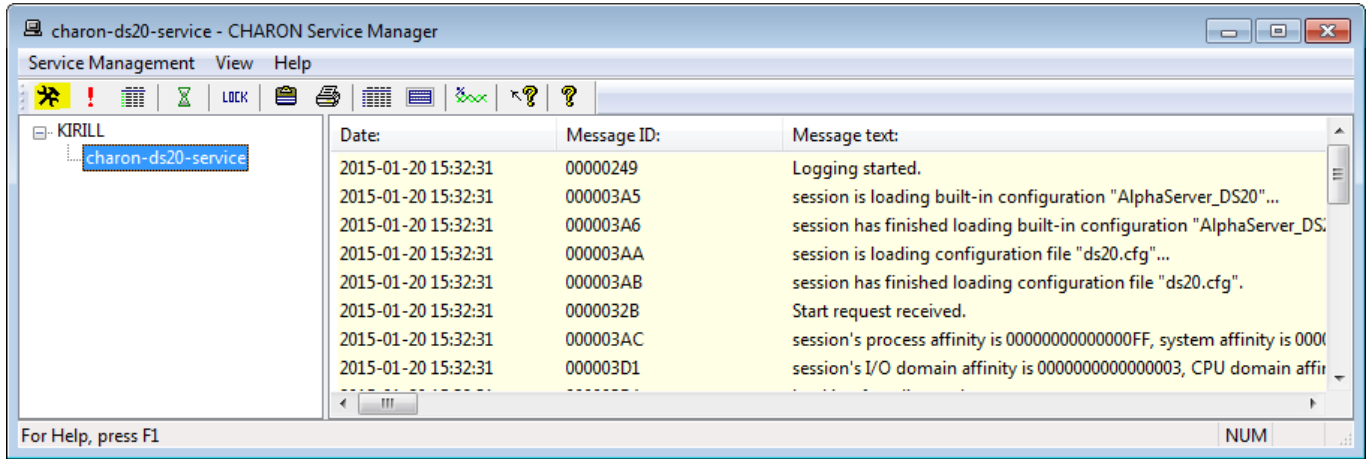
It is also possible to start the "CHARON Service Manager" utility from the system tray menu:



[Back to Table of Contents](#)

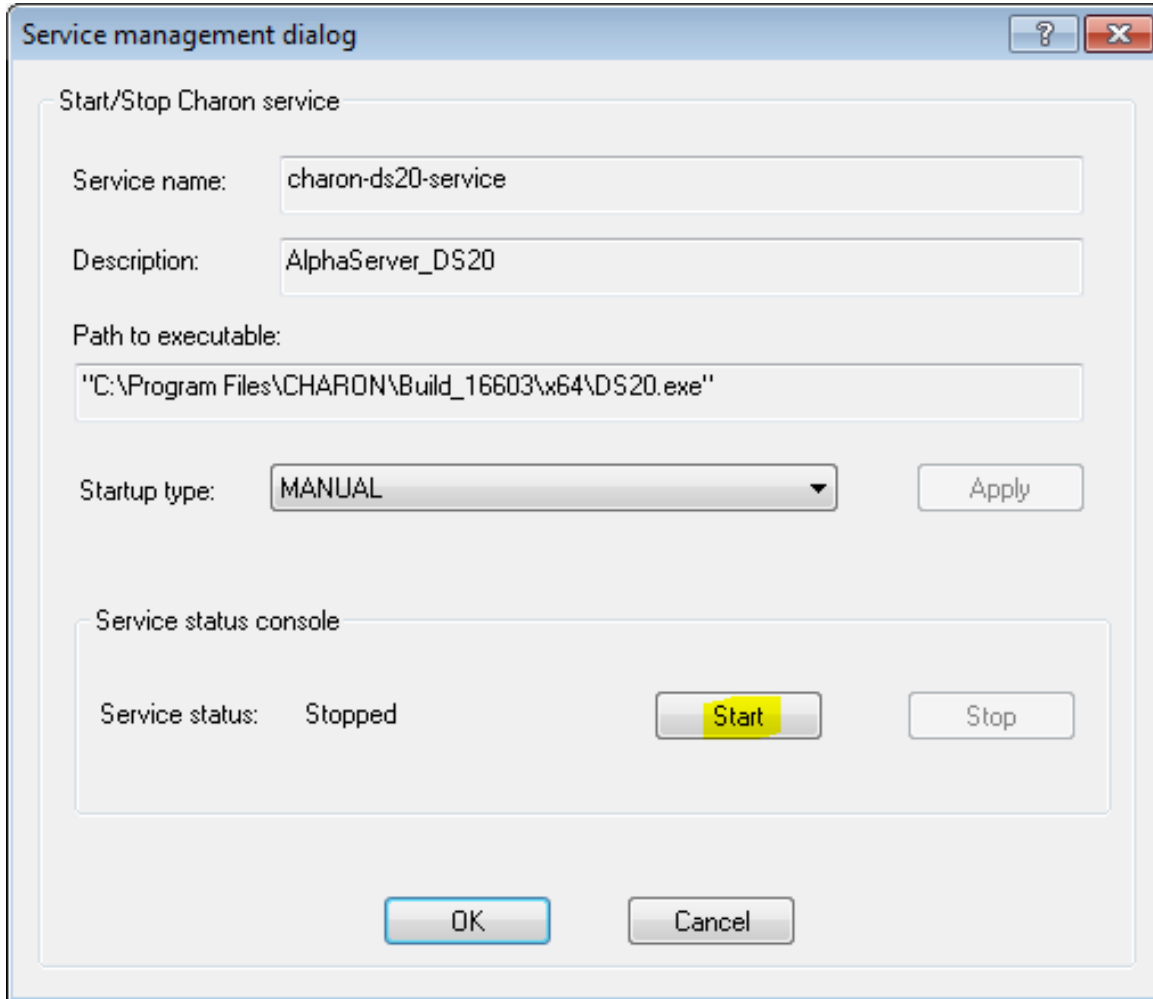
## Starting CHARON service

In the main window of the utility, select the target CHARON service ("charon-ds20-service" in our example) and press the "Service" button (labeled with yellow on the screenshot below):



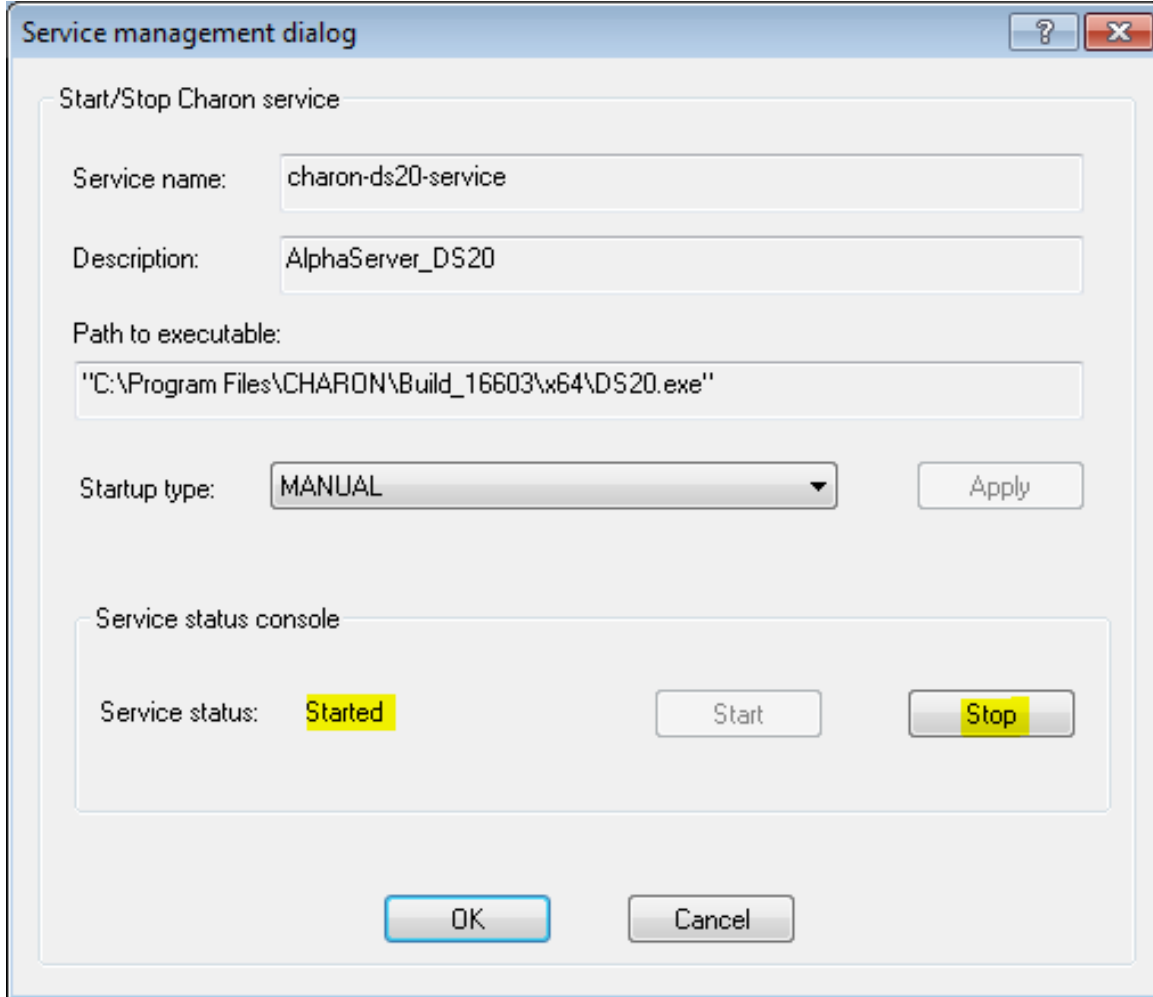


The "Service Management dialog" will appear:



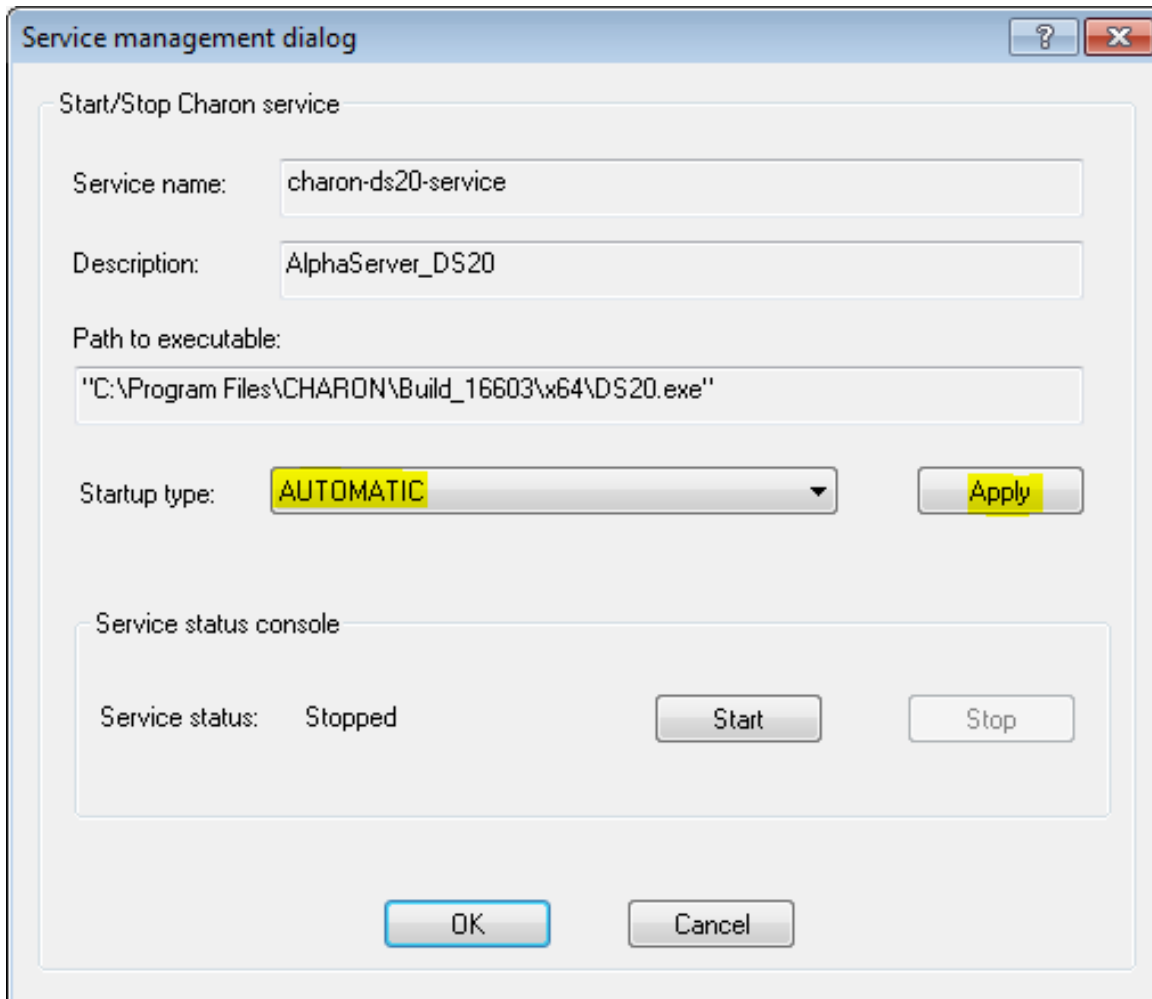
The "Service Management dialog" shows the current status of the selected CHARON service and permits changes to its start up type and to start and stop it. The example screenshot above shows that the service "charon-ds20-service" startup type is "MANUAL" (can be started / stopped only manually) and in the "Stopped" state - these are the default values.

Press the "Start" button to start the service:



CHARON will start and the Service status will be "Started". To stop the service, press the "Stop" button.

It is possible, and often desirable, to configure the selected CHARON service to start when host operating system starts. To enable this capability, select "AUTOMATIC" in the "Startup type" drop-down box and press the "Apply" button:



Note that a certain delay may occur before the Sentinel Run-time process finds the network license on CHARON-AXP host system startup. If the CHARON-AXP service is starting automatically on the host system, it may report a "License not found" error and exit.

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

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

where:

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

Example:

```
set session license_key_lookup_retry = 5
```

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

See [General Settings](#) section for more details.

[Back to Table of Contents](#)

## Managing CHARON service

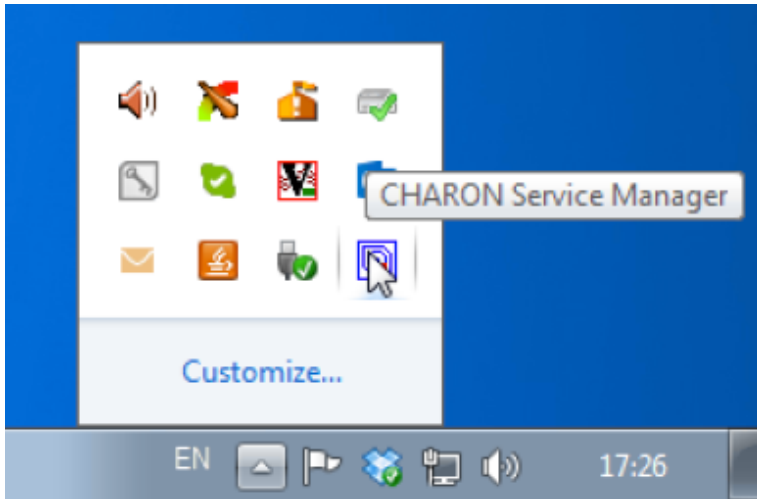
To see the execution log of a CHARON service, select the service ("charon-ds20-service" in the example below) and the current log messages will be displayed in the right panel of the utility interface. You may need to scroll up to see startup/previous messages:

The screenshot shows the 'charon-ds20-service - CHARON Service Manager' window. The left pane shows a tree view with 'KIRILL' expanded and 'charon-ds20-service' selected. The main area displays a log table with columns for Date, Message ID, and Message text. The log entries show the service starting, verifying configuration, receiving a start request, checking for a license key, and displaying system information before stopping.

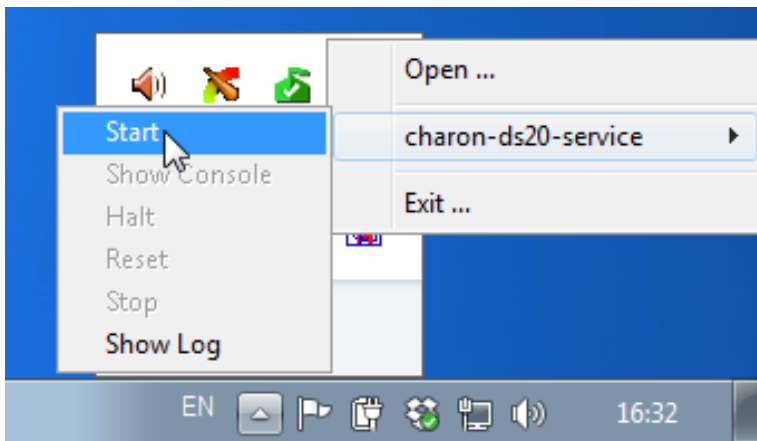
Date:	Message ID:	Message text:
2015-01-20 16:20:56	000003A9	session has verified service configuration checksum.
2015-01-20 16:20:56	0000032B	Start request received.
2015-01-20 16:20:56	000003AC	session's process affinity is 00000000000000FF, system affinity is 00000000000000FF.
2015-01-20 16:20:56	000003D1	session's I/O domain affinity is 0000000000000003, CPU domain affinity is 0000000000
2015-01-20 16:20:56	000003BA	Looking for a license key ...
2015-01-20 16:20:57	000003DC	... found license key 1422726238.
2015-01-20 16:20:57	0000024D	STROMASYS SA, (C) 2009-2014
2015-01-20 16:20:57	00000350	CHARON-AXP (AlphaServer DS20), V 4.6 B 16603, Dec 15 2014 / net-nest / 1422726238
2015-01-20 16:20:57	00000336	The end user of this software has agreed to STROMASYS' Terms and Conditions for S
2015-01-20 16:20:57	0000009D	License info:
2015-01-20 16:20:57	0000009D	CHARON product code: "CHAXP-460xx-WI".
2015-01-20 16:20:57	0000009D	Licensed to: "net-nest".
2015-01-20 16:20:57	0000009D	License KEY driver is installed as hasplms and is running.
2015-01-20 16:20:57	0000009D	Executable "C:\Windows\system32\hasplms.exe", version 16.0.1.41863.
2015-01-20 16:20:57	00000097	OS Environment: Microsoft Windows 7 Professional, 64-bit, Service Pack 1 (Build 7601
2015-01-20 16:20:57	00000098	Host CPU: Intel(R) Xeon(R) CPU E31275 @ 3.40GHz x8.
2015-01-20 16:20:57	00000099	Host Memory: 24320Mb
2015-01-20 16:20:57	0000032C	"AlphaServer_DS20" started.
2015-01-20 16:22:37	0000032D	"AlphaServer DS20" stop request received.
2015-01-20 16:22:37	0000014C	EWA0: Stopping network interface ... please wait.
2015-01-20 16:22:37	00000348	ide0 is being set OFFLINE
2015-01-20 16:22:37	00000348	The previous message has been repeated 2 times.
2015-01-20 16:22:37	0000032E	Stopped.
2015-01-20 16:22:37	0000024A	Logging stopped.

For Help, press F1

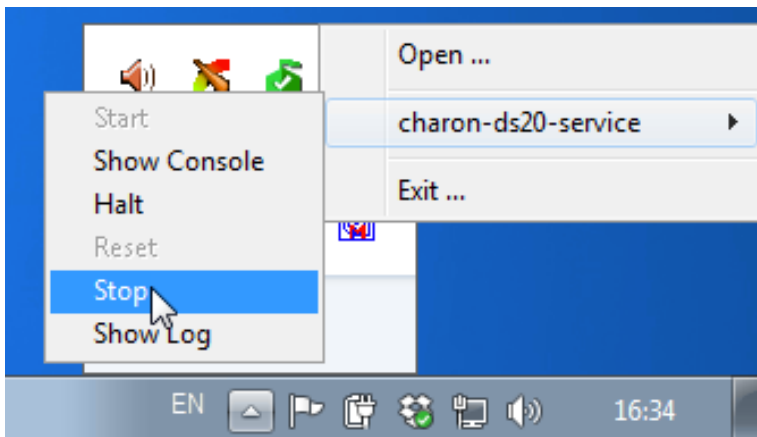
To control execution of CHARON services, open the "CHARON Service Manager" utility from the "Start" menu or use the system tray shortcut to manage the services w/o opening the main window of the utility:



Press the right button of the mouse to see a list of the installed CHARON services ("charon-ds20-service" in our example):



Select the Service to see the list of the available operations:



Explanation:

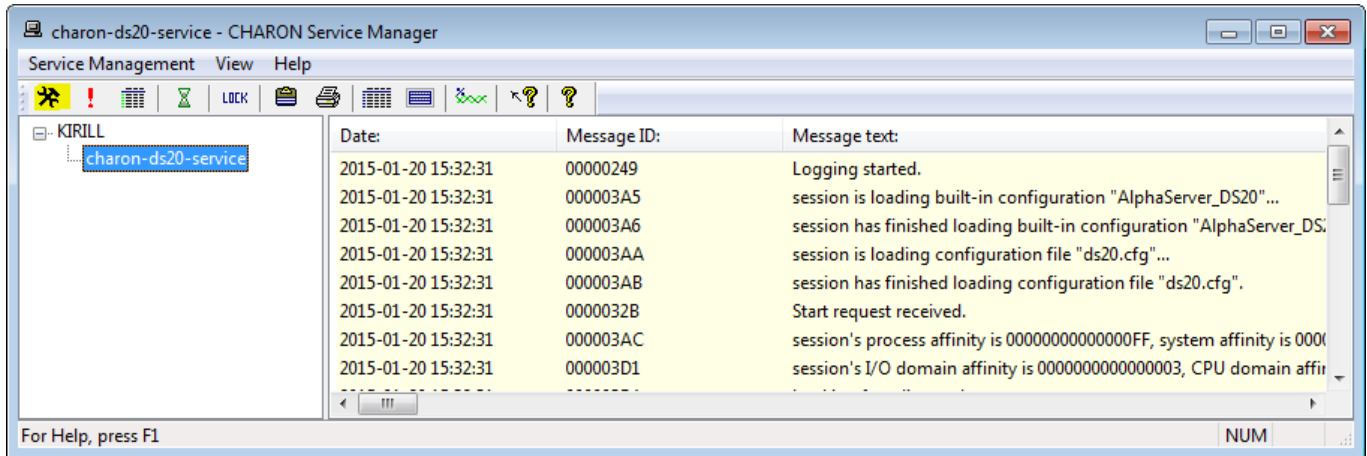
Menu Item	Function
<b>Start</b>	Start CHARON service
<b>Show Console</b>	Show CHARON service console if it has been closed or not active on the automatic service startup
<b>Halt</b>	Halt HP Alpha emulator associated with the selected CHARON service
<b>Reset</b>	Reset the virtual HP Alpha associated with the selected CHARON service
<b>Stop</b>	Stop CHARON service
<b>Show Log</b>	Show execution log of the selected CHARON service

Choose the desired action and click on it.

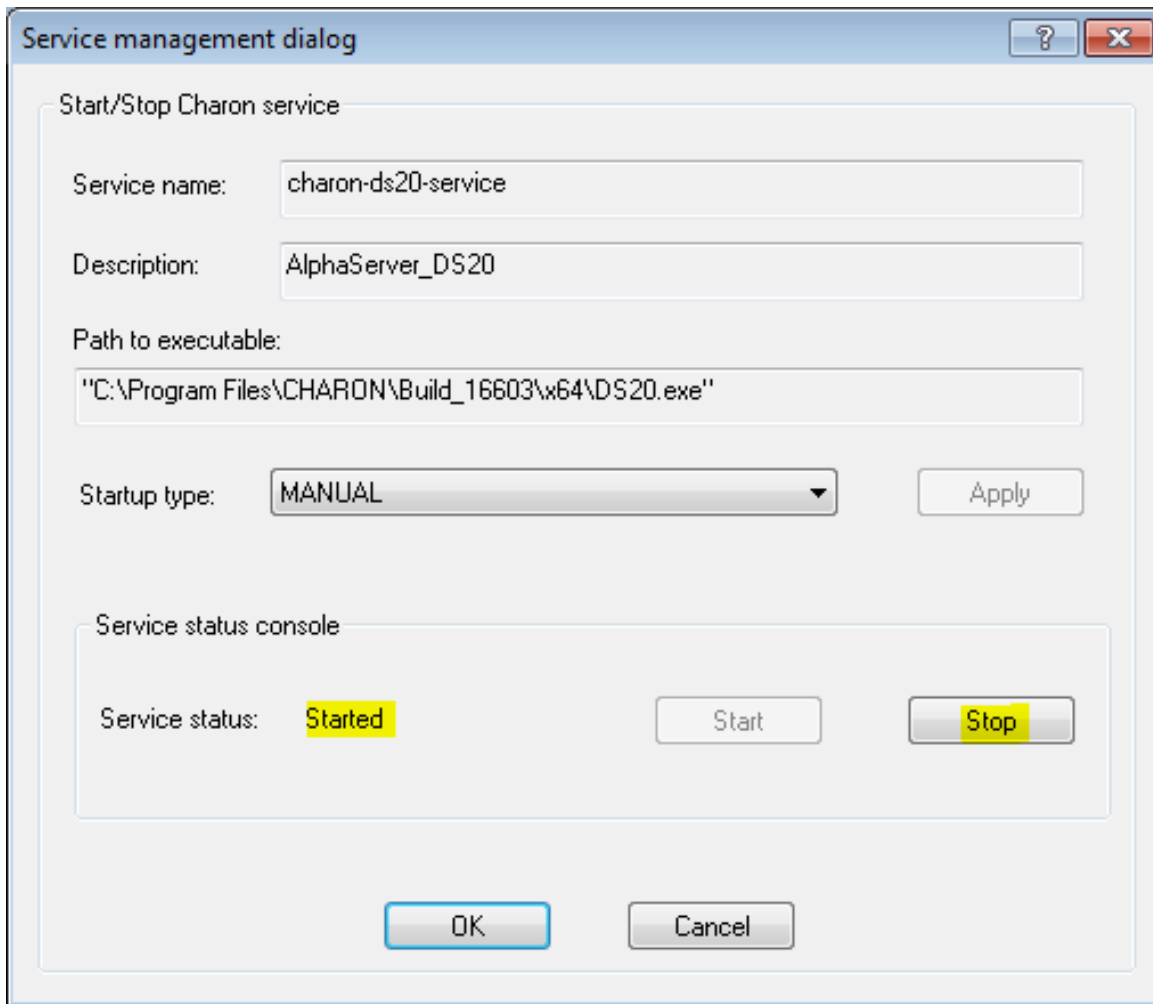
[Back to Table of Contents](#)

## Stopping CHARON service

In the main window of the utility, select the target CHARON service ("charon-ds20-service" in our example) and press the "Service" button (labeled with yellow on the screenshot below):

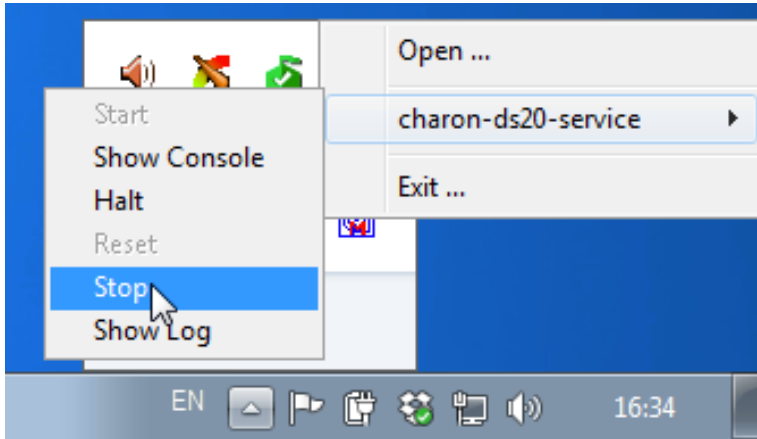


Press the "Stop" button in the resulting dialog box:



The CHARON Service can be Stopped using the shortcut on the Windows Task Bar.

Open up the "CHARON Service Manager" system tray menu, select the service you are going to stop and click on the "Stop" command:

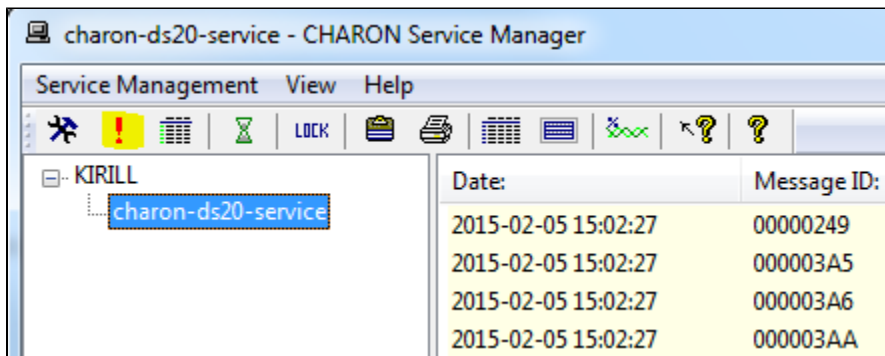


**!** Please note: before stopping the CHARON-AXP service, shutdown the operating system running in CHARON-AXP.

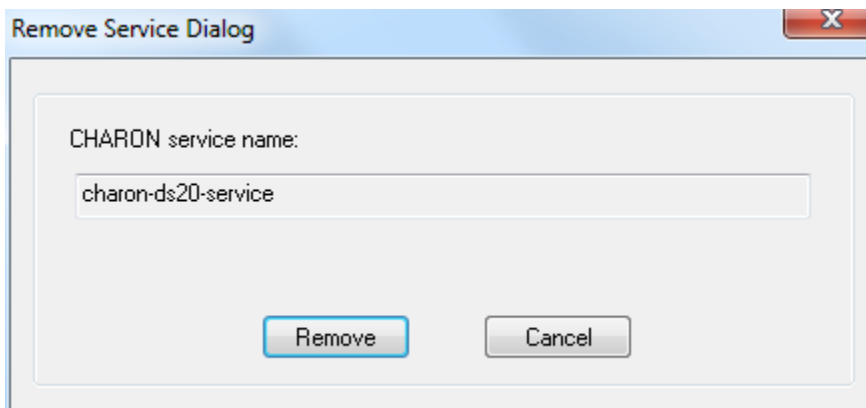
[Back to Table of Contents](#)

## Removing CHARON service

In the main window of the utility select a CHARON service to remove ("charon-ds20-service" in our example) and press the button with an exclamation sign on it:



Confirm removing the selected CHARON service by pressing the "Remove" button:



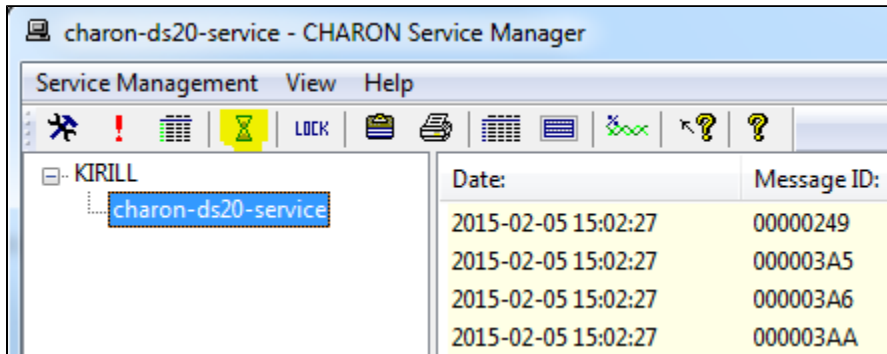
[Back to Table of Contents](#)



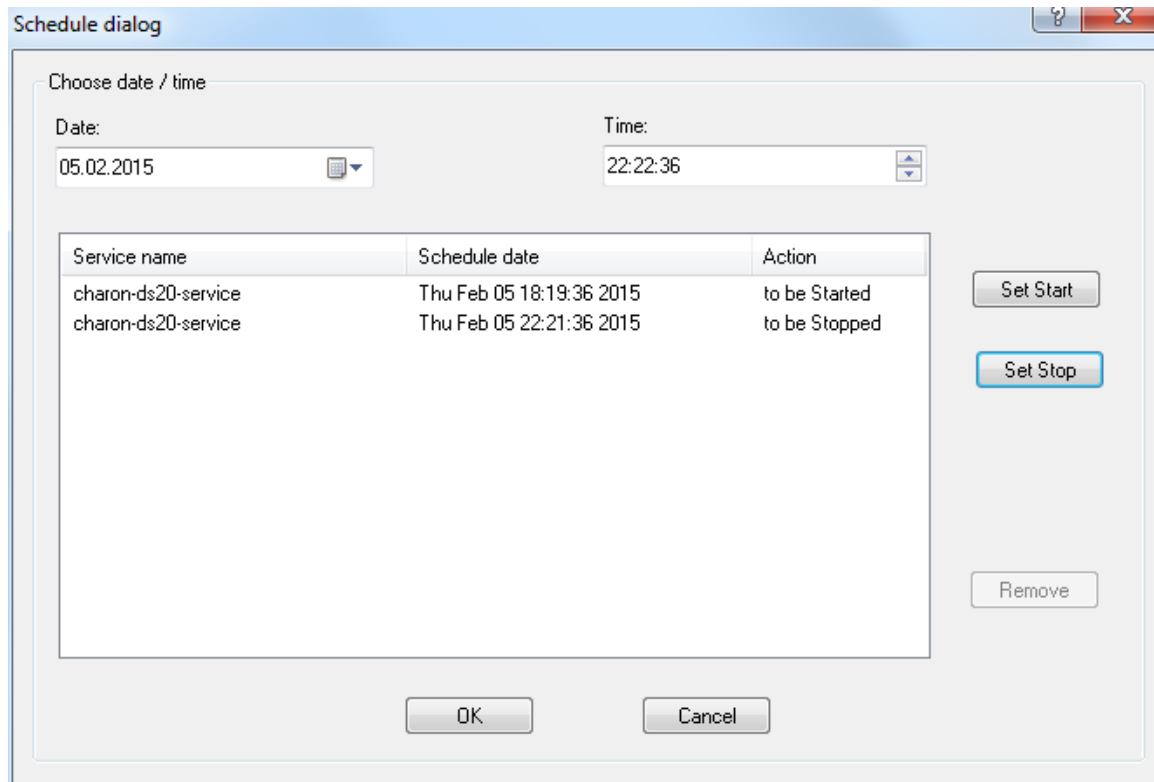
## Scheduling CHARON service

The "CHARON Service Manager" utility provides a function to set the start and stop date/time for each installed CHARON service.

Select the desired CHARON service and press on the hourglass icon:



In the popup specify the start and the stop date and time for the selected service:



Press "Ok" to apply settings.

**⚠** Note you will have to perform a clean shutdown of the HP Alpha system before defining the service to be stopped by a scheduled operation

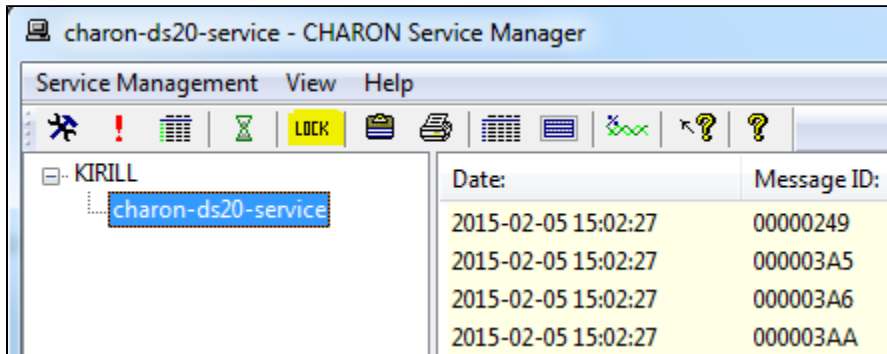
To remove the scheduled event select it and press the "Remove" button.

[Back to Table of Contents](#)

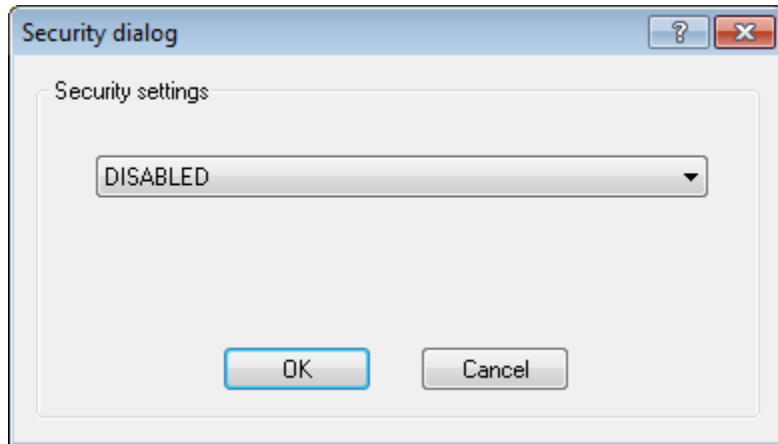
## Security settings

The security settings informs the CHARON host to lock the desktop after a given amount of time of inactivity.

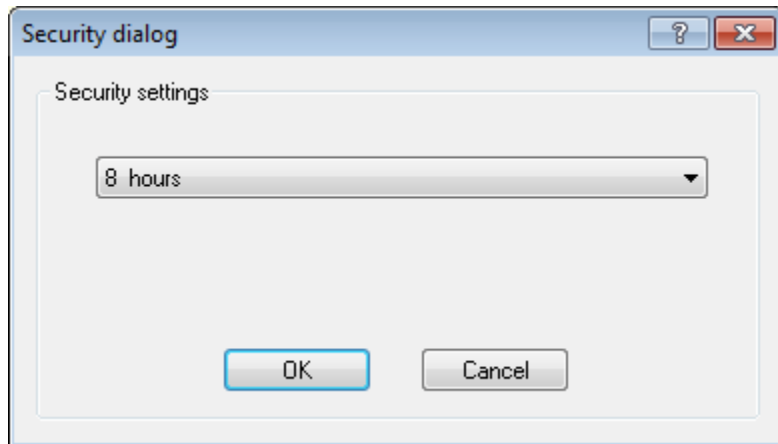
To enable this function press the "Lock" button:



The following popup will appear:



Specify the desired period of time before locking the session and press the "Ok" button:

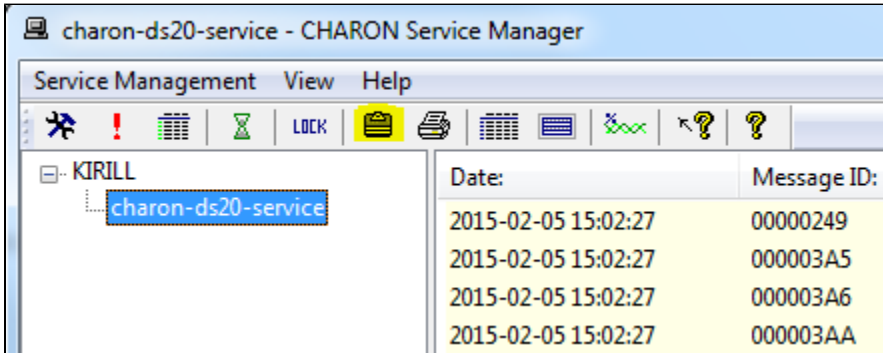


Note that the security settings are active only when the "CHARON Service Manager" is running. On exit from the utility the security settings reset.

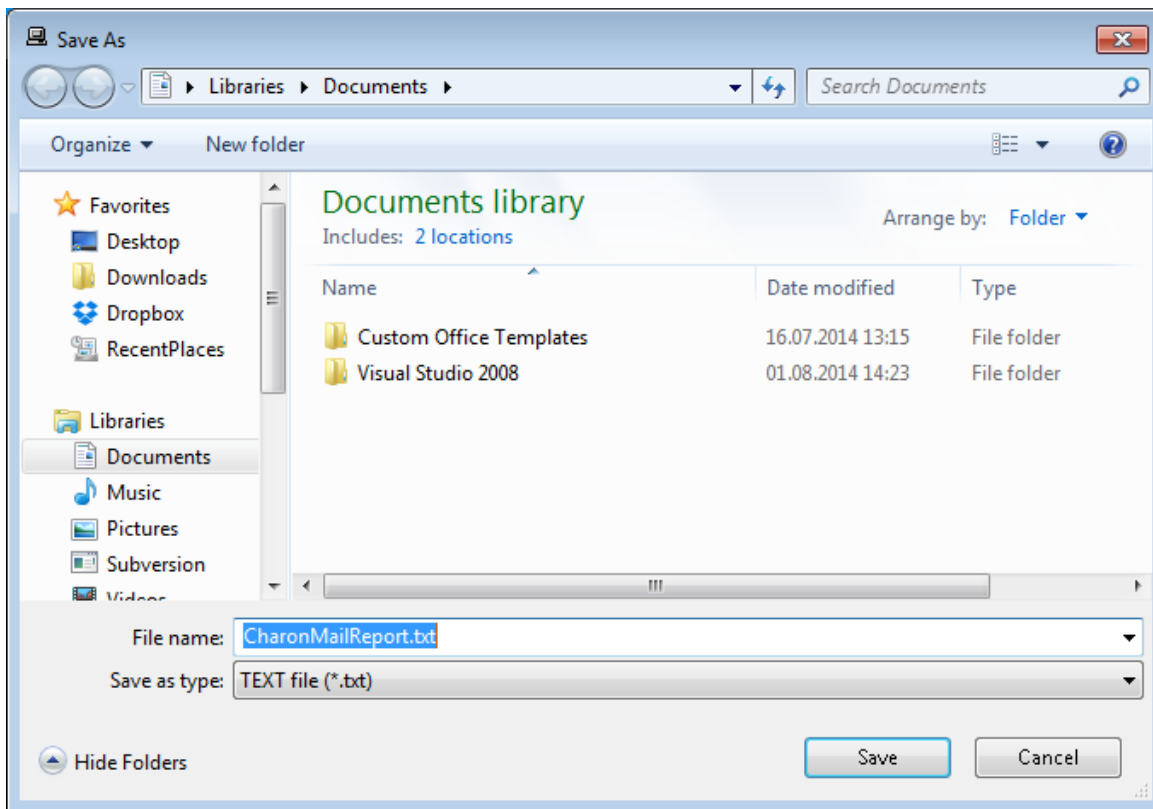
[Back to Table of Contents](#)

## Sending problem report

To collect information for a problem report select the service you have issues with and click on the yellow highlighted button as shown below:



Choose the target filename to store the information for STROMASYS customer support and press the "Save" button:

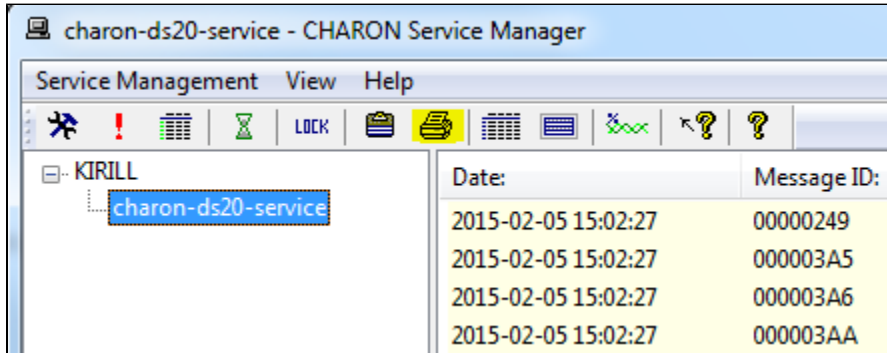


Send the file to the STROMASYS customer support along with a detailed description of the problem.

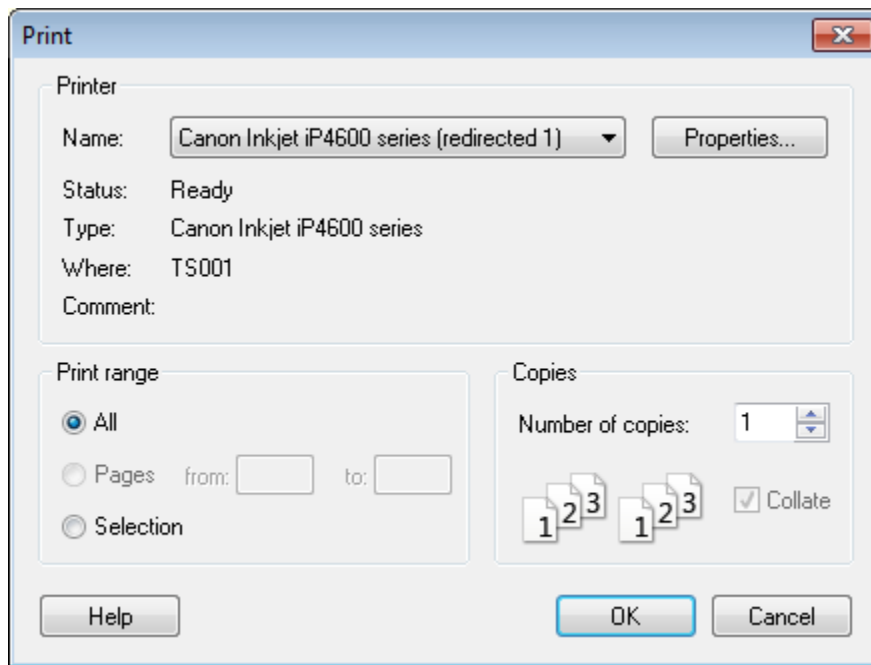
[Back to Table of Contents](#)

## Printing the CHARON service log file

To print the CHARON service log file select the target CHARON service and click on the yellow highlighted button as shown below:



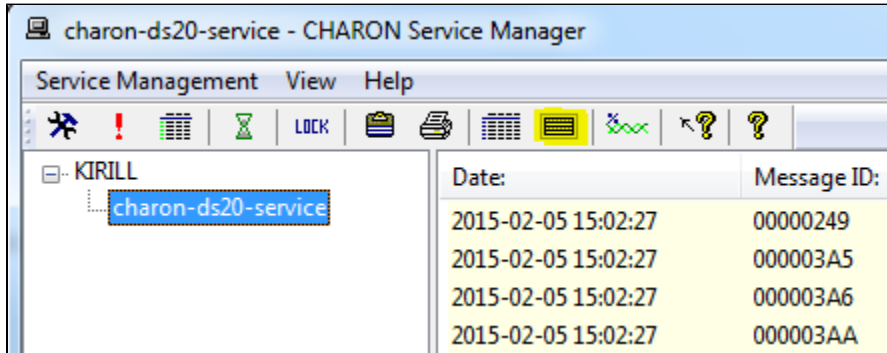
Choose the target printer and press the "Ok" button:



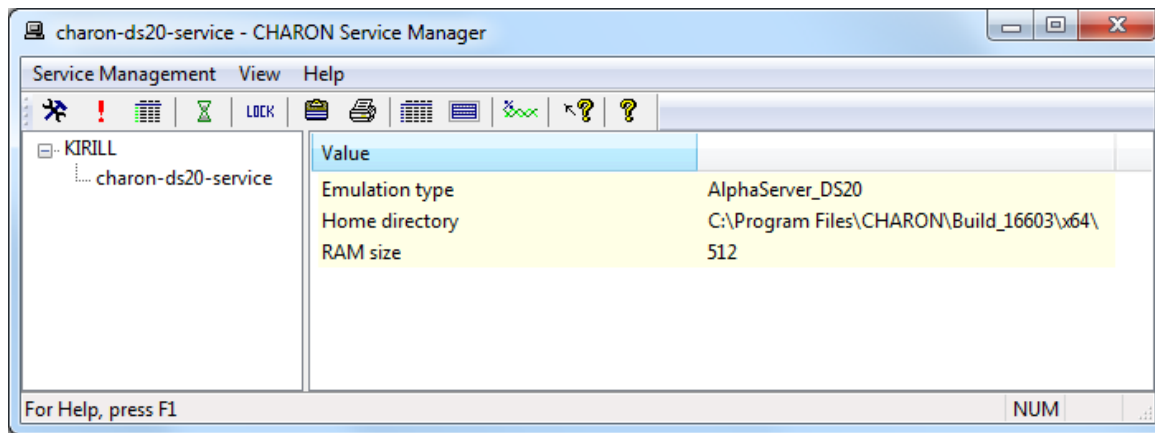
[Back to Table of Contents](#)

## Reviewing the CHARON service configuration

To review the CHARON service configuration select the target CHARON service and click on the yellow highlighted button as shown below:



This utility will display the selected CHARON service configuration:



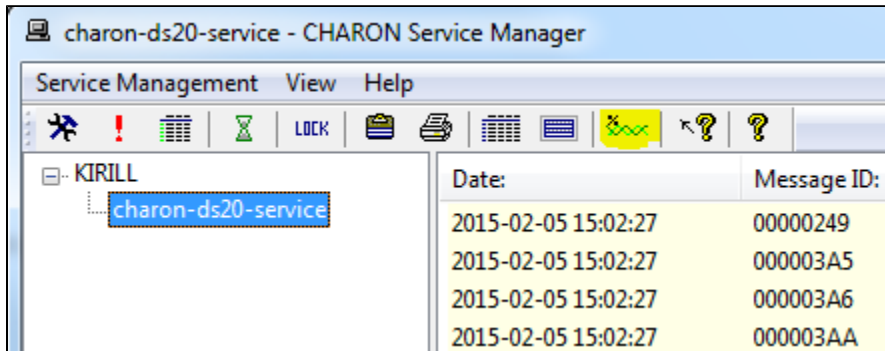
Press the "Print" button to print it if needed.

[Back to Table of Contents](#)

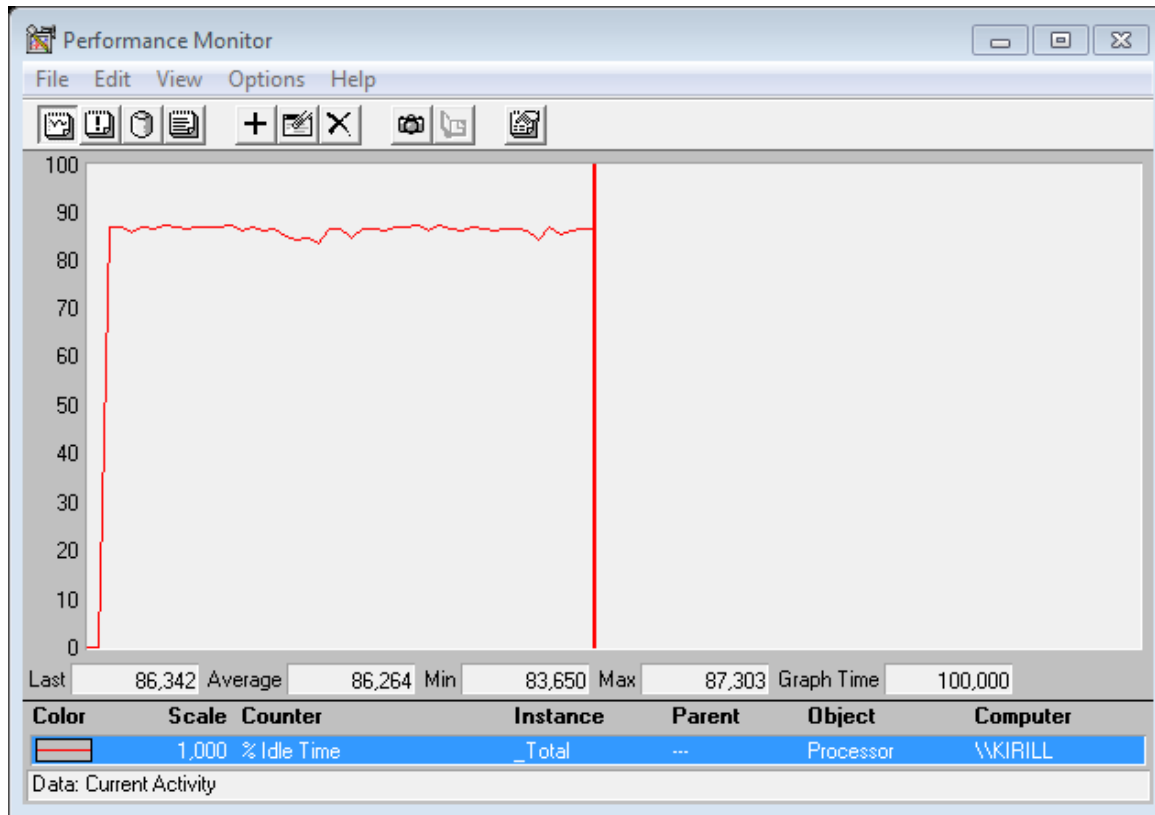
## Tracing system resources

The "CHARON Service Manager" has a functionality to trace usage of some system resources in real time viewing.

Click on the yellow highlighted button as shown below:



In the popup it is possible to set the parameters to trace:



Choose the desired parameters to view using the "+" button and watch their values change in real time.

Note that the "Performance Monitor" utility is a standard Windows tool (just included in the CHARON distributive for convenience), so for more information please refer to the Microsoft documentation.

[Back to Table of Contents](#)

# Host Device Check

## Table of Contents

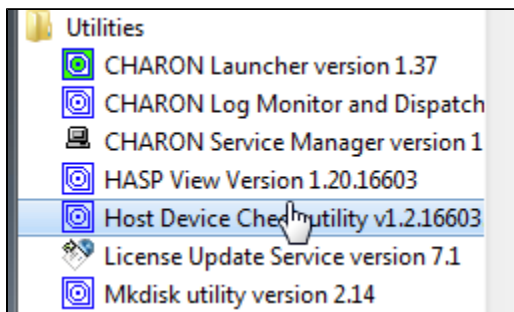
- General Description
- Reviewing available physical disks
- Reviewing all the available host resources
- Collecting the configuration strings

[Back to Table of Contents](#)

## General Description

The "Host Device Check" utility is used to review system resources that can be mapped to CHARON.

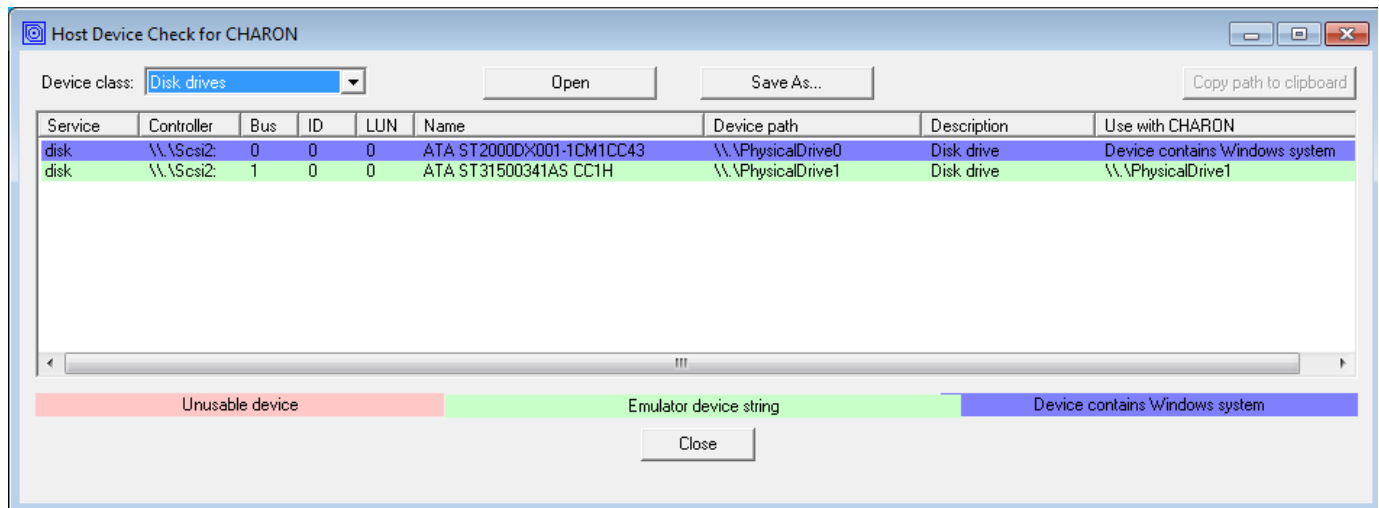
To start the utility click on the following icon under the "Start" menu:



[Back to Table of Contents](#)

## Reviewing available physical disks

By default the "Host Device Check" utility reports the available physical disks:

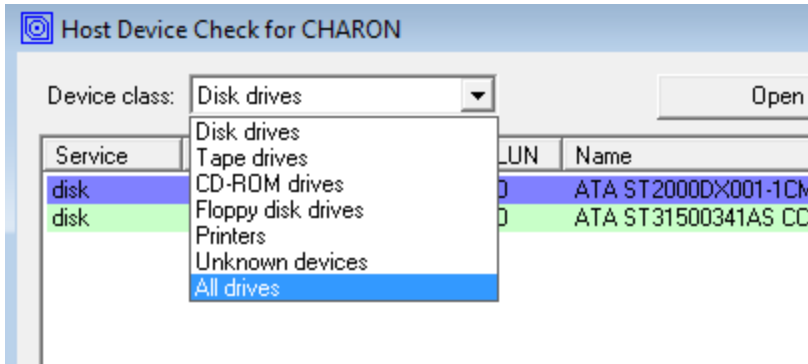


Please note that the "Unusable device" (marked with red) and "Device contains Windows system" **must not** be mapped to CHARON!

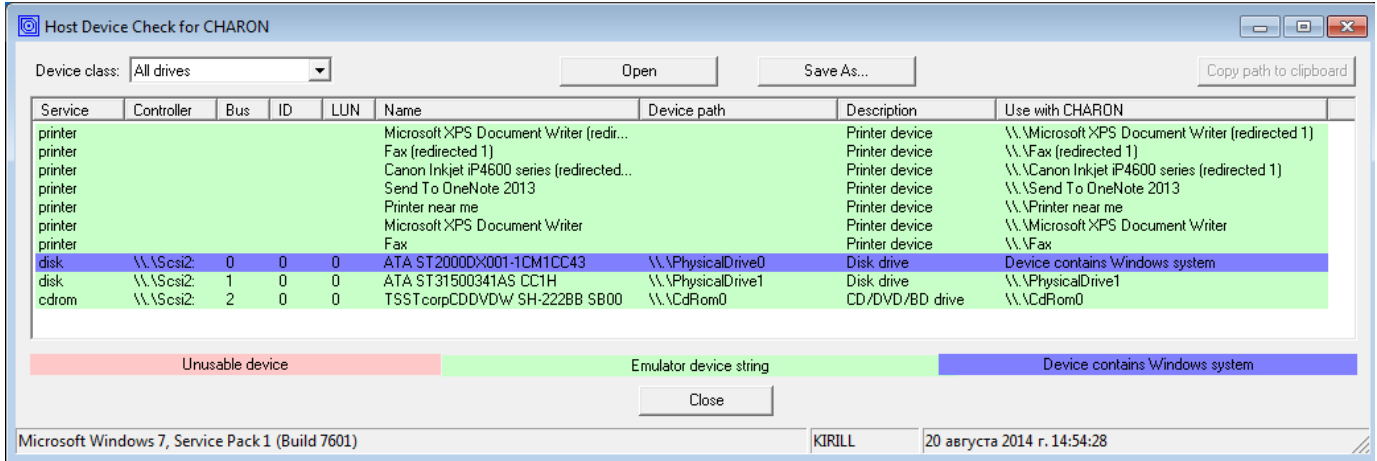
[Back to Table of Contents](#)

## Reviewing all the available host resources

Open "All drives" in the "Device class" drop-down box:



The "Host Device Check" utility will display all the host resources:



It is also possible to choose all the other categories to narrow the possible mapping options list.

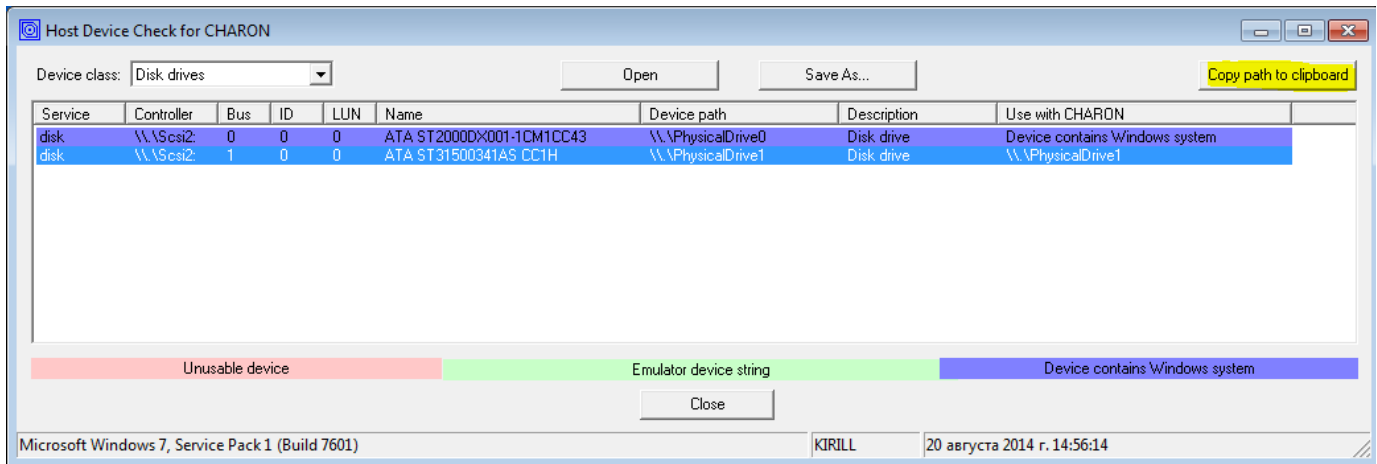
Note the "Use with CHARON" column contains the actual configuration options for each available device to be inserted, if needed, in the CHARON configuration file.

[Back to Table of Contents](#)



## Collecting the configuration strings

To collect the actual configuration strings to be used in the CHARON configuration file select the target device and press the "Copy path to clipboard" button:



The selected configuration string will be copied to the clipboard and then can be pasted to the CHARON configuration file with "Ctrl-V" keys combination.

The buttons "Open" and "Save As..." helps to open up the displayed options in form of text file and save this text file under some given name.

[Back to Table of Contents](#)

# Network Control Center

## Table of Contents

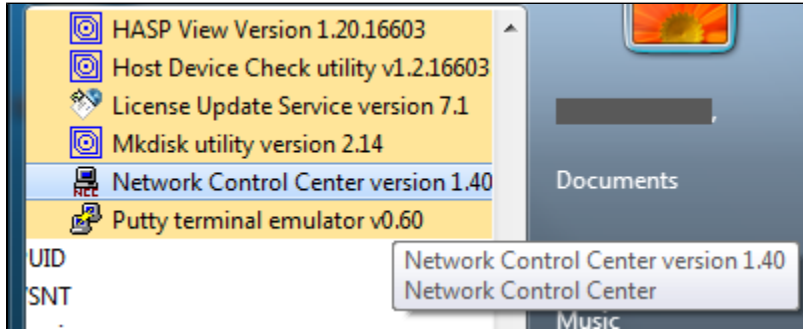
- General Description
- Installation of CHARON network drivers
- Deinstallation of the CHARON network driver
- Configuring the host network interfaces for CHARON
- Release of the host network interfaces
- Troubleshooting the CHARON network interfaces configuration
- Disable "offload" parameters
- Monitoring the CHARON network activity

[Back to Table of Contents](#)

## General Description

The "Network Control Center" utility is used to configure, verify dedicated network interfaces and trace network activity for a CHARON network.

Click on the following menu entry, under the "Start" menu, to start the utility:

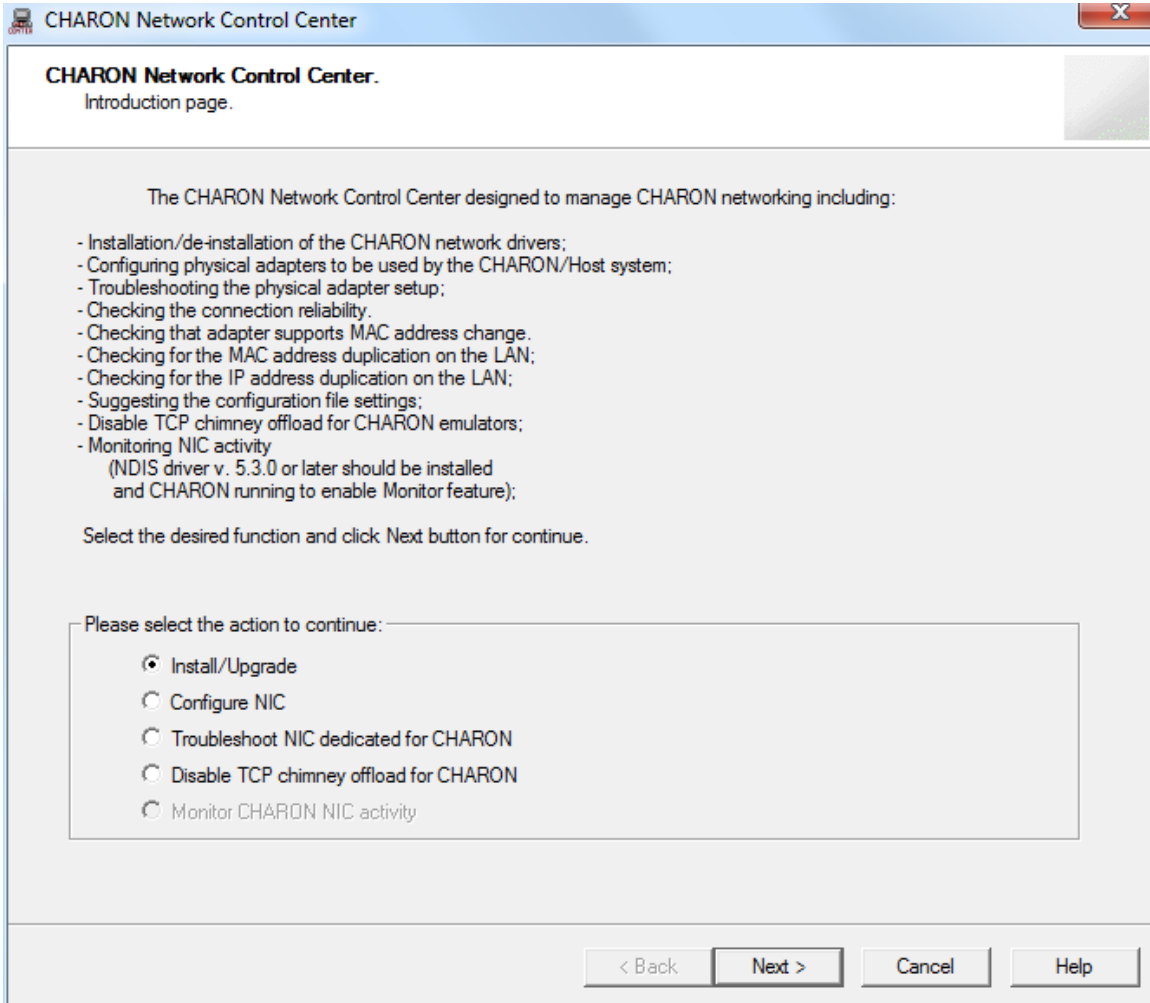


[Back to Table of Contents](#)

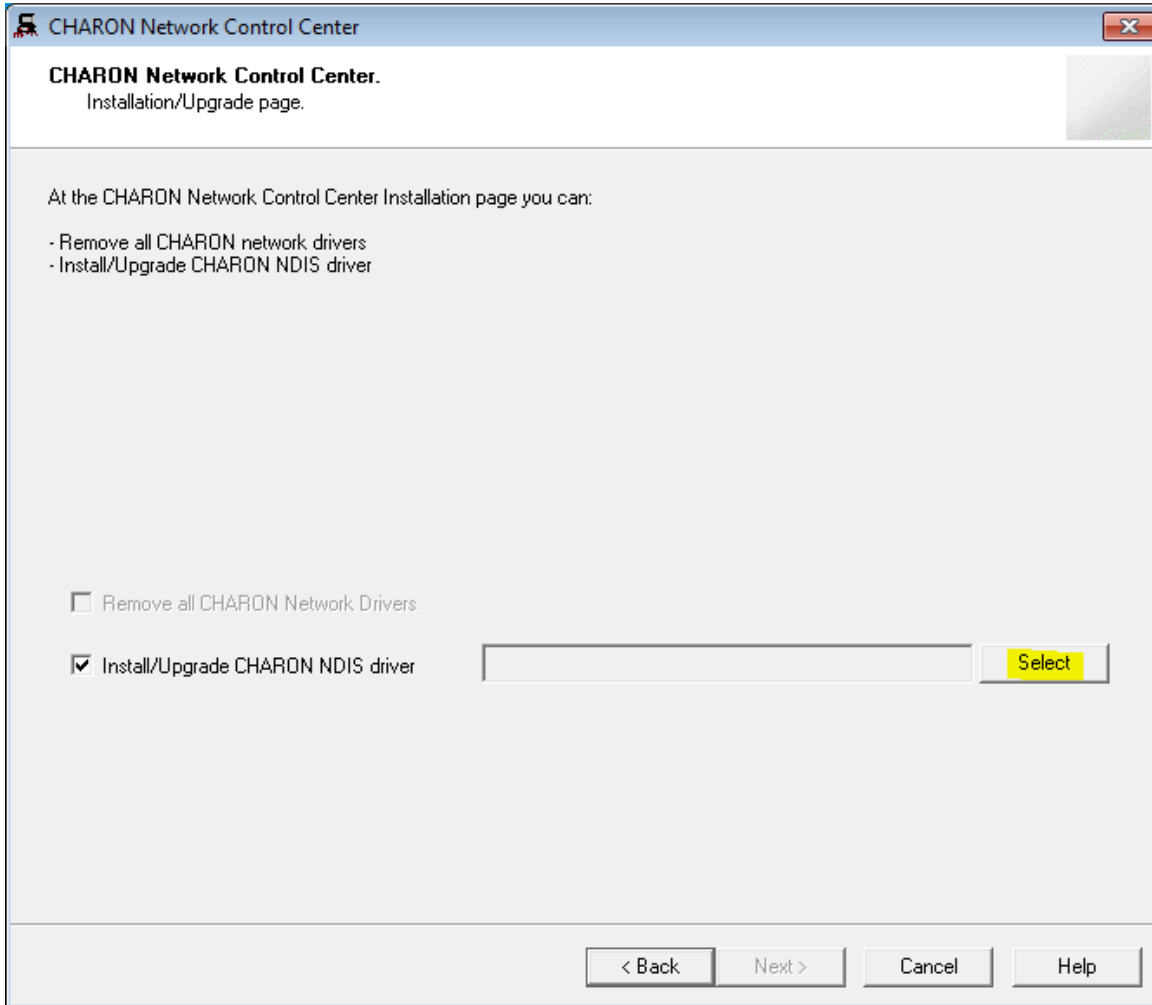
## Installation of CHARON network drivers

By default a CHARON network driver is automatically installed by the CHARON installation procedure.

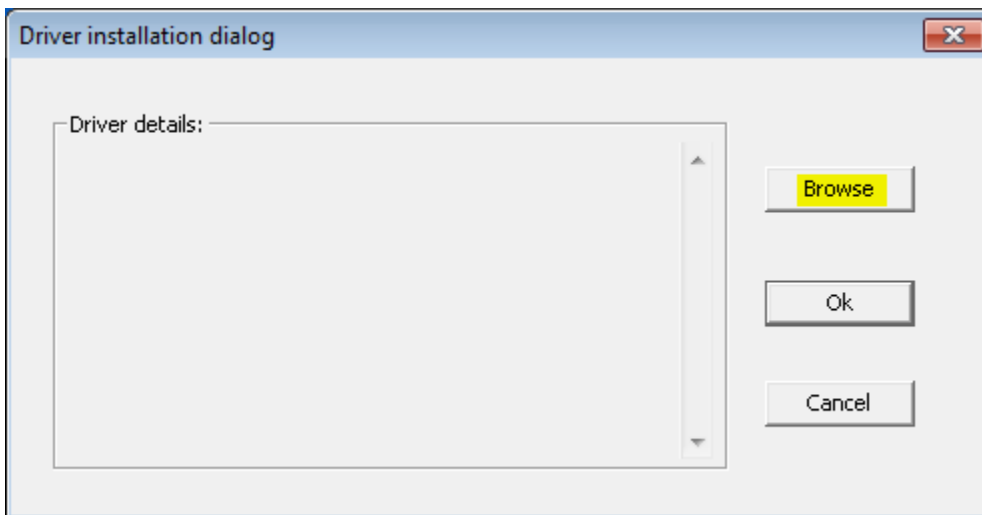
When it is needed to install a modified driver, use the following procedure. Select "Install/Upgrade" and press "Next":



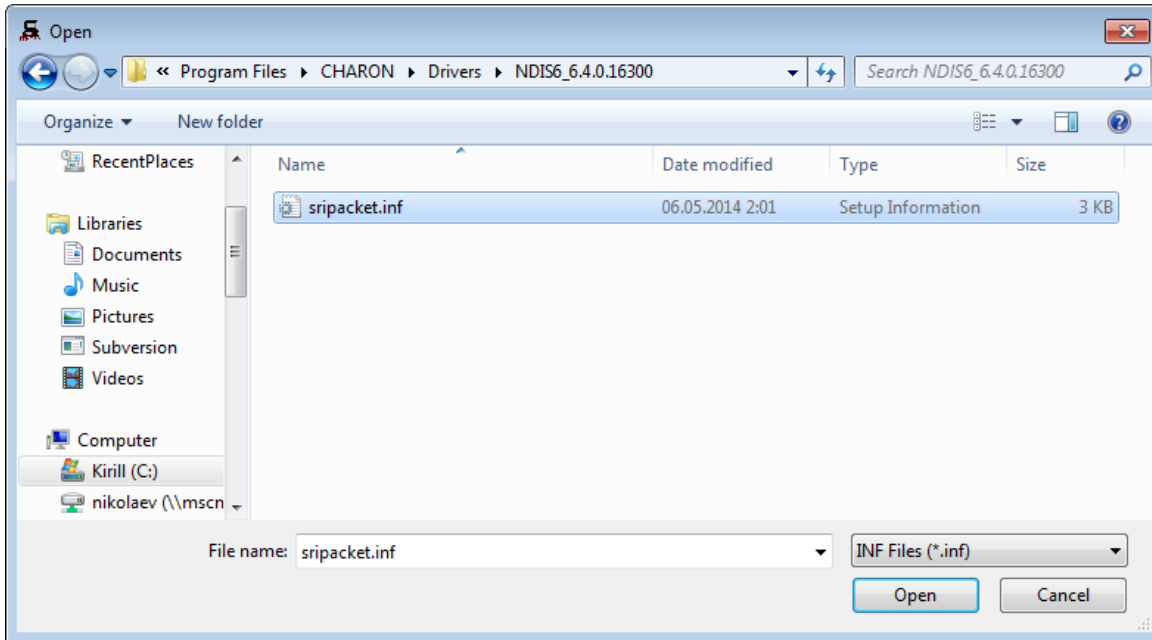
Select "Install/Upgrade CHARON NDIS driver" checkbox and press "Select":



Press "Browse":

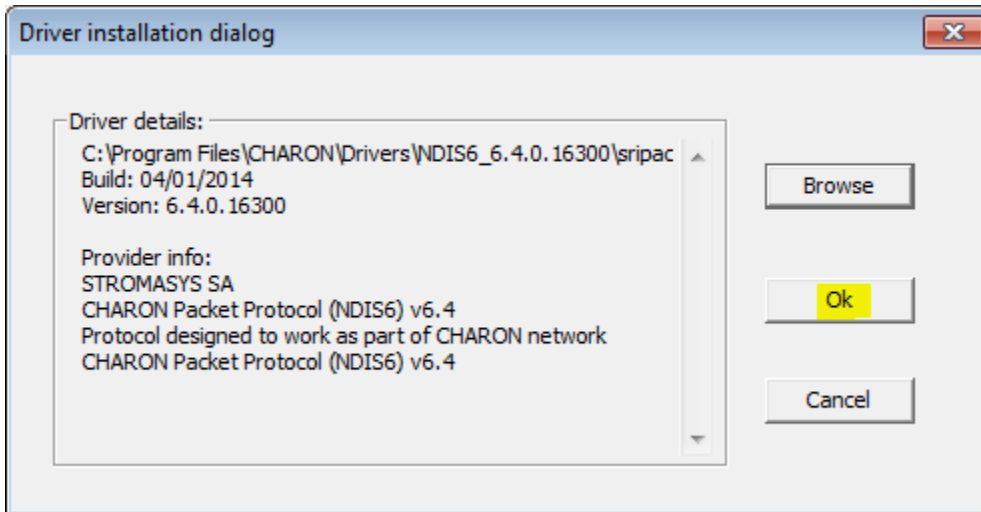


Browse for the target "sripacket.inf" file (by default it is located in the "C:\Program Files\CHARON\Drivers\NDIS6\_X.X.X.XXXXX" directory), select it and press "Open":

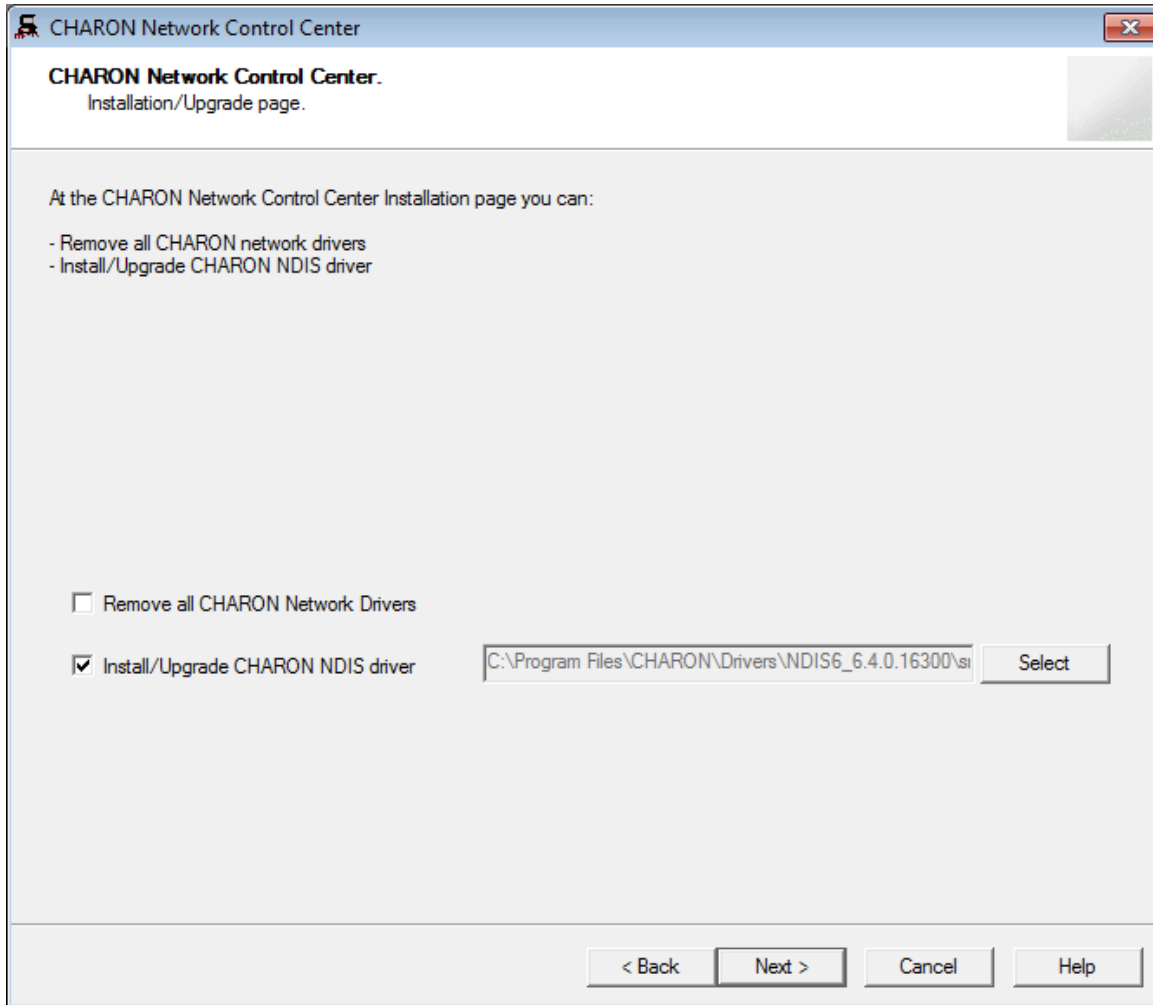


If the CHARON network driver has been acquired directly from STROMASYS, put it in a temporary directory and choose the "sripacket.inf" file from this directory in the dialog above.

Review the version of the driver and, if it is correct, press "Ok":



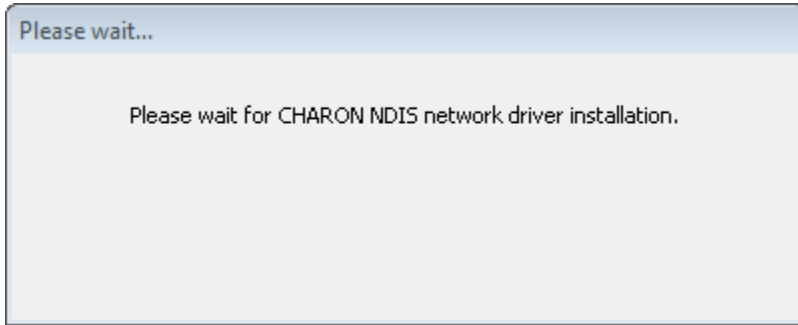
The Network Control Center will display the previous dialog with the path to the CHARON network driver. Press "Next" in the popup below to continue:



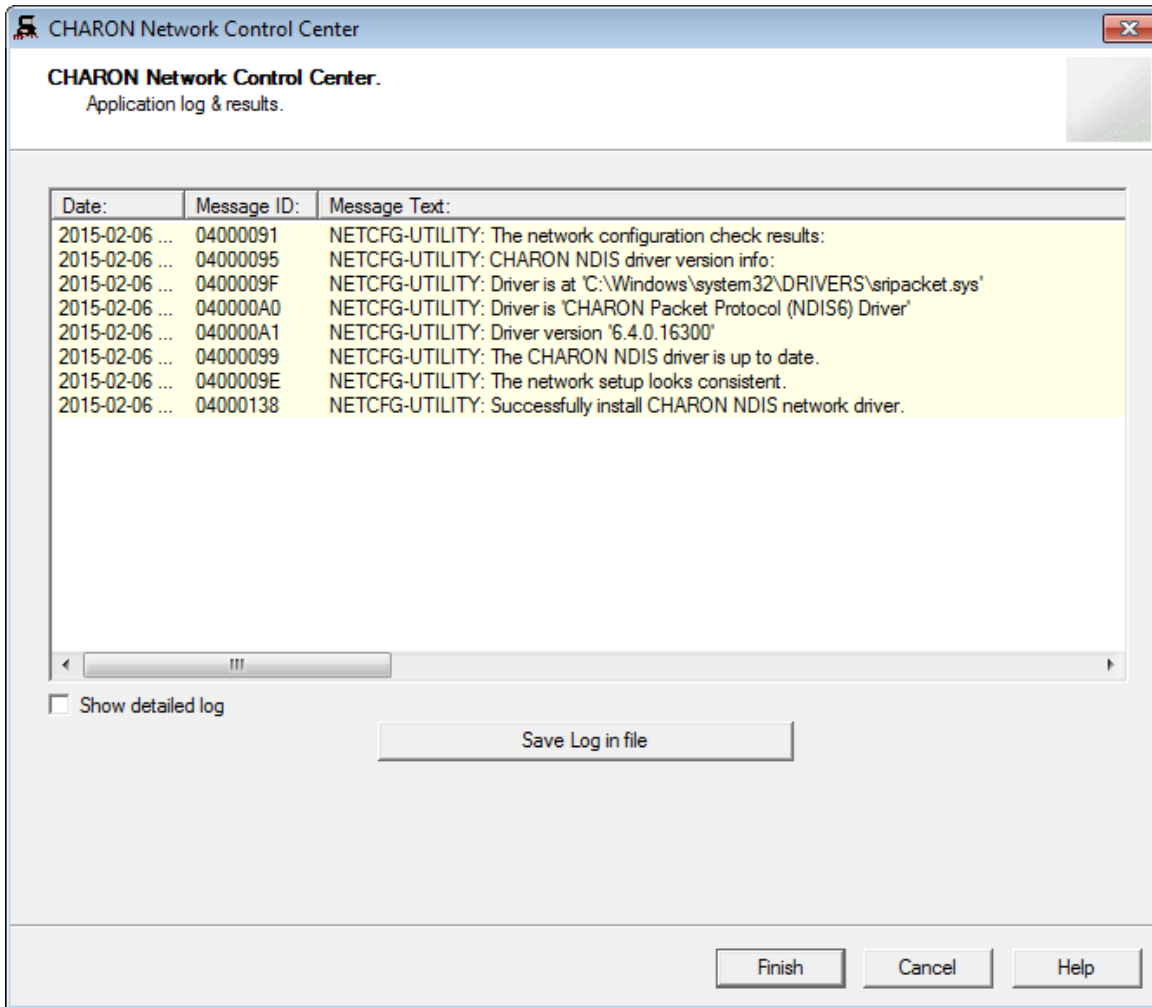
The utility will begin the CHARON network driver installation. If Windows Security asks you to confirm the driver installation, press "Install":



The driver installation will resume:



At the completion of the network installation procedure, the utility will display the following log:



Review the log, make sure it is correct. Save the log to a file by pressing "Save Log in file" in case the log file has to be sent to Stromasys Customer Support.

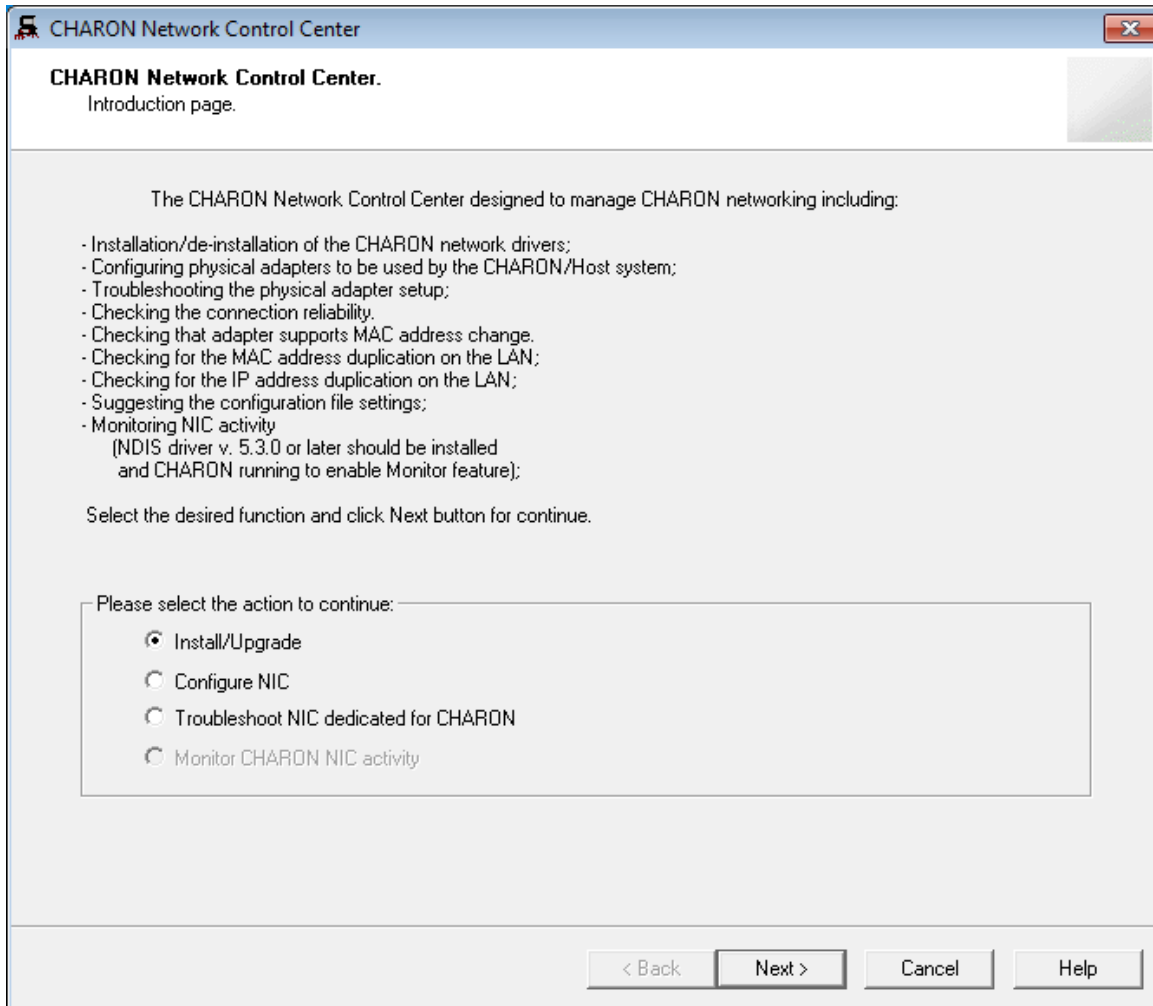
Select "Show detailed log" to display more detail.

Press "Finish" to exit.

[Back to Table of Contents](#)

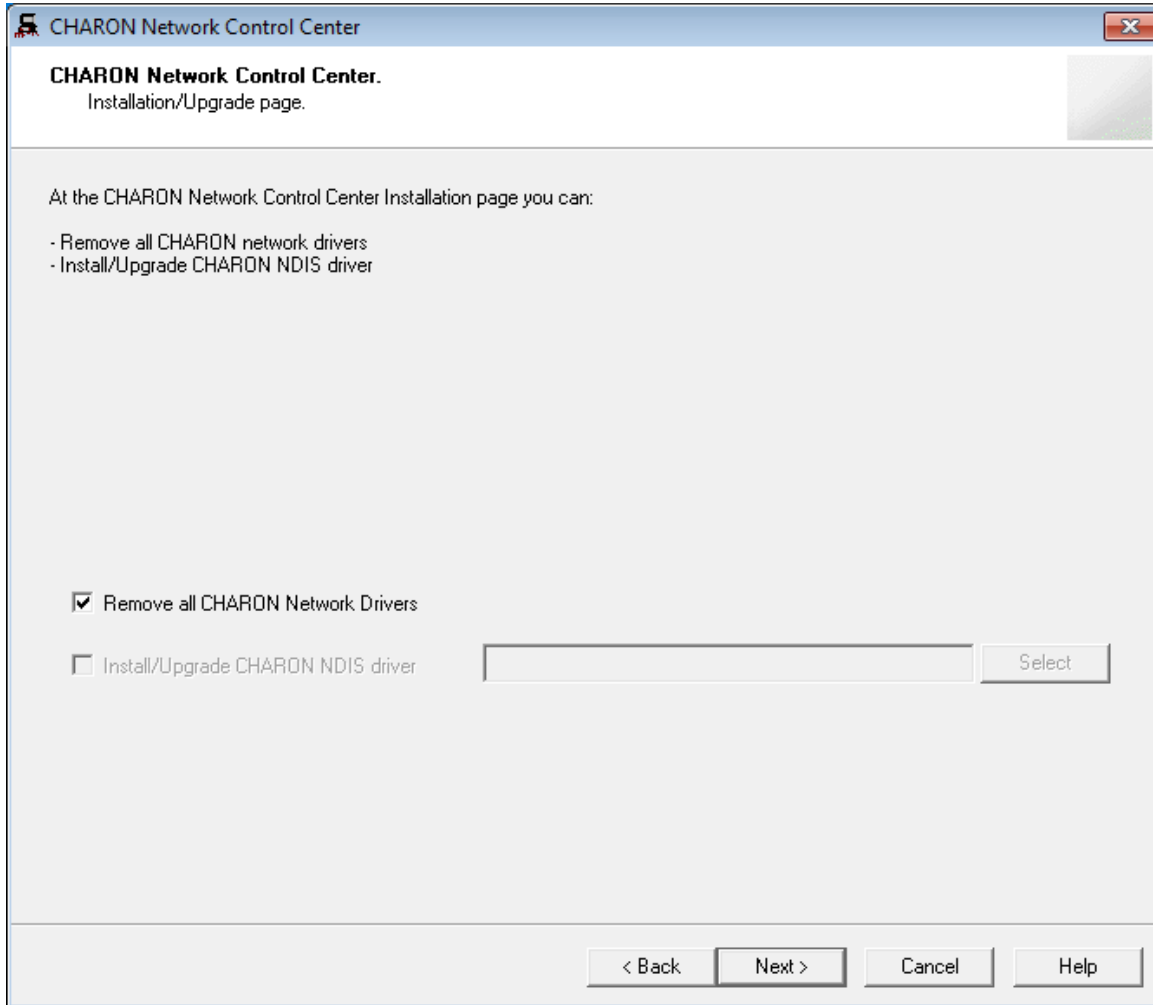
## Deinstallation of the CHARON network driver

Start the utility and select "Install/Upgrade"; press "Next":

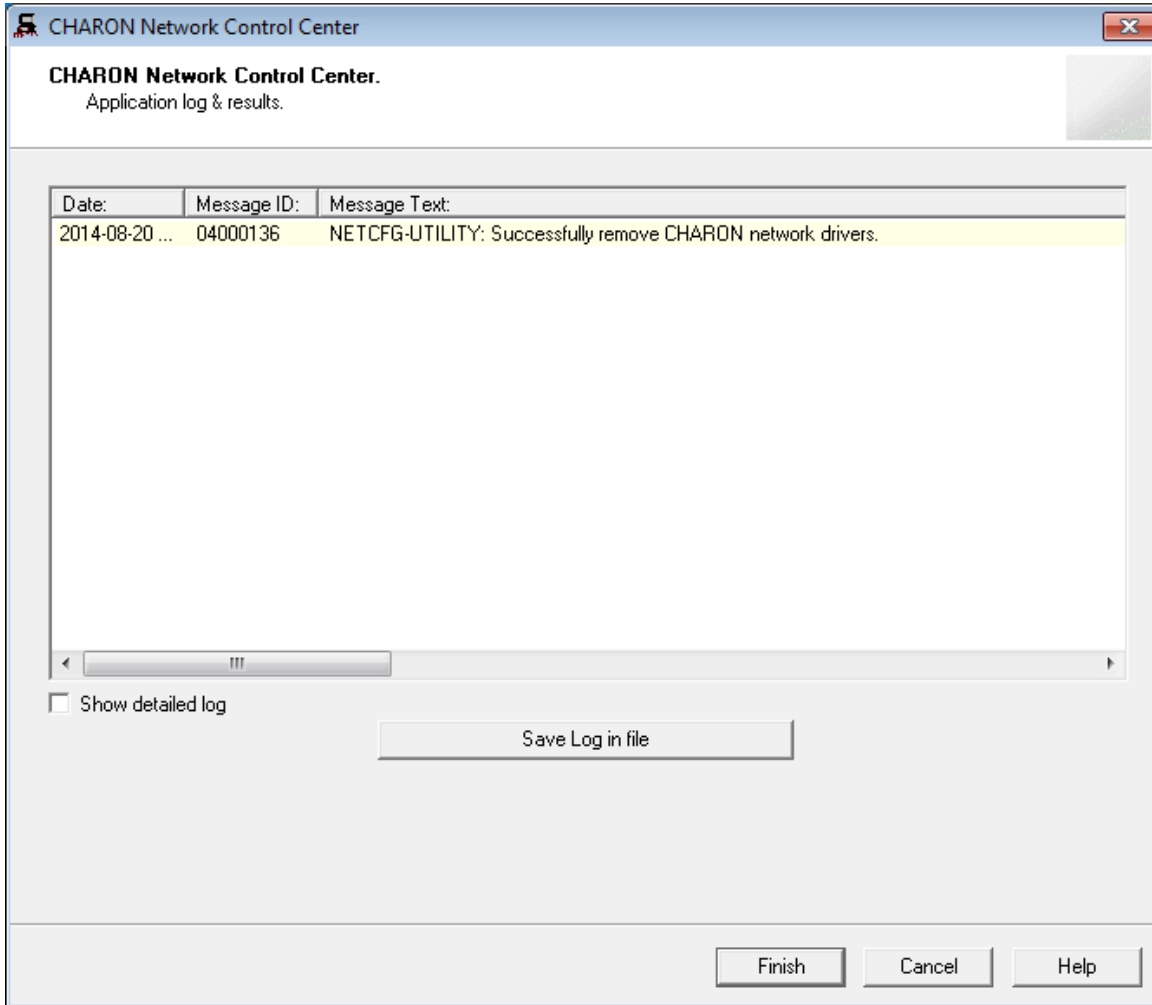




Select "Remove all CHARON Network Drivers" and press "Next":

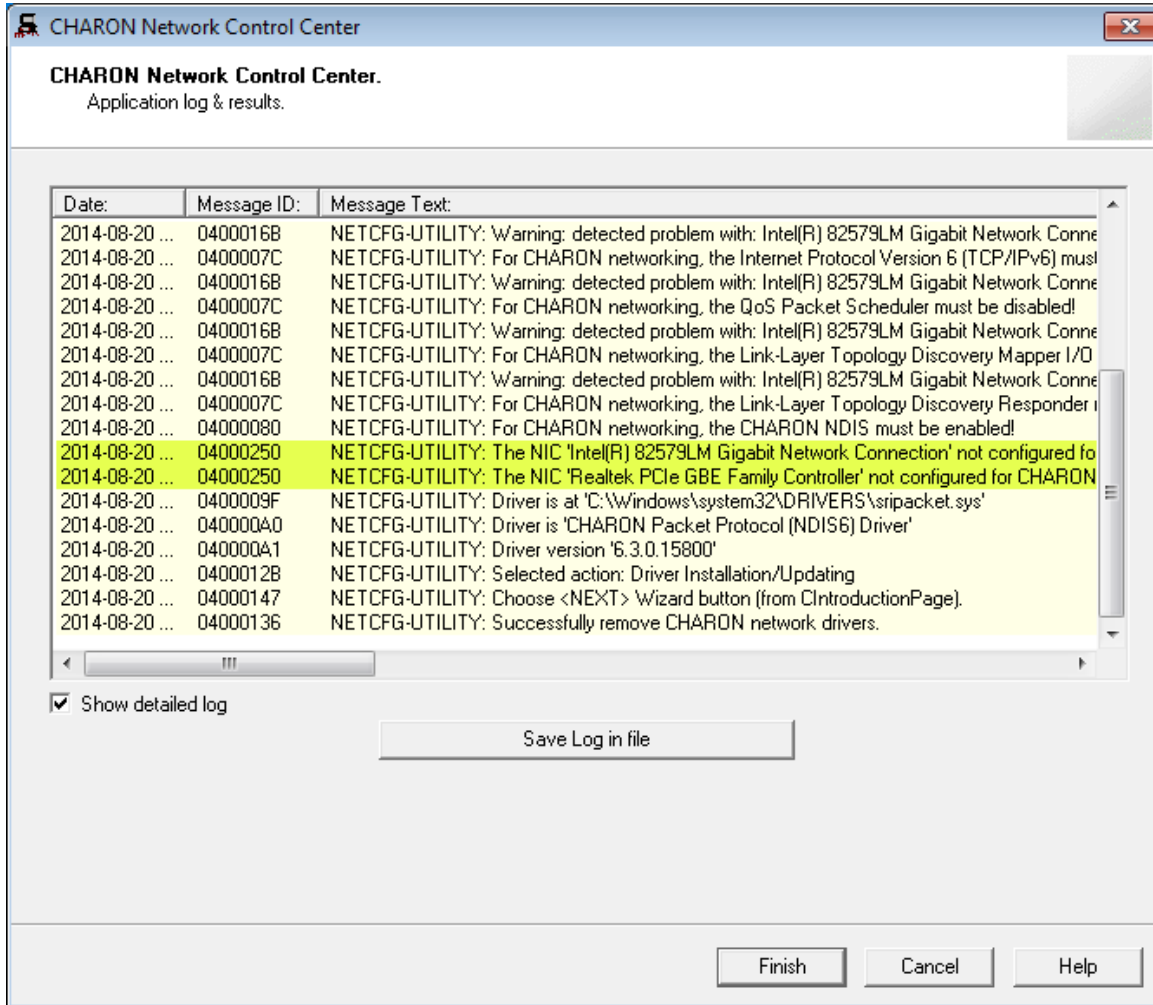


The utility will report that the CHARON drivers have been removed:



Review the log for errors. Save the log to a file by pressing "Save Log in file" in case the log file has to be sent to Stromasys Customer Support.

Select "Show detailed log" to display more detail.

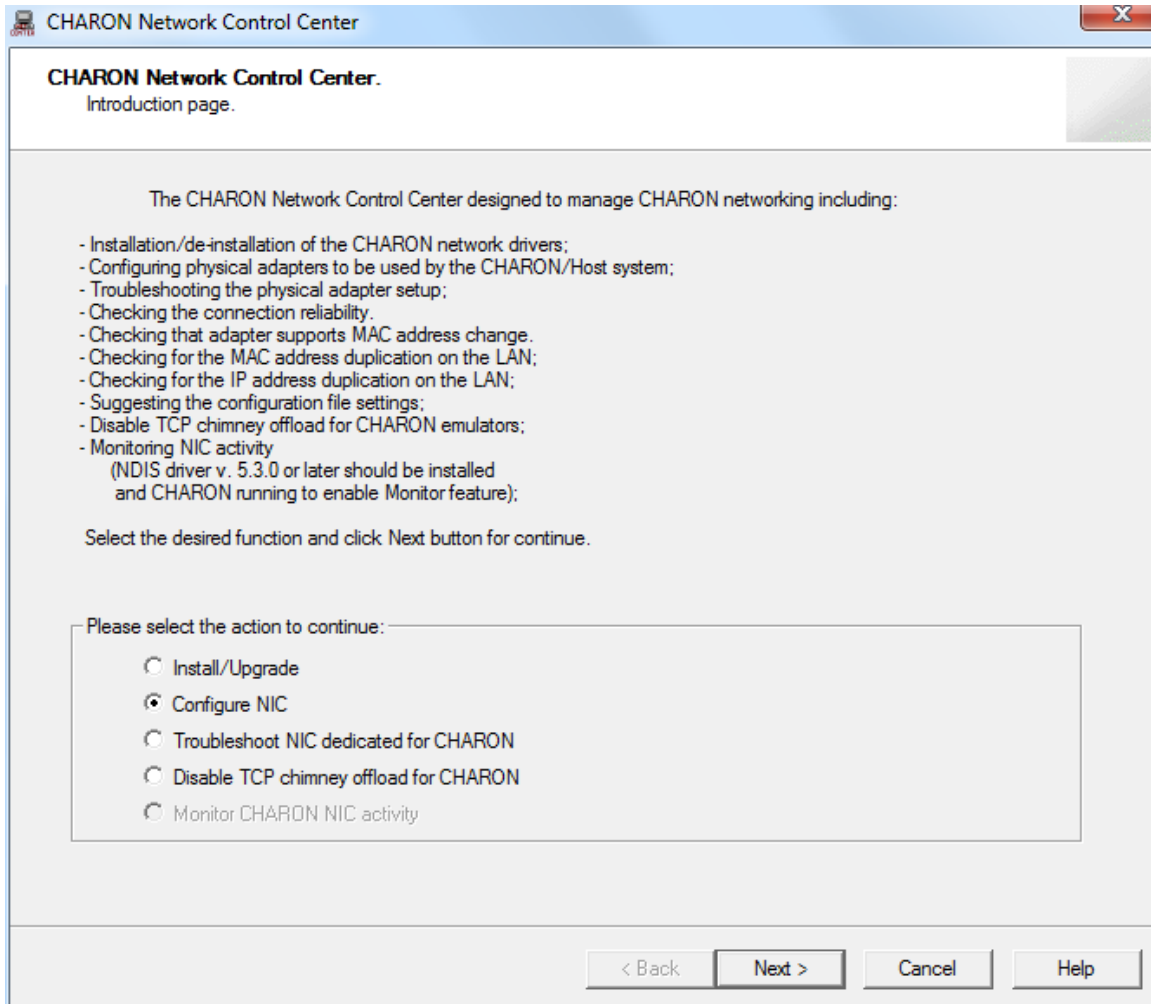


Press "Finish" to exit.

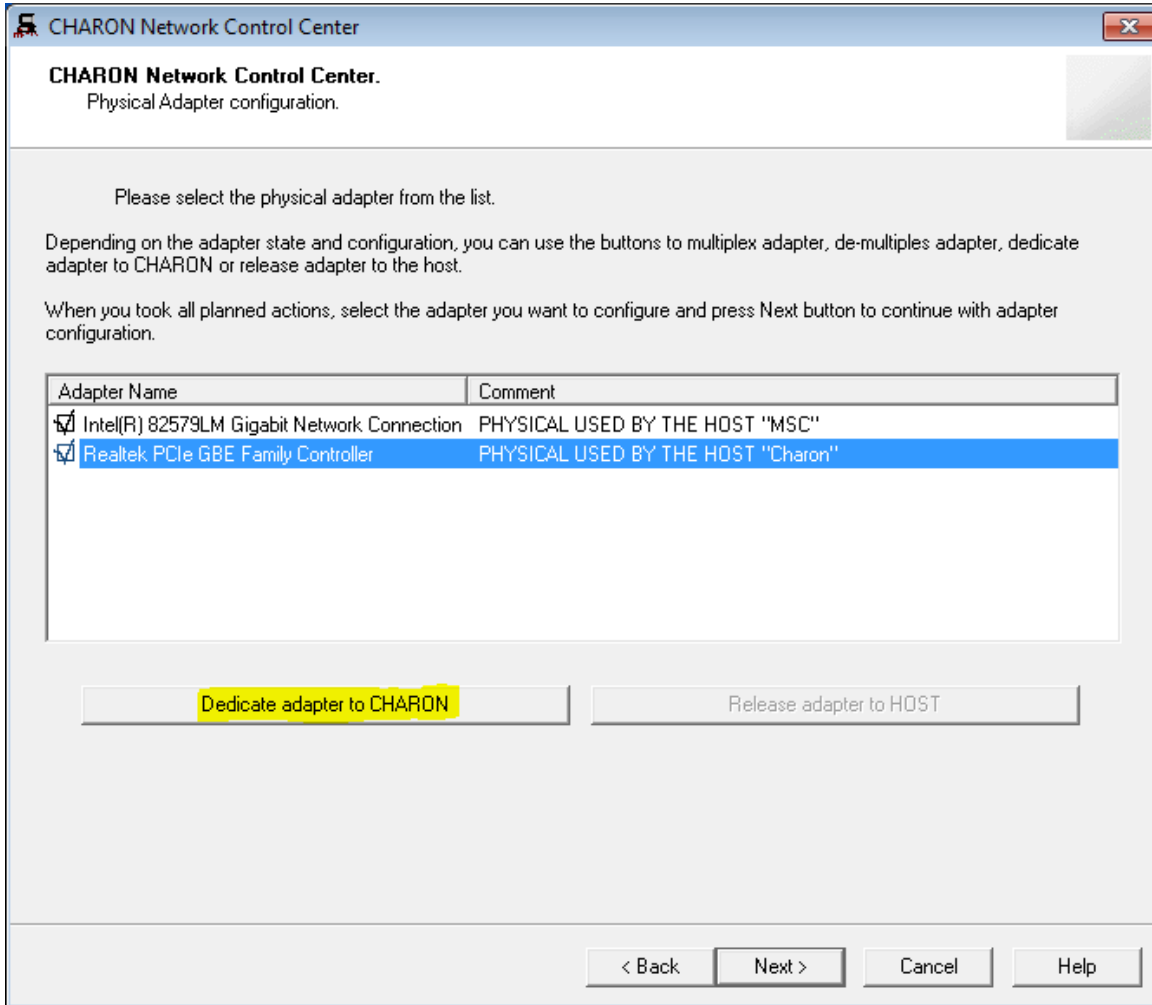
[Back to Table of Contents](#)

## Configuring the host network interfaces for CHARON

Start the utility and select "Configure NIC"; press "Next":

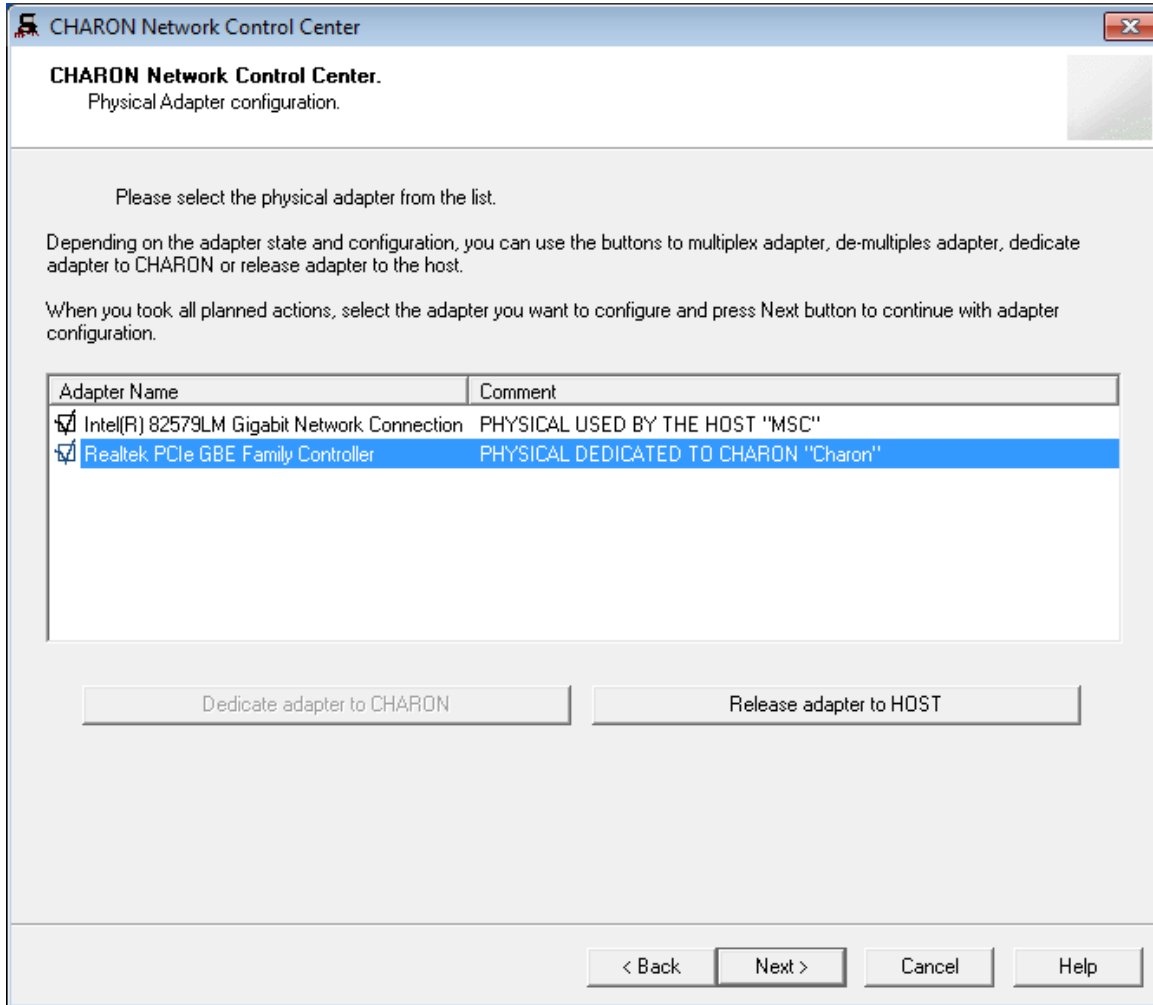


Select the host interface to be dedicated to CHARON (in the example below it is "Charon"), press "Dedicate adapter to CHARON":

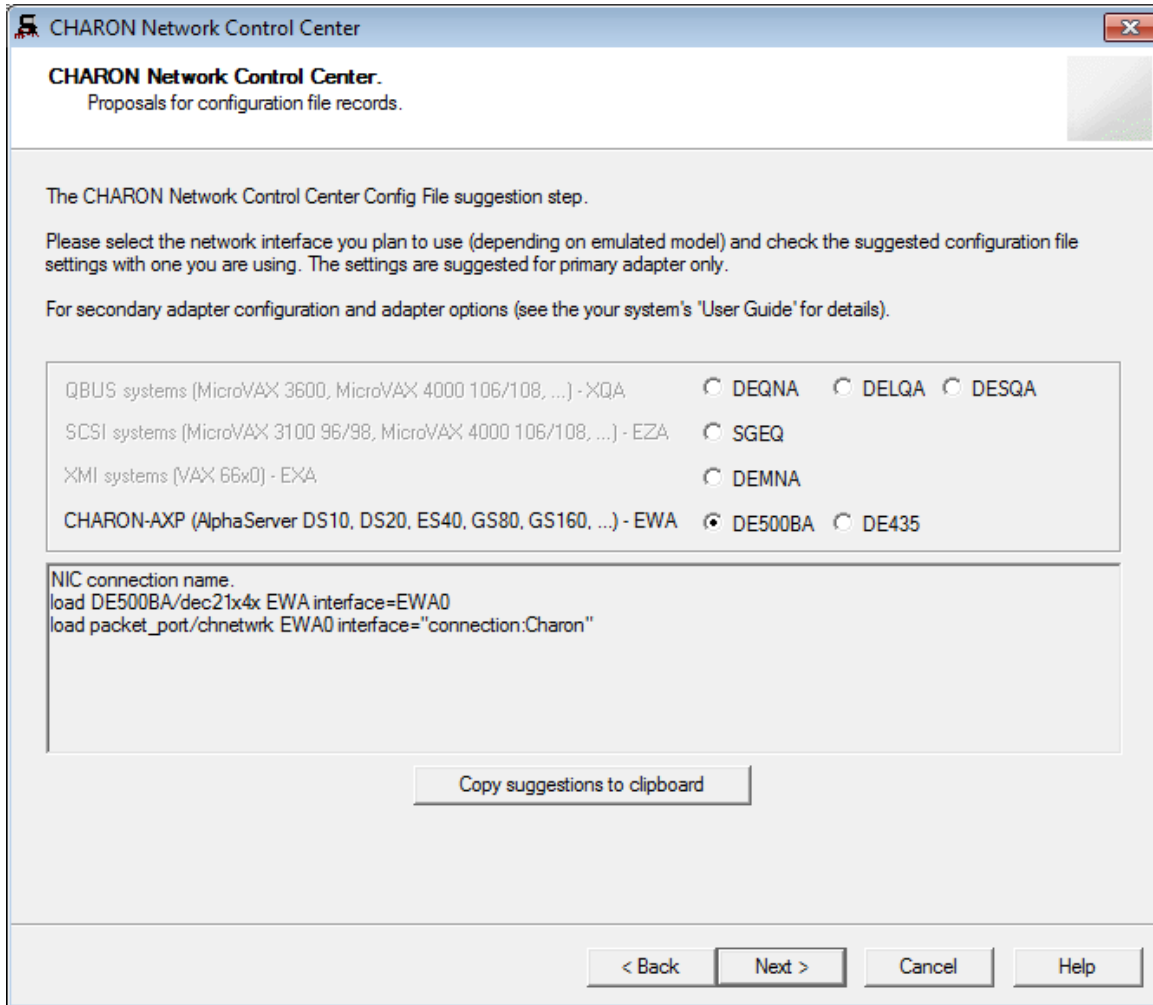


Install and configure VLAN adapters, according to the vendor's User's Guide, if required. Select the VLAN adapter in the dialog box in the above example. A VLAN adapter is not configured differently, the same procedure should be followed.

The Network Control Center will dedicate the selected adapter to CHARON. Press "Next".



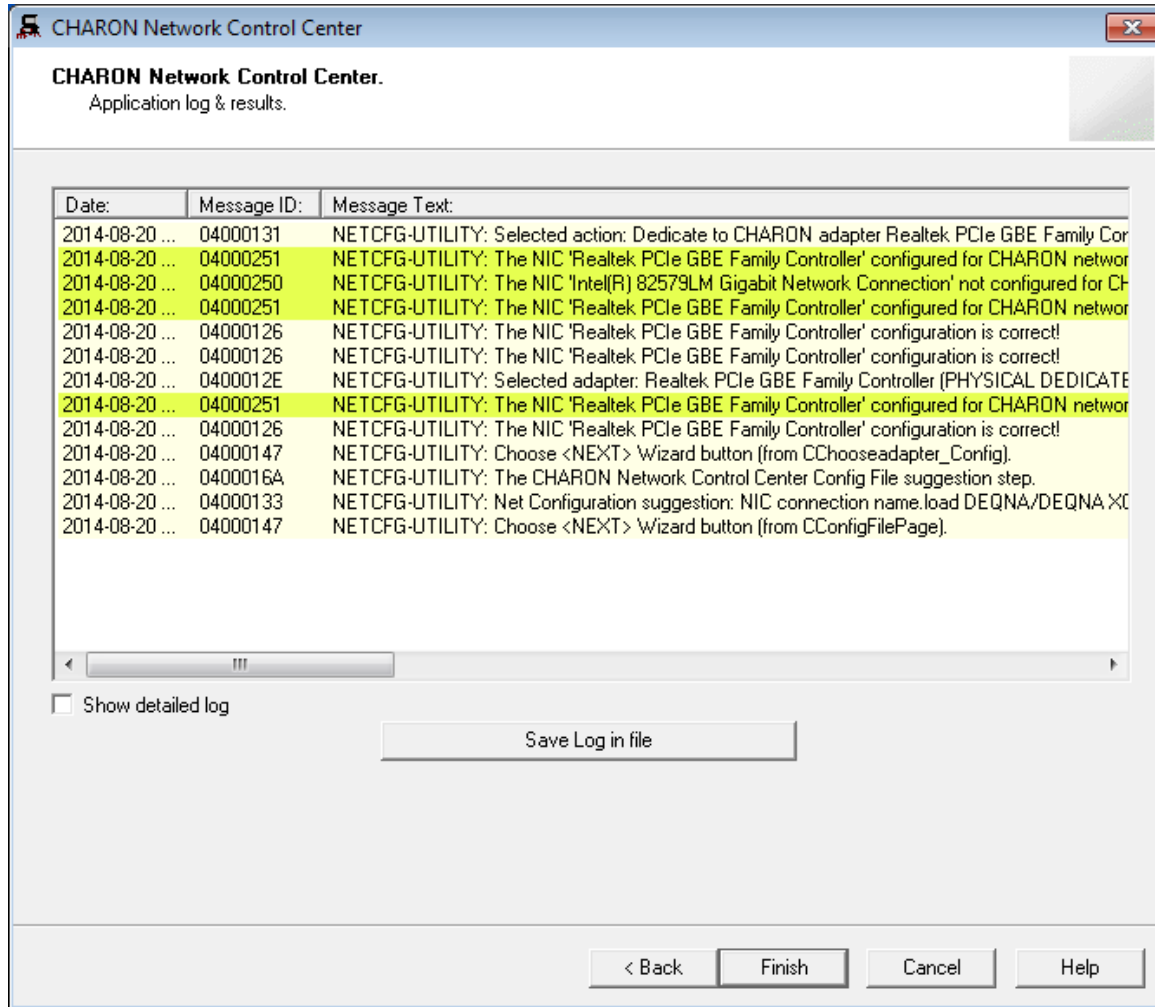
The next dialog helps to define the correct lines for the CHARON configuration file. Select the target emulated network adapter and press "Copy suggestions to clipboard":



It is possible to paste the content of the clipboard to the CHARON configuration file as shown in the example below:

```
...
load DE500BA/dec21x4x EWA interface=EWA0
load packet_port/chnetwrk EWA0 interface="(disabled)"
load packet_port/chnetwrk EWA0 interface="connection:Charon"
...
```

Press "Next" to see the log file:



Review the log for errors. Save the log to a file by pressing "Save Log in file" in case the log file has to be sent to Stromasys Customer Support.

Select "Show detailed log" to display more detail.

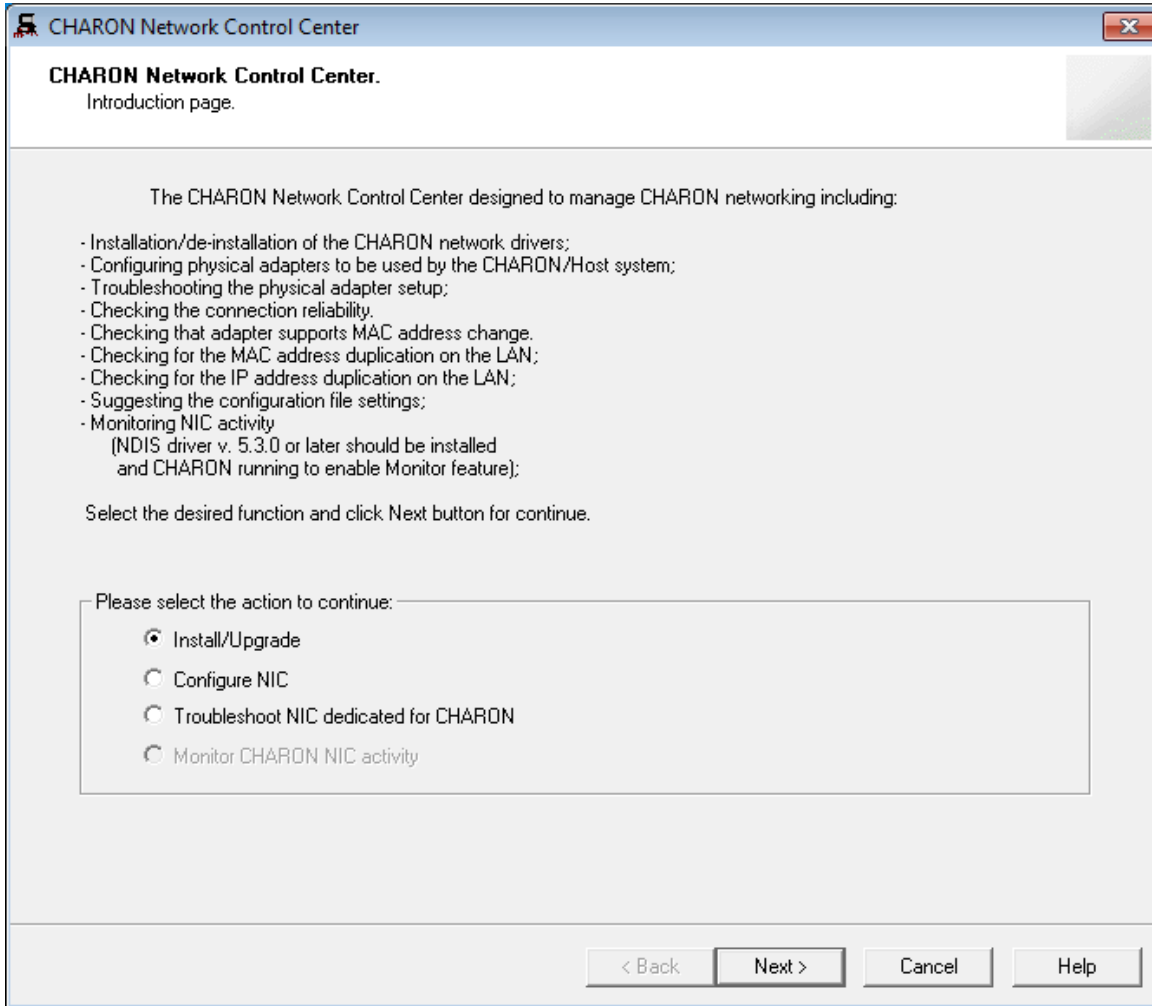
Press "Finish" to exit.

[Back to Table of Contents](#)

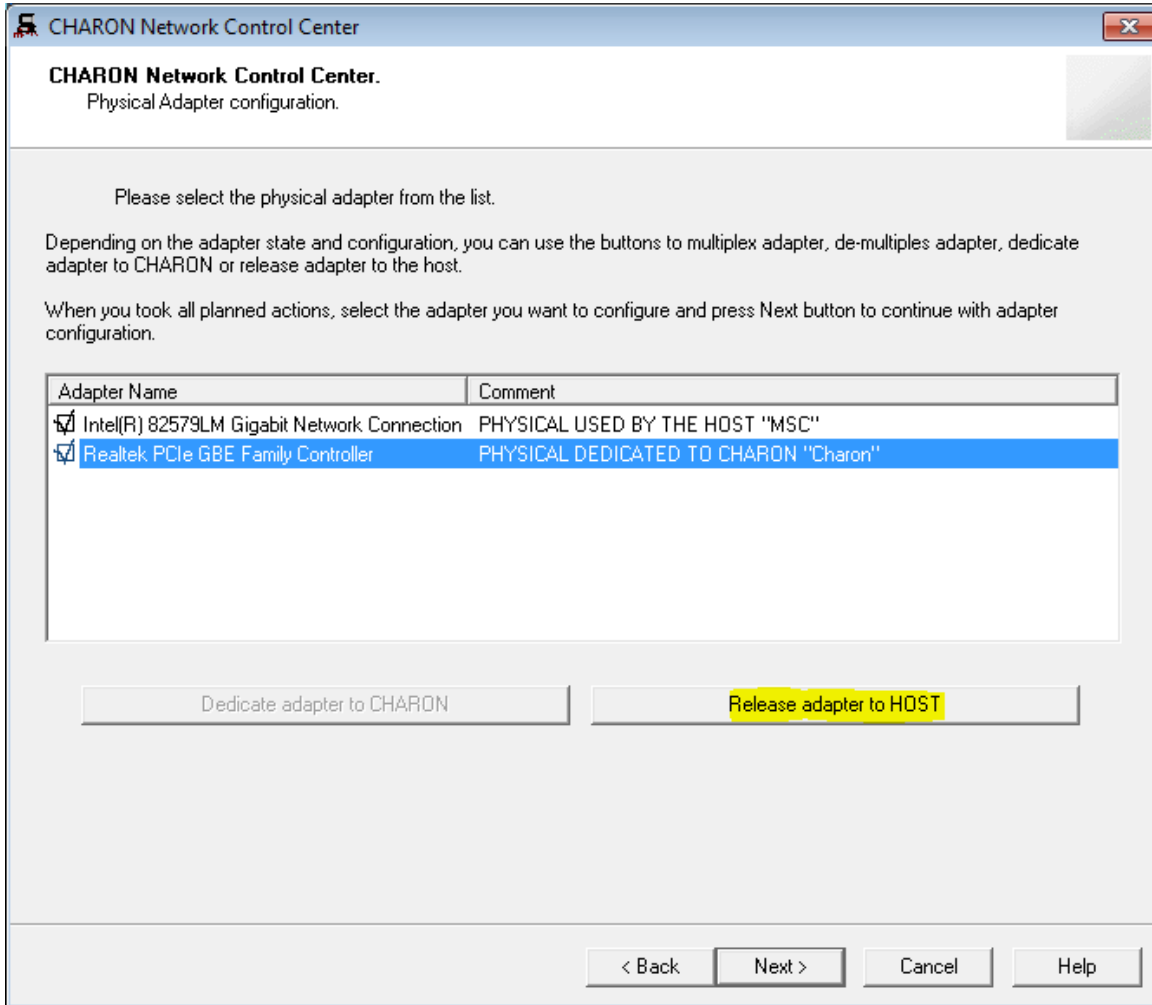


## Release of the host network interfaces

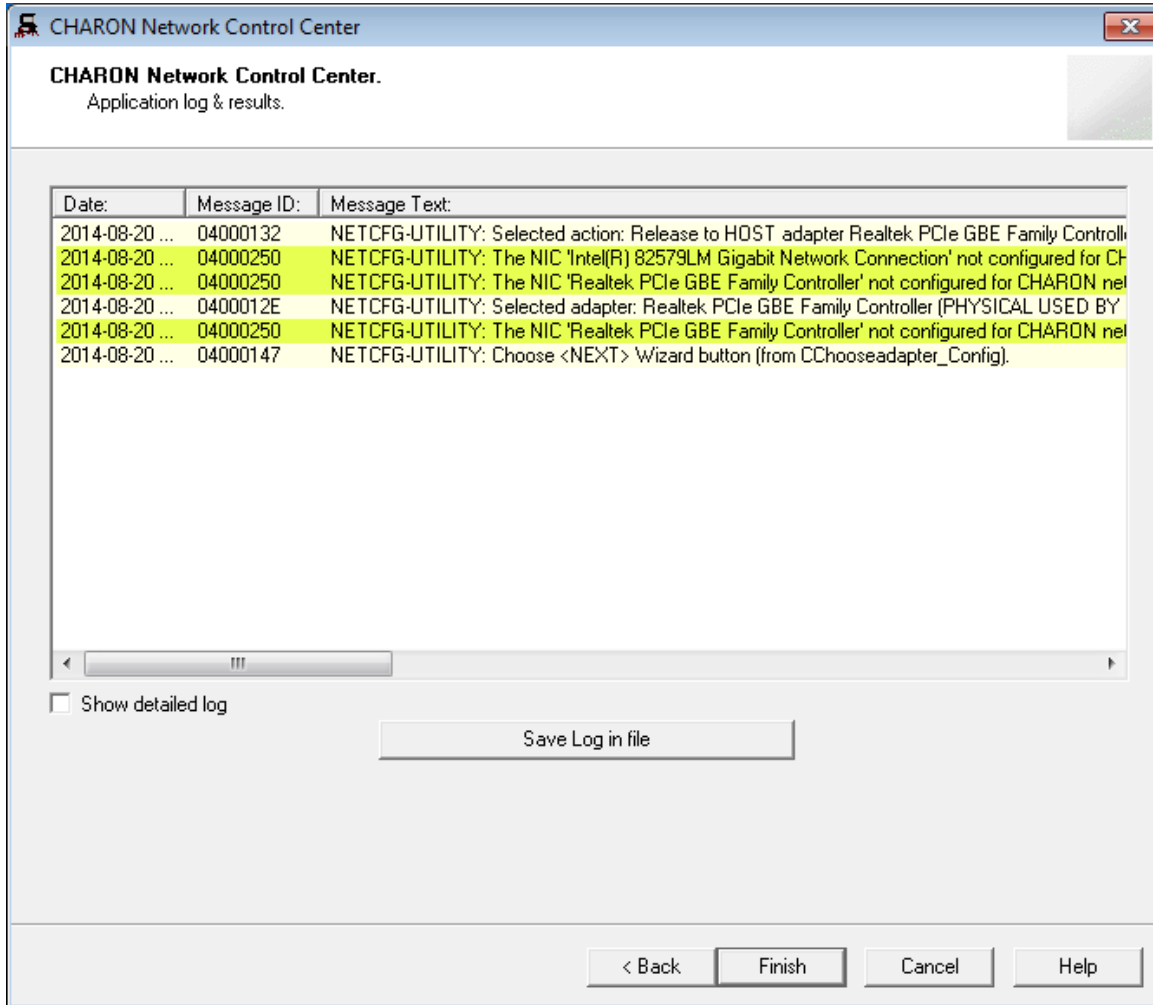
Start the utility and select "Configure NIC"; press "Next":



Select the interface to be released back to the host (in the example below it is "Charon"), press "Release adapter to HOST":



Press "Next" and review the log of this operation:



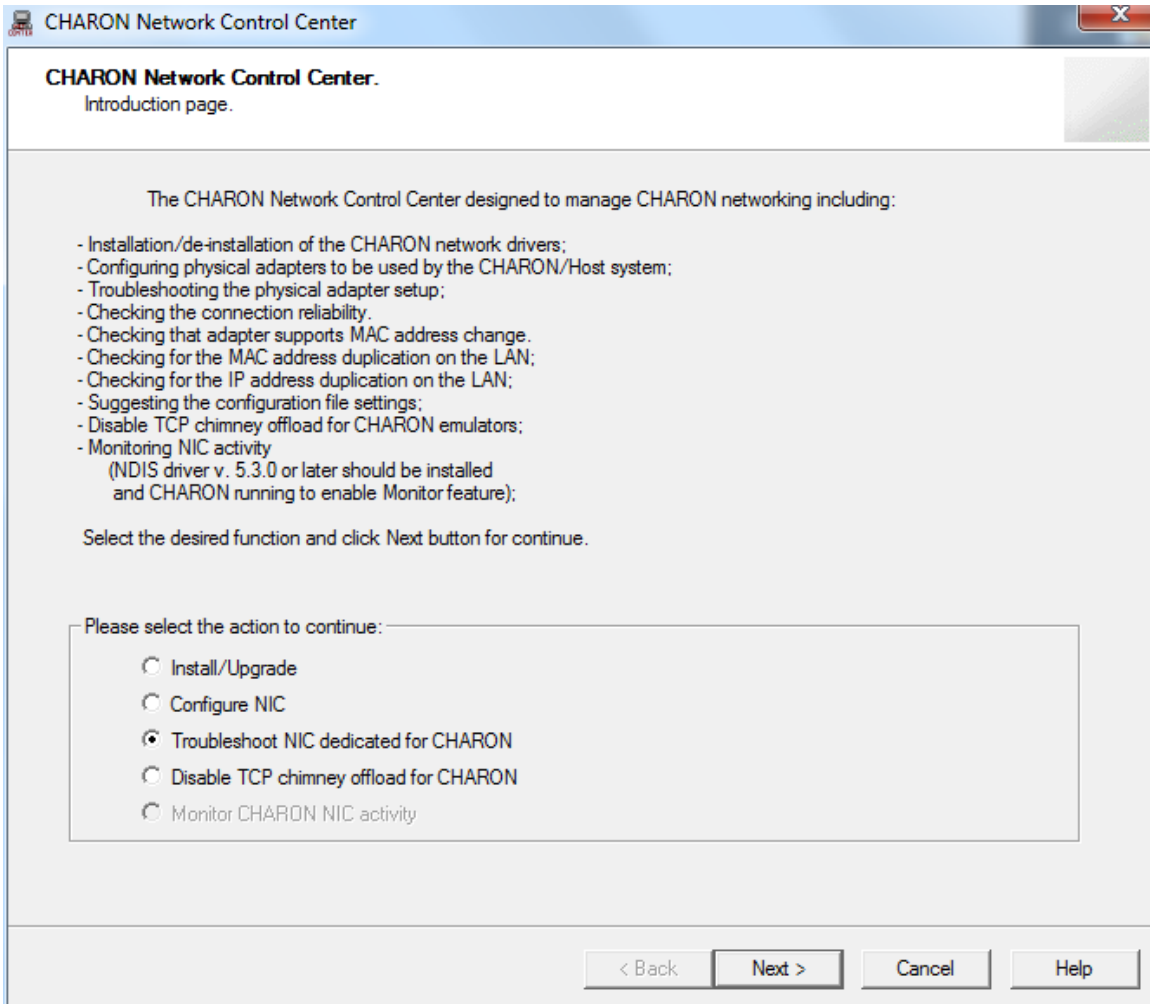
Select "Show detailed log" to display more detail.

Press "Finish" to exit.

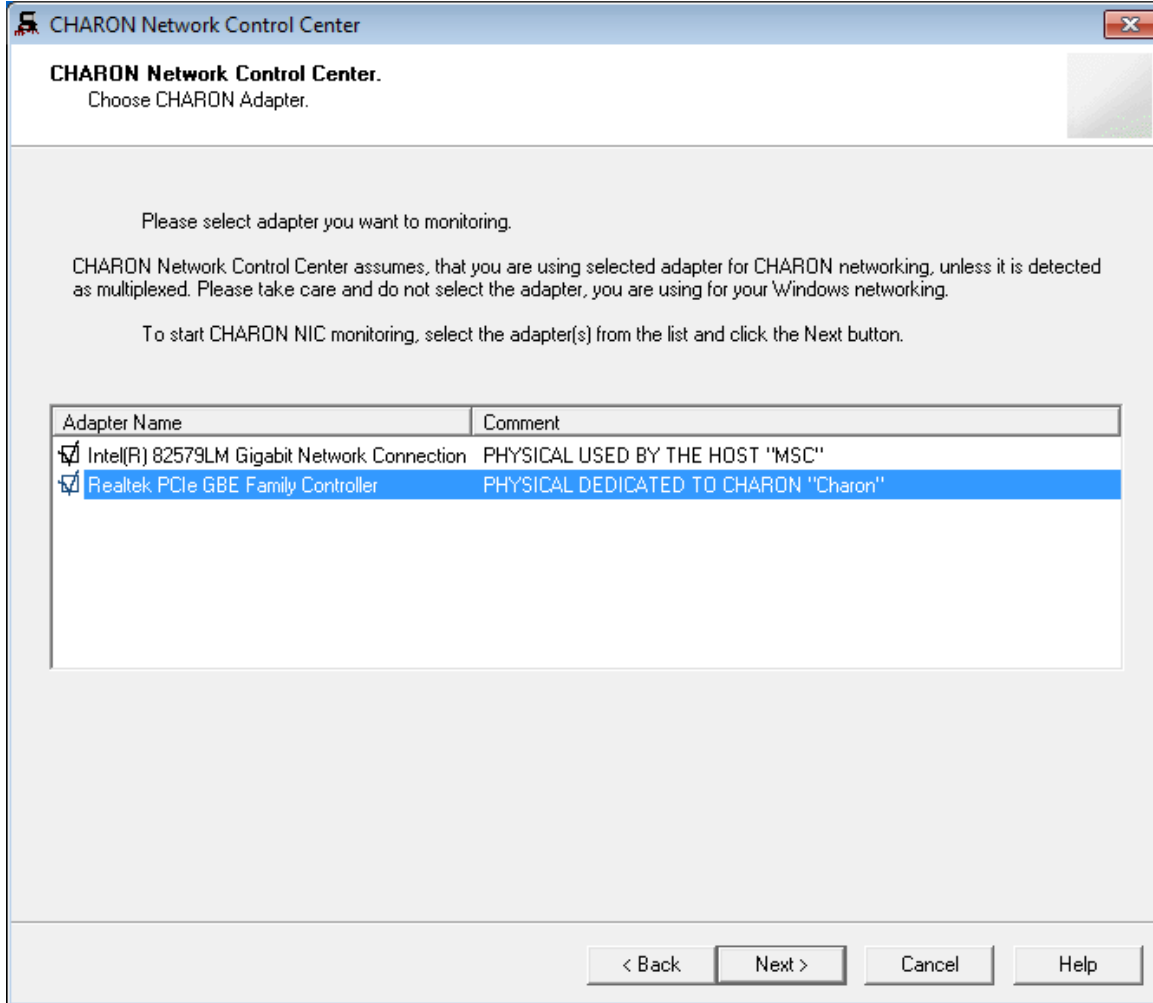
[Back to Table of Contents](#)

## Troubleshooting the CHARON network interfaces configuration

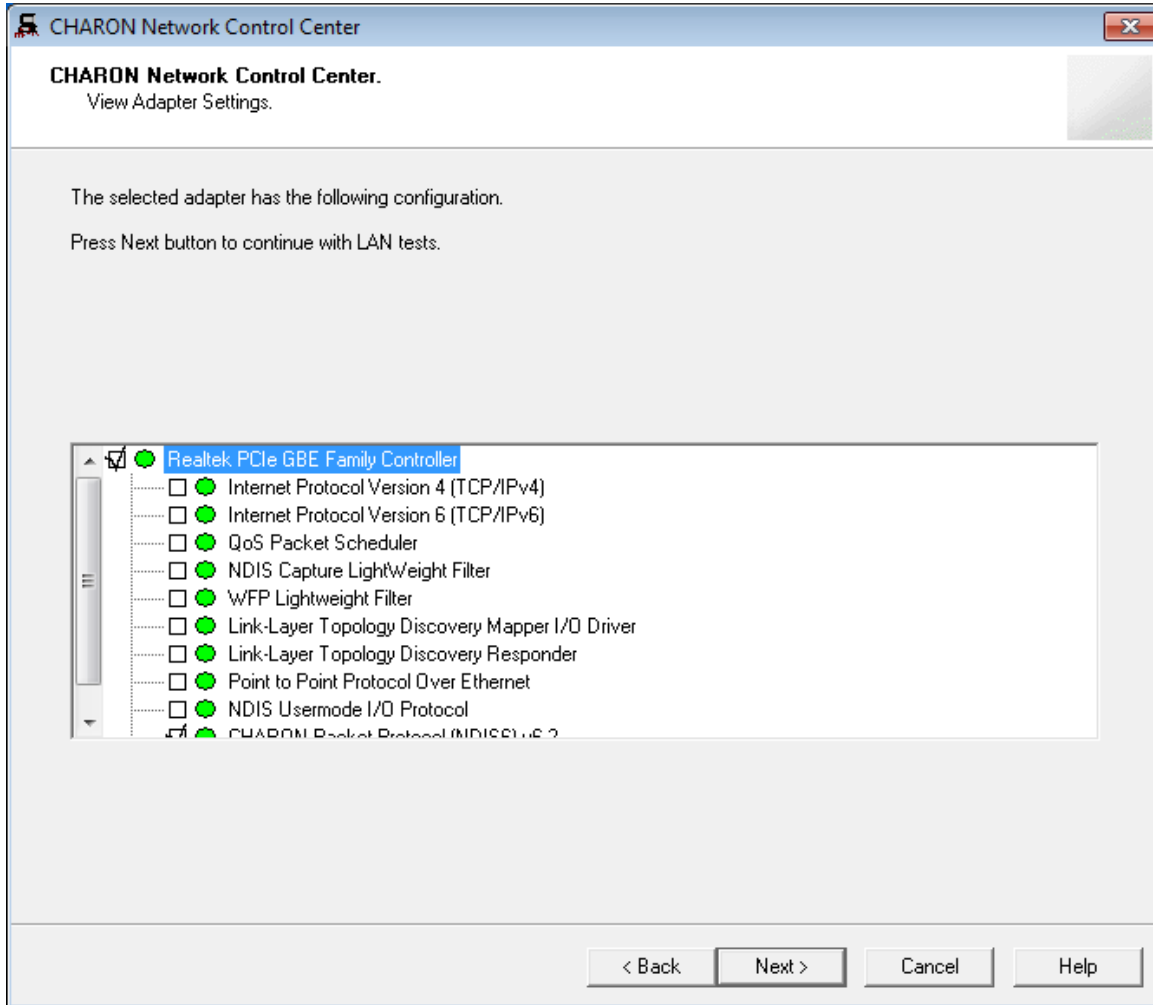
Start the utility and select "Troubleshoot NIC dedicated for CHARON"; press "Next":



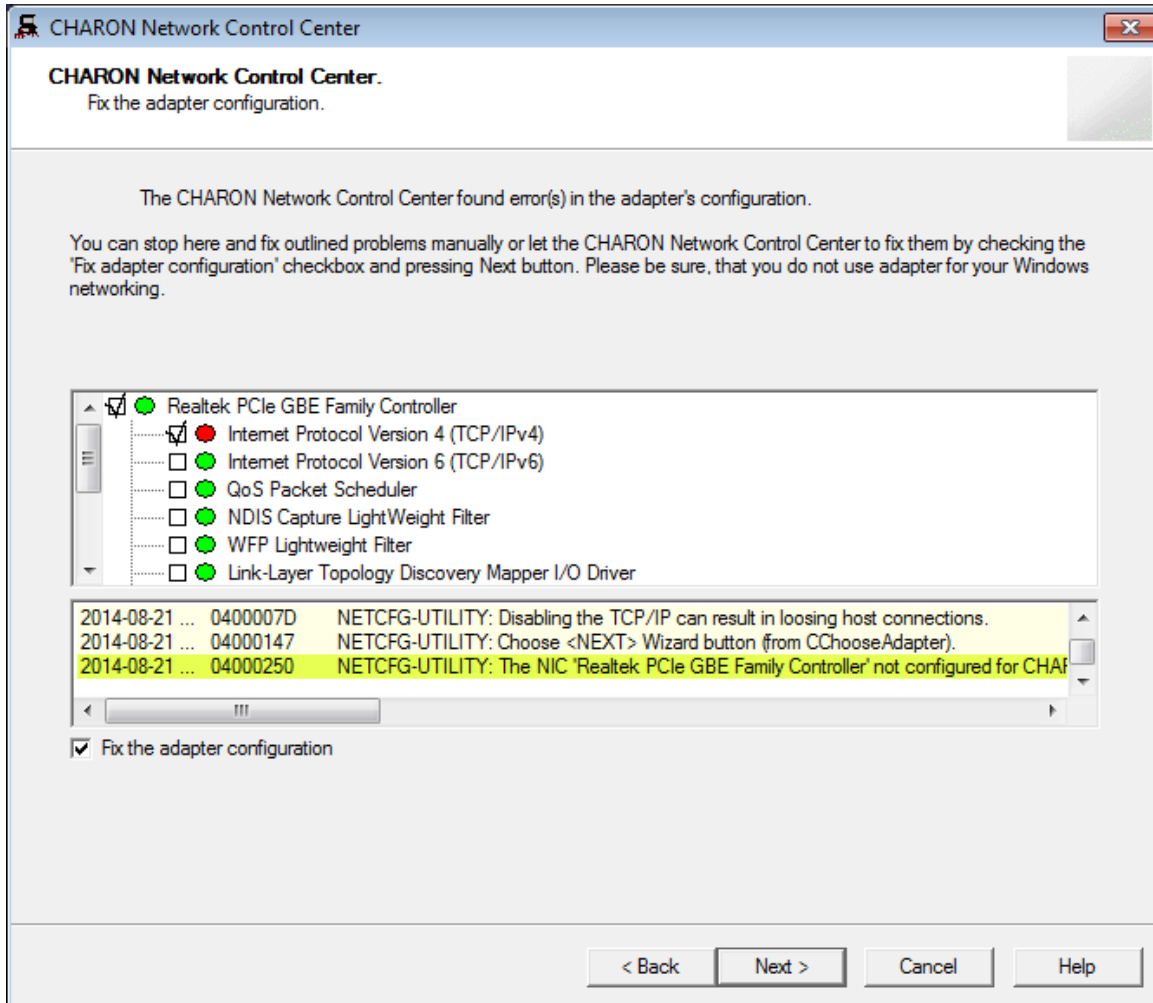
Select the target interface and press "Next":



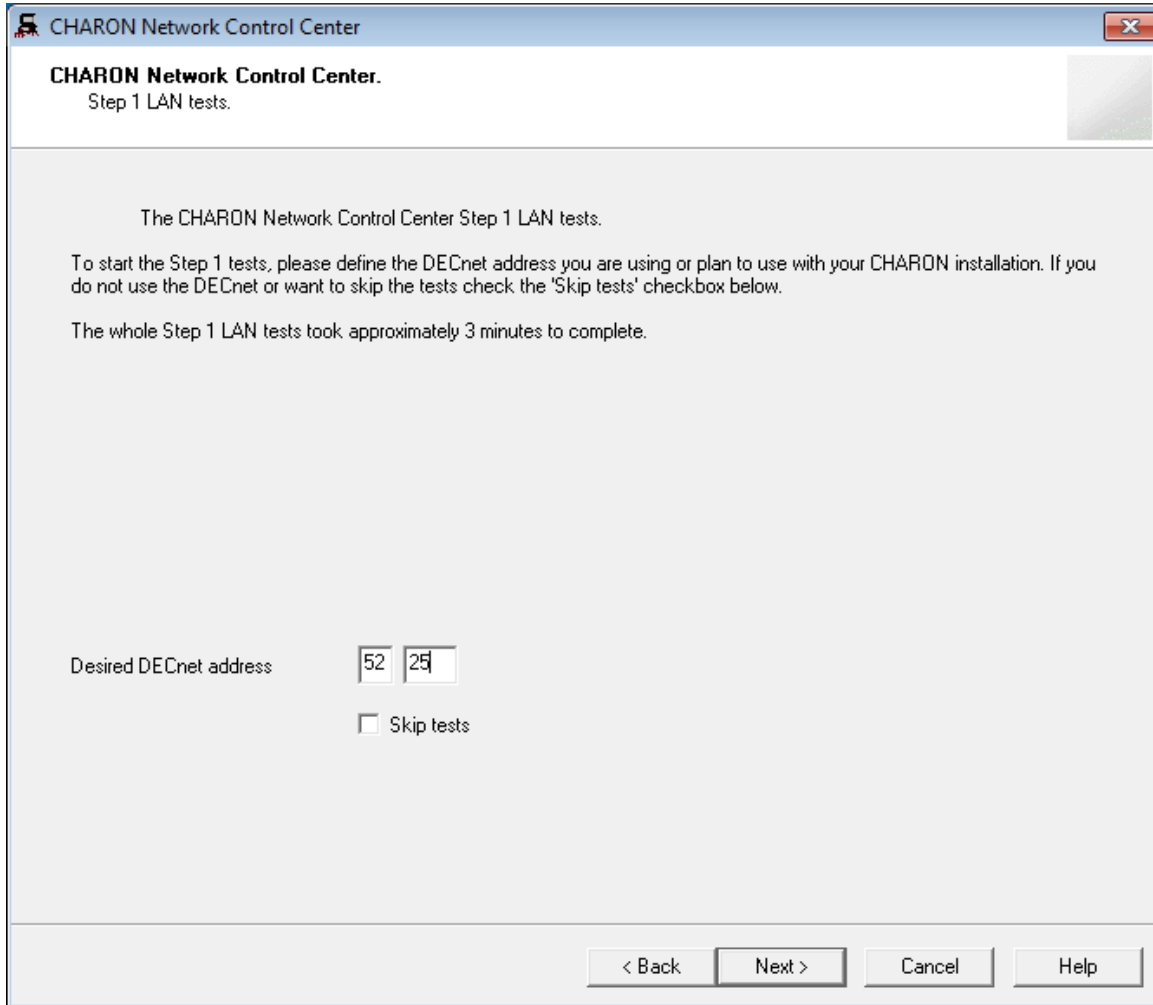
Review the status of the interface:



In the example below, there is an issue with the host TCP/IP being enabled on the CHARON interface. Review the problem description, press "Fix the adapter configuration" and press "Next".

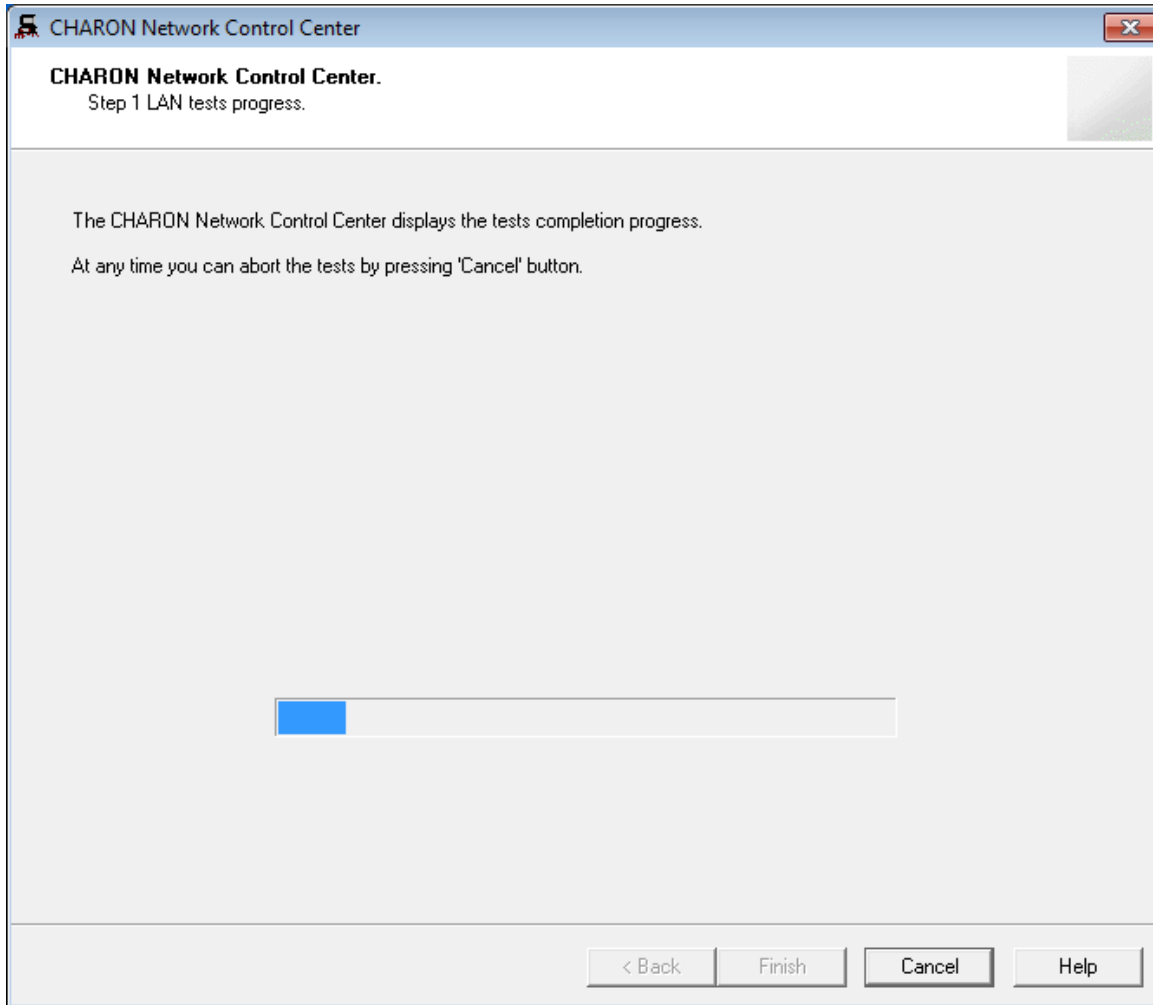


The utility will fix any issues and report a good status for the interface. Press "Next", the following dialog will appear:

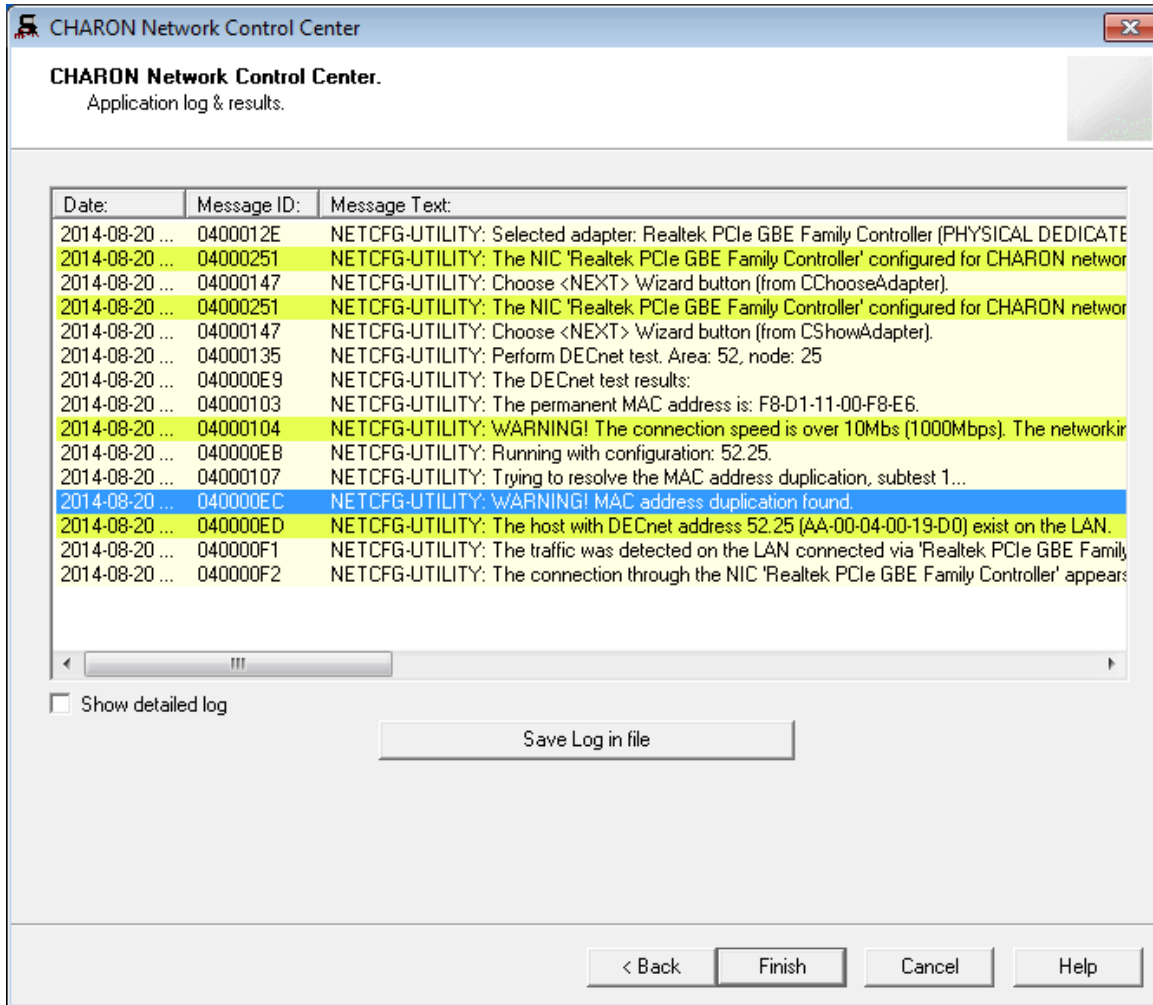




The Network Control Center offers to check whether the DECnet address, to be used by CHARON, is unique on the network in this step. Enter the desired DECnet address, for the CHARON guest, and press "Next":



Once the checking is completed, if there are any issues, the utility will abort the test and display a log containing information about the issues found:



If no issues are found, the utility will offer to check the CHARON TCP/IP networking the same way:

The screenshot shows a Windows-style dialog box titled "CHARON Network Control Center" with a close button in the top right corner. The main content area is titled "CHARON Network Control Center. Step 2 LAN tests." and contains the following text:

The CHARON Network Control Center Step 2 LAN tests.

To start the Step 2 tests, please define the IP address and subnet mask you are using or plan to use with your CHARON installation or your LAN's subnet address and subnet mask. If you want to skip that tests check the 'Skip tests' checkbox below.

WARNING: The IP address duplication test can result in the loosing connections by the Windows host that using the IP address you specify.

The whole Step 2 tests duration depends on the size of you subnet. Typically, for class C networks, or subnetted networks with amount of node ~256 the whole Step 2 tests took ~15 minutes maximum.

Below the text are two input fields: "Desired IP address (or subnet address)" and "Subnet mask", each with a text box containing three dots. Below these is a checkbox labeled "Skip tests" which is currently unchecked.

At the bottom of the dialog box are four buttons: "< Back", "Next >", "Cancel", and "Help".

Enter the IP address and subnet mask to be used by the CHARON guest, then press "Next":

The CHARON Network Control Center Step 2 LAN tests.

To start the Step 2 tests, please define the IP address and subnet mask you are using or plan to use with your CHARON installation or your LAN's subnet address and subnet mask. If you want to skip that tests check the 'Skip tests' checkbox below.

WARNING: The IP address duplication test can result in the loosing connections by the Windows host that using the IP address you specify.

The whole Step 2 tests duration depends on the size of you subnet. Typically, for class C networks, or subnetted networks with amount of node ~256 the whole Step 2 tests took ~15 minutes maximum.

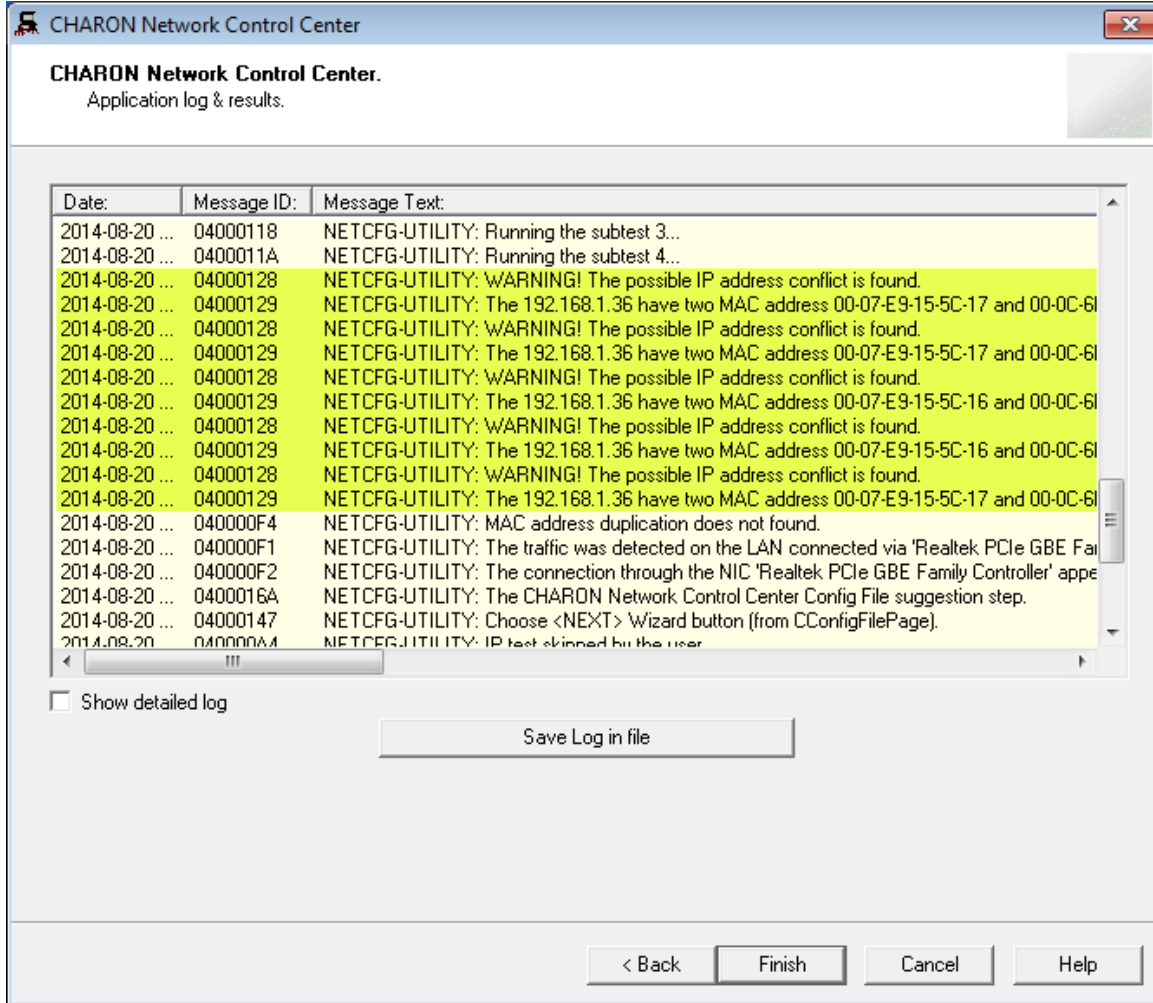
Desired IP address (or subnet address)

Subnet mask

Skip tests

< Back   Next >   Cancel   Help

The Network Control Center will display the log of the performed operations:



Select "Show detailed log" for more detail.

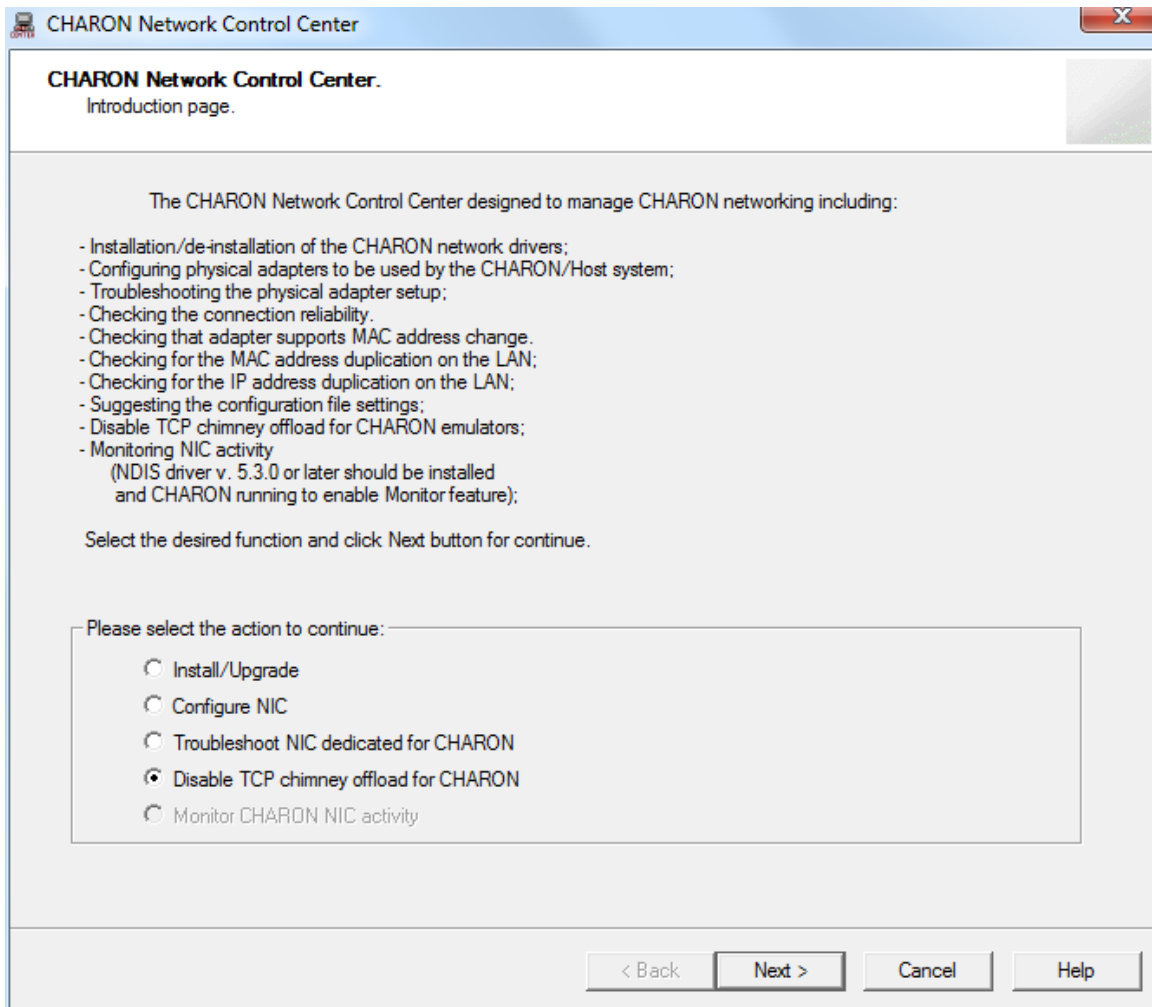
Press "Finish" to exit.

[Back to Table of Contents](#)

## Disable "offload" parameters

The Network Control Center is able to disable "offload" parameters for CHARON. It is highly recommended to disable them to avoid any problems with TCP/IP networking under CHARON.

Start the utility and select "Disable TCP chimney offload for CHARON"; press "Next":



The "offload" parameters will be reset for all CHARON emulators.

Press "Finish" to exit.

 Please note to restart the CHARON host to enable this settings.

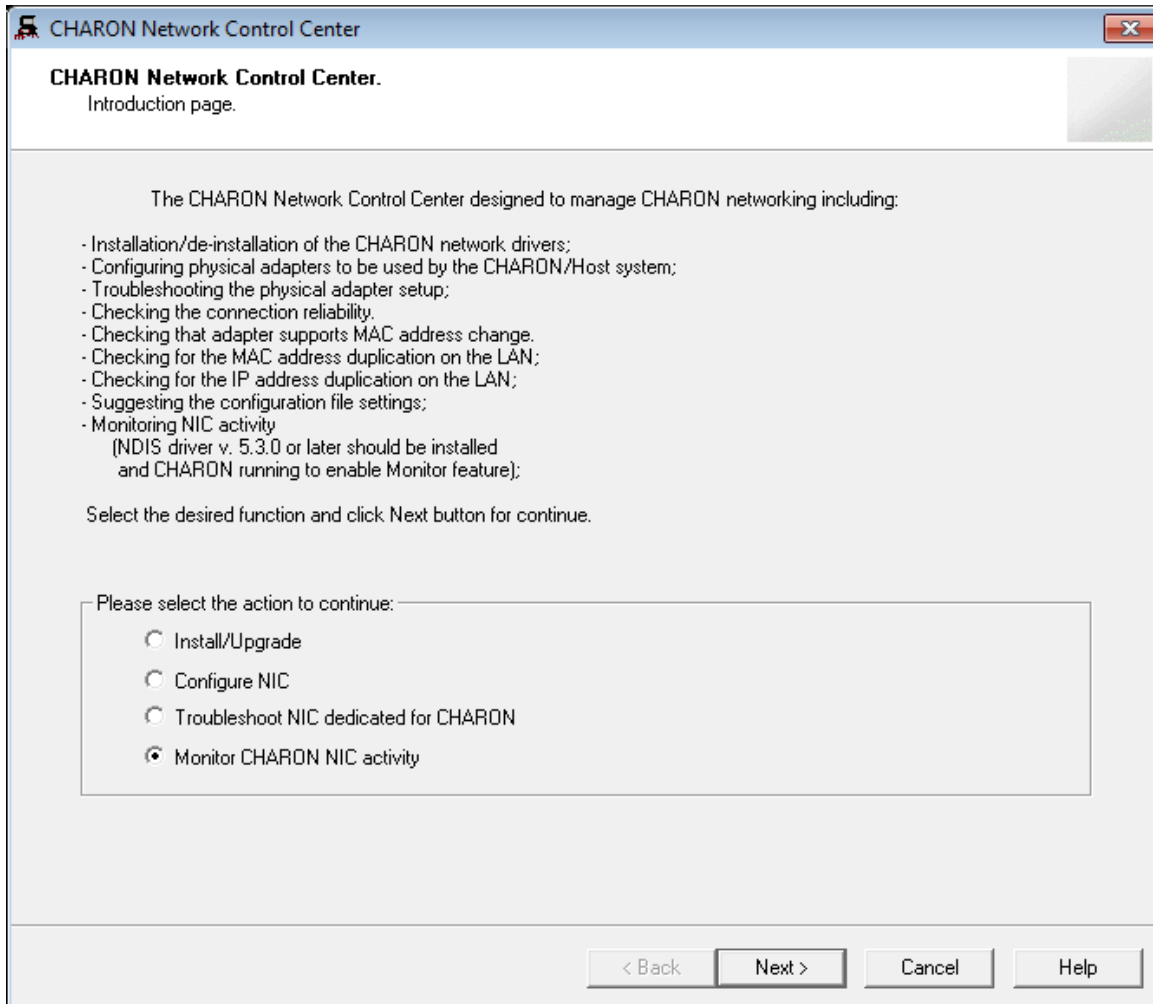
More information on details of disabling TCP chimney offload is available in [this article](#)

[Back to Table of Contents](#)

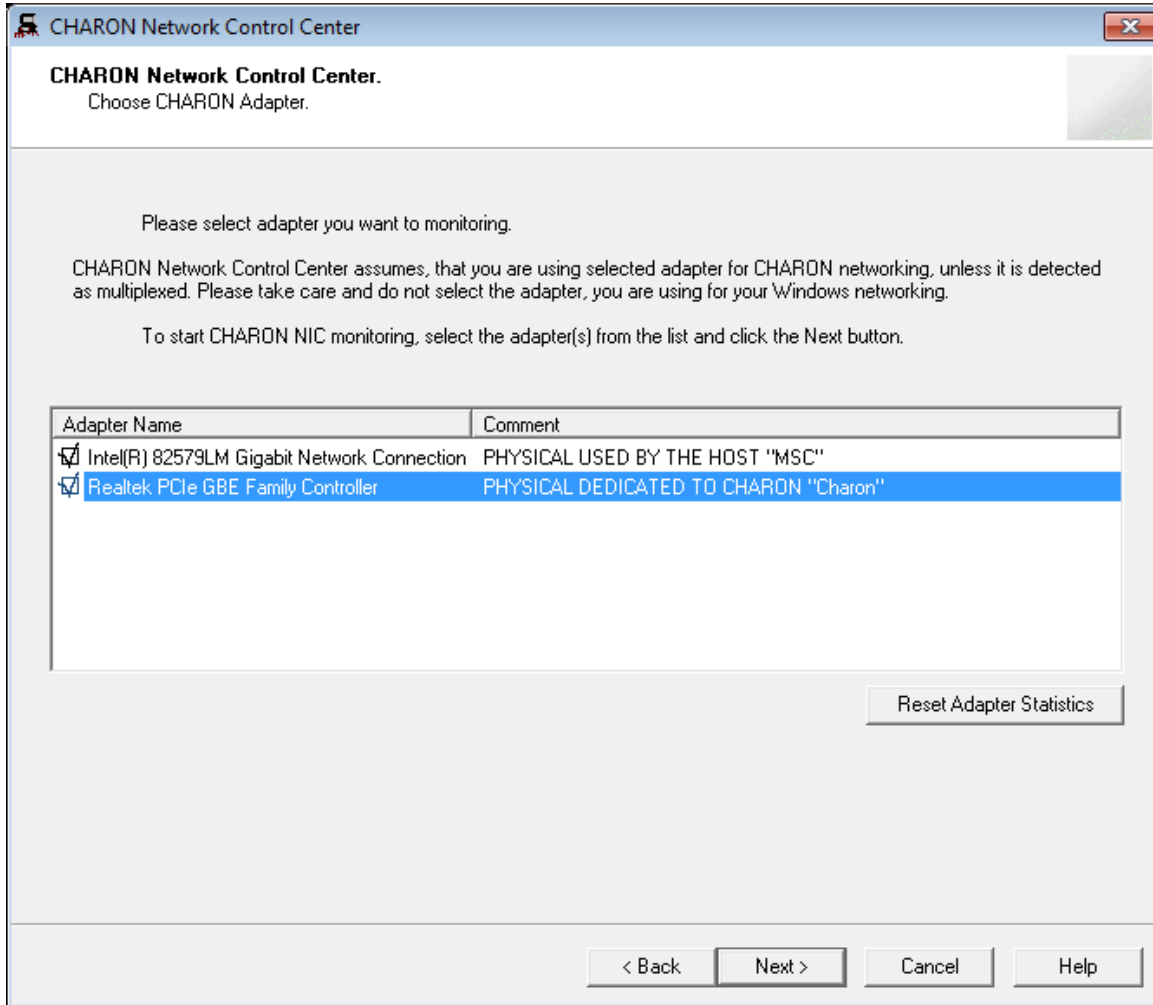
## Monitoring the CHARON network activity

The Network Control Center is able to monitor CHARON network activity. This function is available only if CHARON is running.

Start the utility and select "Monitor CHARON NIC activity"; press "Next":

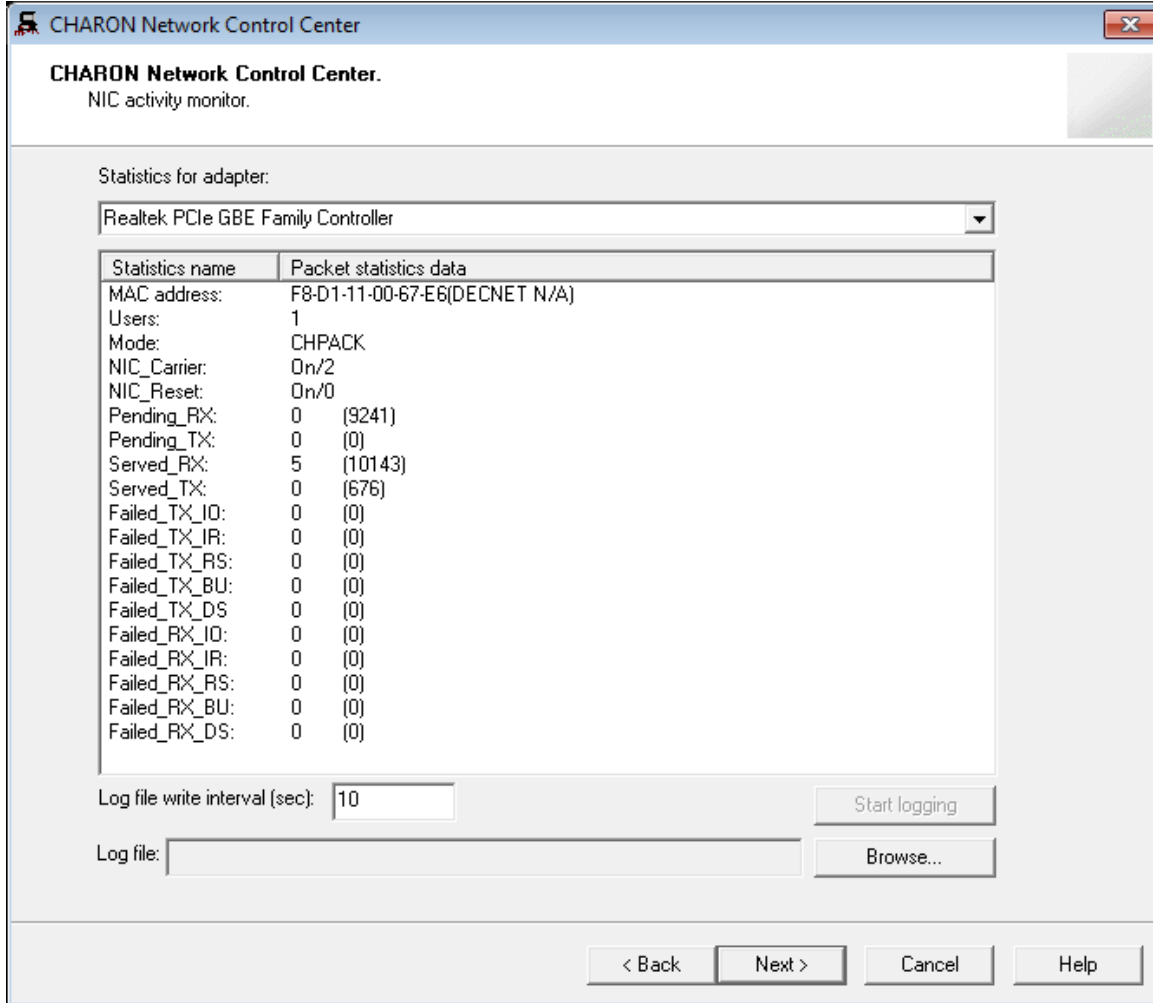


Select the network interface to monitor (it must be dedicated to CHARON); press "Next":

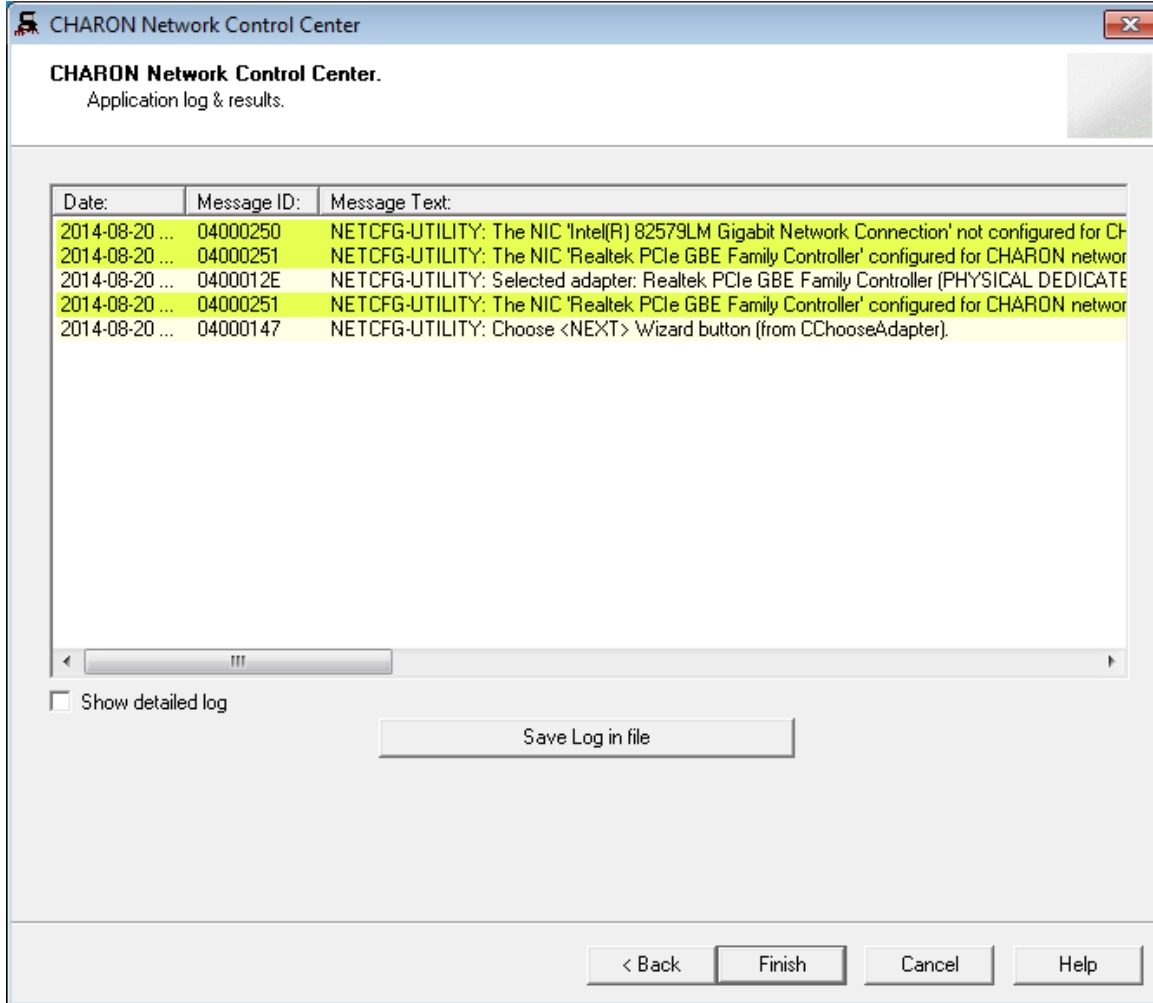




The utility will display statistics updated in a real time. Note, it is possible to record the statistics in a log file with a selected write interval.



Press "Next" button to stop the recording and to see the log of this operation:



Select "Show detailed log" for more detail.

Press "Finish" to exit.

[Back to Table of Contents](#)

# MkDisk

## Table of Contents

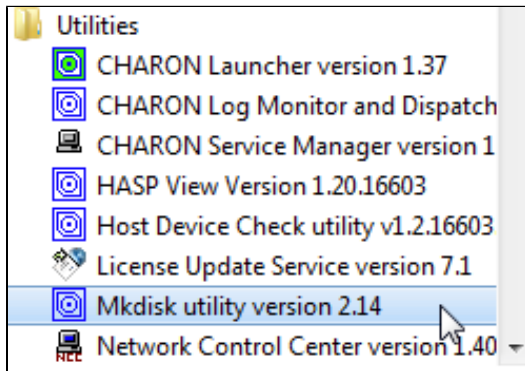
- General Description
- Creating empty disk container
- Creating disk metadata
- Creating custom disk image
- Getting information about available disk types

[Back to Table of Contents](#)

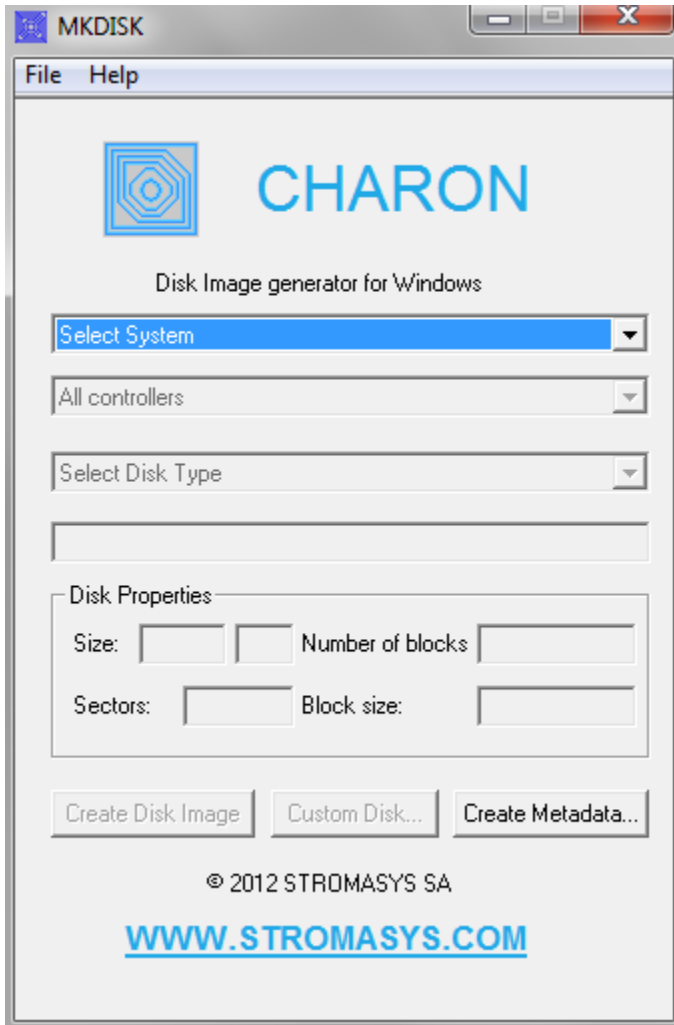
## General Description

The "MkDisk" utility is used to create VAX/PDP11 standard or custom empty disk containers (disk images).

To start the utility, click on the following icon under the "Start" menu:



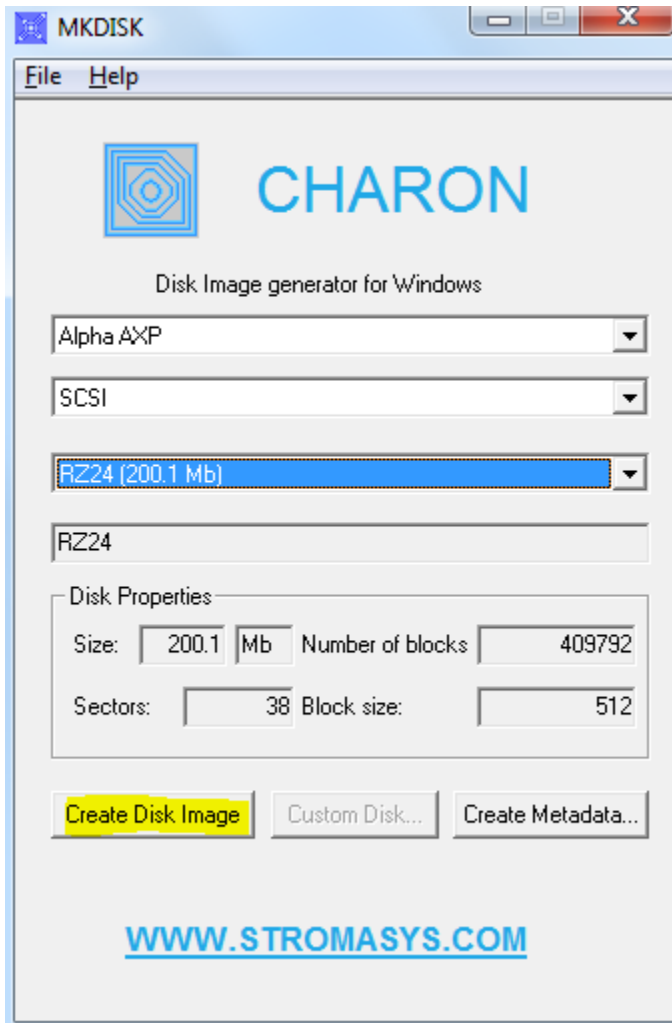
The main dialog of the utility will appear:



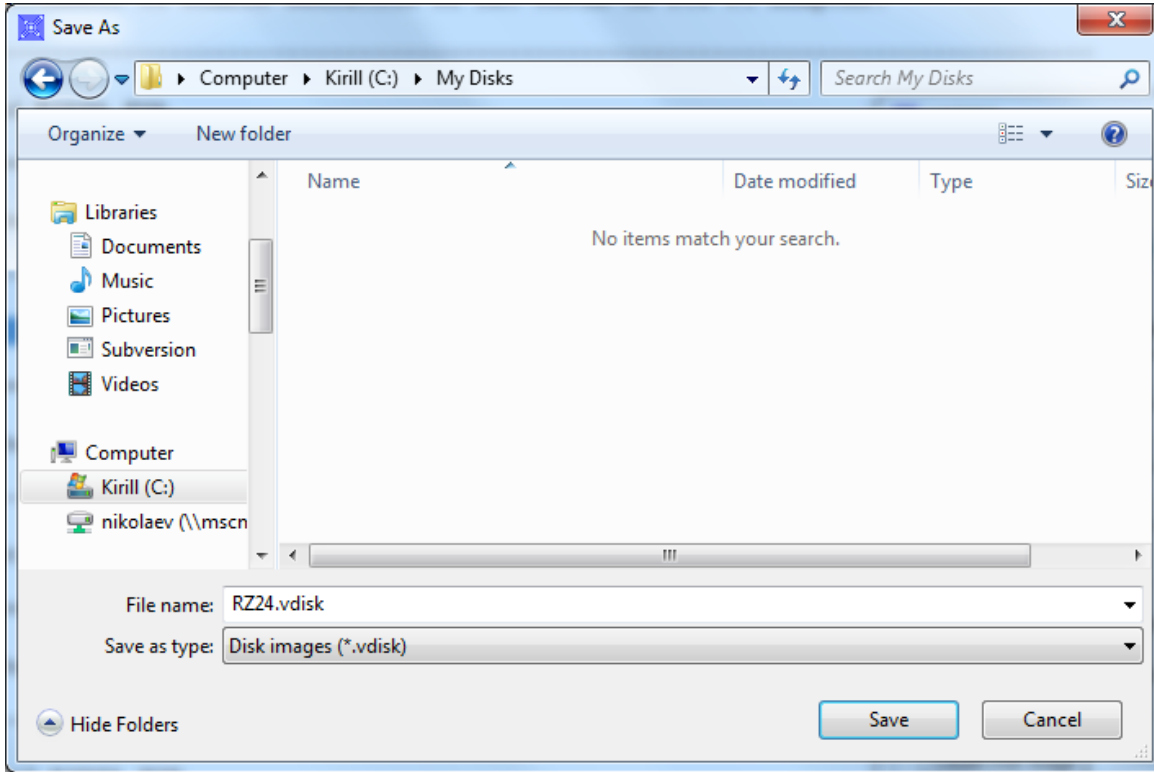
[Back to Table of Contents](#)

## Creating empty disk container

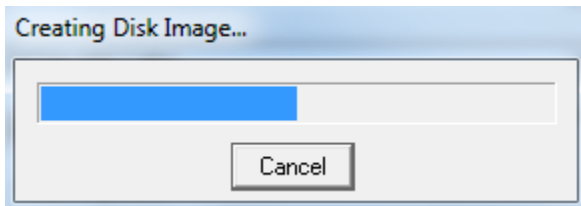
Select "Alpha AXP" in the "Select System" drop-down box, "SCSI" in the "All Controllers" drop-down box and choose the desired disk type in the "Select Disk Type" drop-down box to create an empty disk container:



Press "Create Disk Image" to proceed. A dialog asking to specify the name of the disk image will appear:



Browse to the target directory, specify the name of the disk image and press "Save". The process of creating the disk container will start:



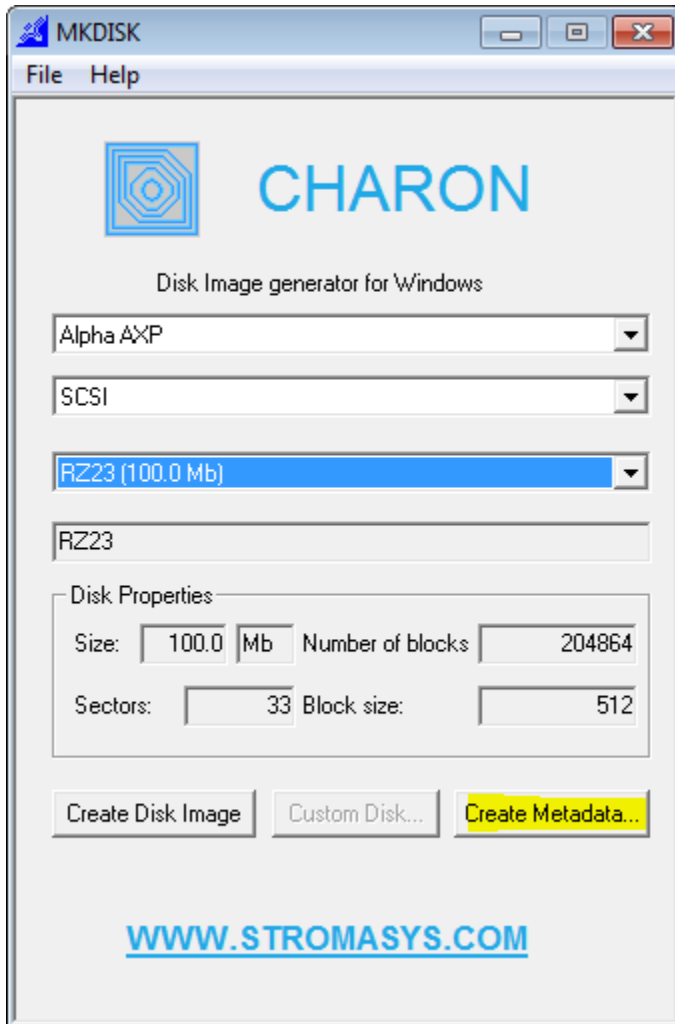
The target disk image is created - along with its metadata file (see below) once the process completes.

[Back to Table of Contents](#)

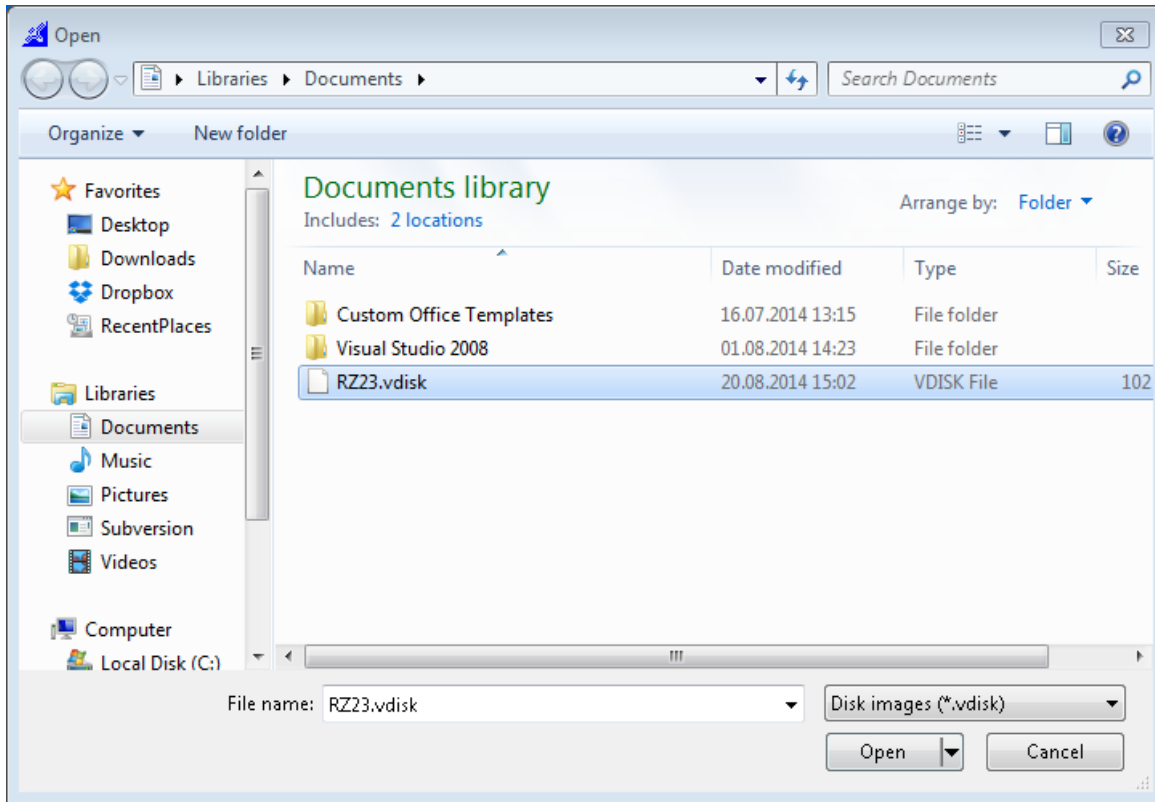
## Creating disk metadata

The disk image metadata file contains important information about the target disk image structure - it is used by CHARON, automatically created with the same name with a different extension.

To create a metadata file press "Create Metadata...":



The "MkDisk" utility will ask you to select a disk container for which to create the metadata file:





Select the target disk image and press "Open". The utility will display a dialog for adjusting/correcting the disk image parameters:

**Metadata configuration dialog**

**Geometry**

Bytes per sector: 512

Sectors per track: 33

Tracks per cylinder: 8

Cylinders per unit: 776

Sectors per unit: 204864

**SCSI**

Vendor: DEC

Product: RZ23

Revision: 0200

UDID:

WWID:

**MSCP**

Media type class: DK

Media type name: RZ23

**SCS**

Node name: RZ23

System ID: 0

**Special**

Controller ID: SCSI Description: DEC RZ23 WINCHESTER

Bad block table:  YES  NO

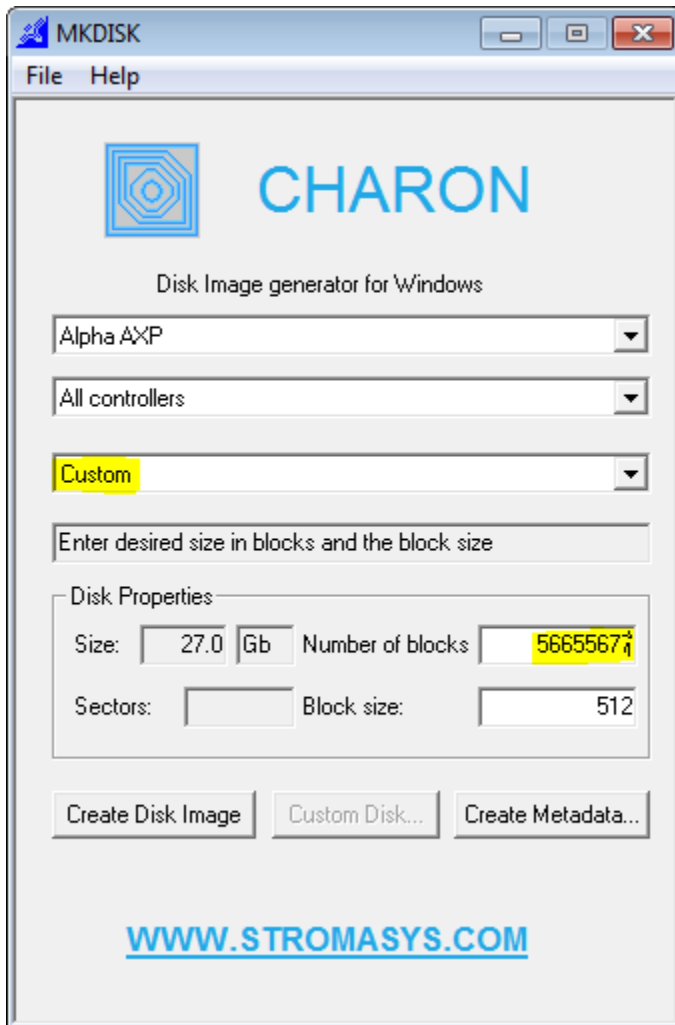
OK Cancel

Correct the desired parameters (if needed) and press "Ok". The "MkDisk" utility will create a metadata file, having the same name as the target disk container, with extension ".avdisk"

[Back to Table of Contents](#)

## Creating custom disk image

Select "Custom" in the "Select System" drop-down box, enter the desired number of blocks in the "Number of blocks" input box, specify the block size (if it is different from the default value of 512) in the "Block size" input box and press "Create Disk Image" to proceed:

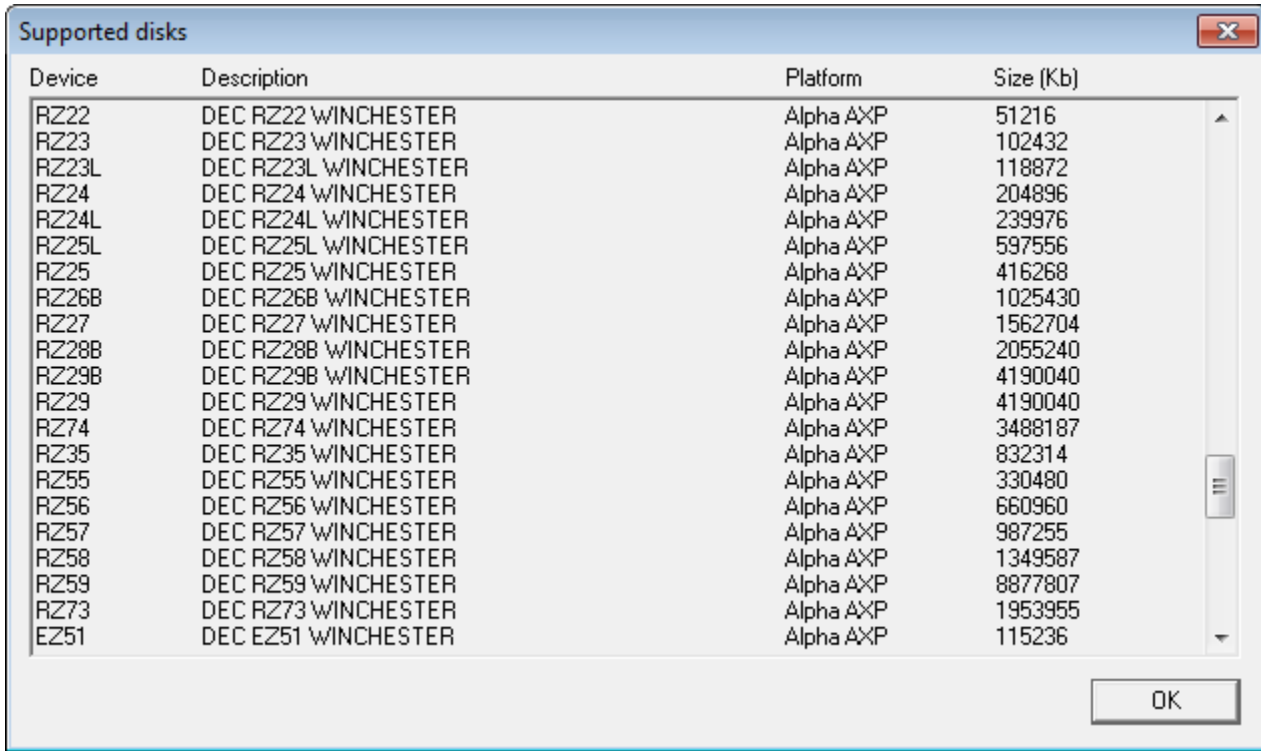


Specify the location and name of the target disk container and press "Save". The utility will create the disk image.

[Back to Table of Contents](#)

## Getting information about available disk types

Information about the available disk types the "MkDisk" utility is able to create can be obtained by selecting the "Help->Disk Table" menu item. The following information dialog will appear:



Device	Description	Platform	Size (Kb)
RZ22	DEC RZ22 WINCHESTER	Alpha AXP	51216
RZ23	DEC RZ23 WINCHESTER	Alpha AXP	102432
RZ23L	DEC RZ23L WINCHESTER	Alpha AXP	118872
RZ24	DEC RZ24 WINCHESTER	Alpha AXP	204896
RZ24L	DEC RZ24L WINCHESTER	Alpha AXP	239976
RZ25L	DEC RZ25L WINCHESTER	Alpha AXP	597556
RZ25	DEC RZ25 WINCHESTER	Alpha AXP	416268
RZ26B	DEC RZ26B WINCHESTER	Alpha AXP	1025430
RZ27	DEC RZ27 WINCHESTER	Alpha AXP	1562704
RZ28B	DEC RZ28B WINCHESTER	Alpha AXP	2055240
RZ29B	DEC RZ29B WINCHESTER	Alpha AXP	4190040
RZ29	DEC RZ29 WINCHESTER	Alpha AXP	4190040
RZ74	DEC RZ74 WINCHESTER	Alpha AXP	3488187
RZ35	DEC RZ35 WINCHESTER	Alpha AXP	832314
RZ55	DEC RZ55 WINCHESTER	Alpha AXP	330480
RZ56	DEC RZ56 WINCHESTER	Alpha AXP	660960
RZ57	DEC RZ57 WINCHESTER	Alpha AXP	987255
RZ58	DEC RZ58 WINCHESTER	Alpha AXP	1349587
RZ59	DEC RZ59 WINCHESTER	Alpha AXP	8877807
RZ73	DEC RZ73 WINCHESTER	Alpha AXP	1953955
EZ51	DEC EZ51 WINCHESTER	Alpha AXP	115236

[Back to Table of Contents](#)

# MkDskCmd

## Table of Contents

- General Description
- Creating disk images
- Transferring disk images

[Back to Table of Contents](#)

## General Description

The "MkDskCmd" utility:

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

To start the utility open "cmd.exe" in the Start menu and switch to the CHARON-AXP x86 utilities directory ("C:\Program Files\CHARON\Utilities\_X.X.XXXXX\x86").

[Back to Table of Contents](#)

## Creating disk images

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

```
...> mkdiskcmd -list
```

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

Choose the desired disk type (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 to 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 the "-blcount" (blocks count) and "-blsize" (blocks size) switches.

For a list of all available parameters use the "--help" switch:

```

...> mkdiskcmd -help

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

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

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

-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

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

Examples:
mkdiskcmd -help
mkdiskcmd -list
mkdiskcmd -avtable \etc\mkdisk.vtable -output "\etc\rk07.vdisk" -disk rk07
mkdiskcmd -output \etc\disks\custom.vdisk -blsize 512 -blcount 16384
mkdiskcmd -t E:\disks\rz22.vdisk rz59 -a "C:\Program Files\CHARON\disks\mkdisk.vtable

```

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

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

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

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

An example of the syntax follows:

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

where:

- <source disk file name> - the file name of the disk image to be transferred
- <source disk parameters> - the disk type taken from the list of available disk types displayed by the "mkdskcmd -list" command or the disk geometry specification (see below).
- -shrink or -k - used to force the shrink when the target disk size is smaller than the source disk size.

Example:

```
...> mkdskcmd -transfer "C:\My Disks\rz22.vdisk" rz25
```

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

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

Example:

```
...> mkdskcmd -t "C:\My Disks\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 up to several minutes in the case of very large disk transfers. This is because the host operating system needs time for the actual allocation of the enlarged file on HDD.

[Back to Table of Contents](#)

# HASP View

## Table of Contents

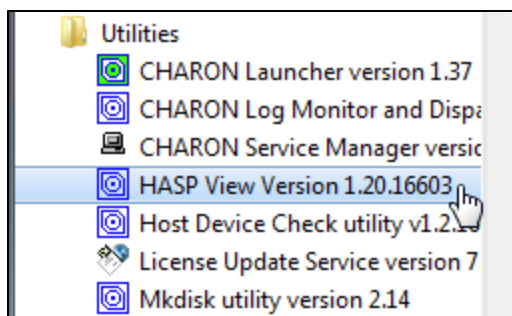
- General Description
- Getting CHARON licenses content

[Back to Table of Contents](#)

## General Description

The "HASP View" utility is used to display the CHARON license content.

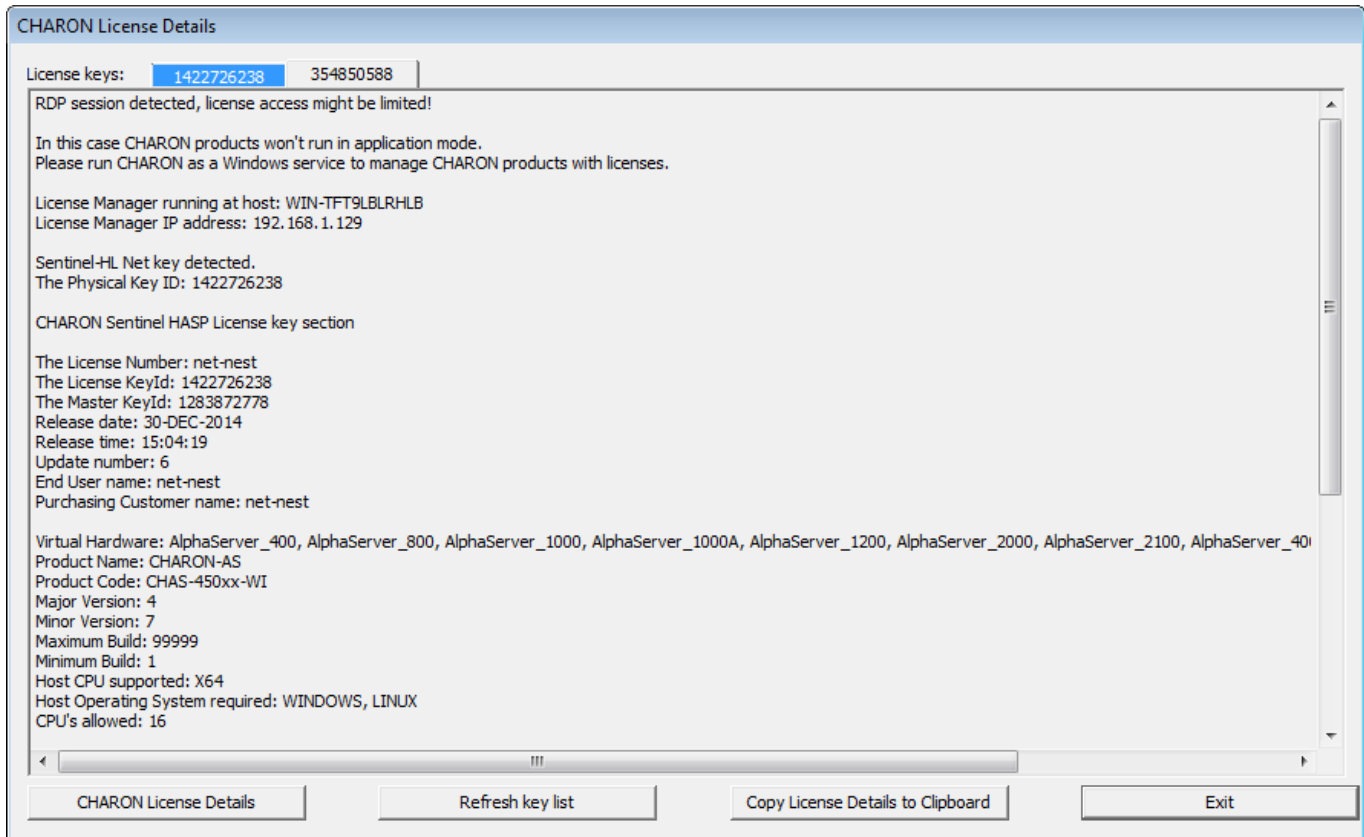
Execute the following program from the "Start" menu to start the utility:



[Back to Table of Contents](#)

## Getting CHARON licenses content

The "HASP View" utility displays the content of the licenses to be used by CHARON:



Use the "License keys" tabs to view the content of the CHARON licenses found.

Press the "Refresh key list" button to look for available licenses.

**Note:** collecting the license content may take some time. Use the buttons located at the bottom of the window to refresh the license content and to copy the license details to the clipboard.

Press "Exit" to exit from the utility.

[Back to Table of Contents](#)



# License Update Service

## Table of Contents

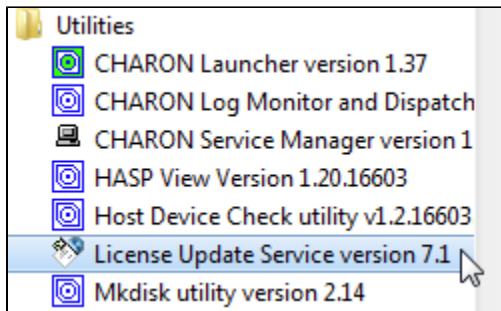
- General Description
- Collecting the host fingerprint and information on an existing license
- Installation and update of a CHARON license
- CHARON software license transfer
- Software License Removal

[Back to Table of Contents](#)

## General Description

The "License Update Service" utility is used to manage CHARON licenses, collect the host system fingerprint and transfer software licenses from one host to another.

Click on the following icon under "Start" menu to start the utility:

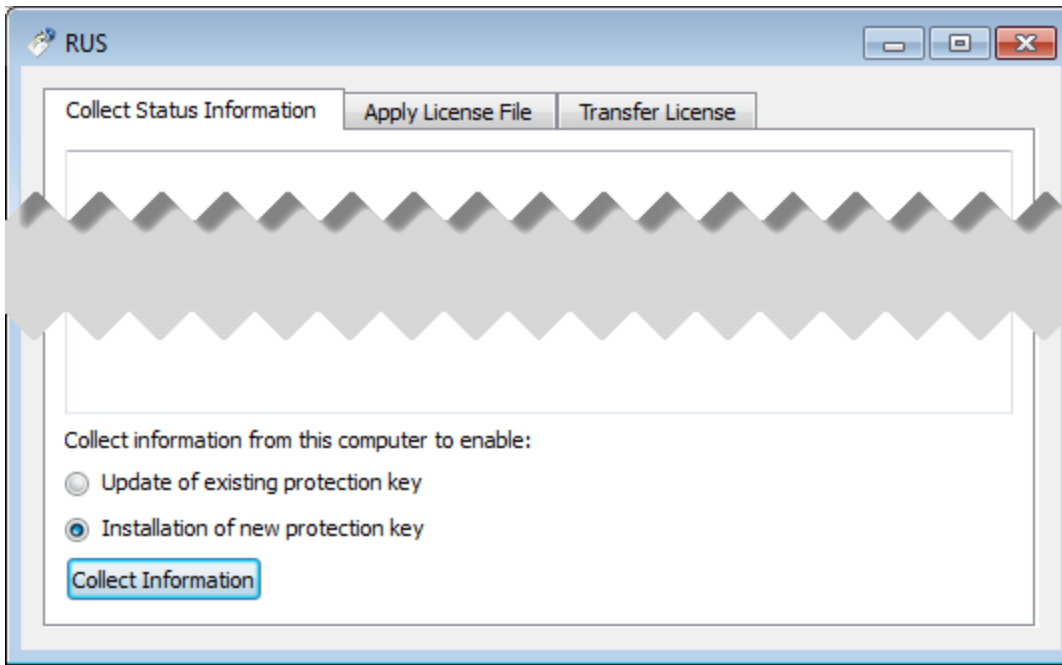


[Back to Table of Contents](#)

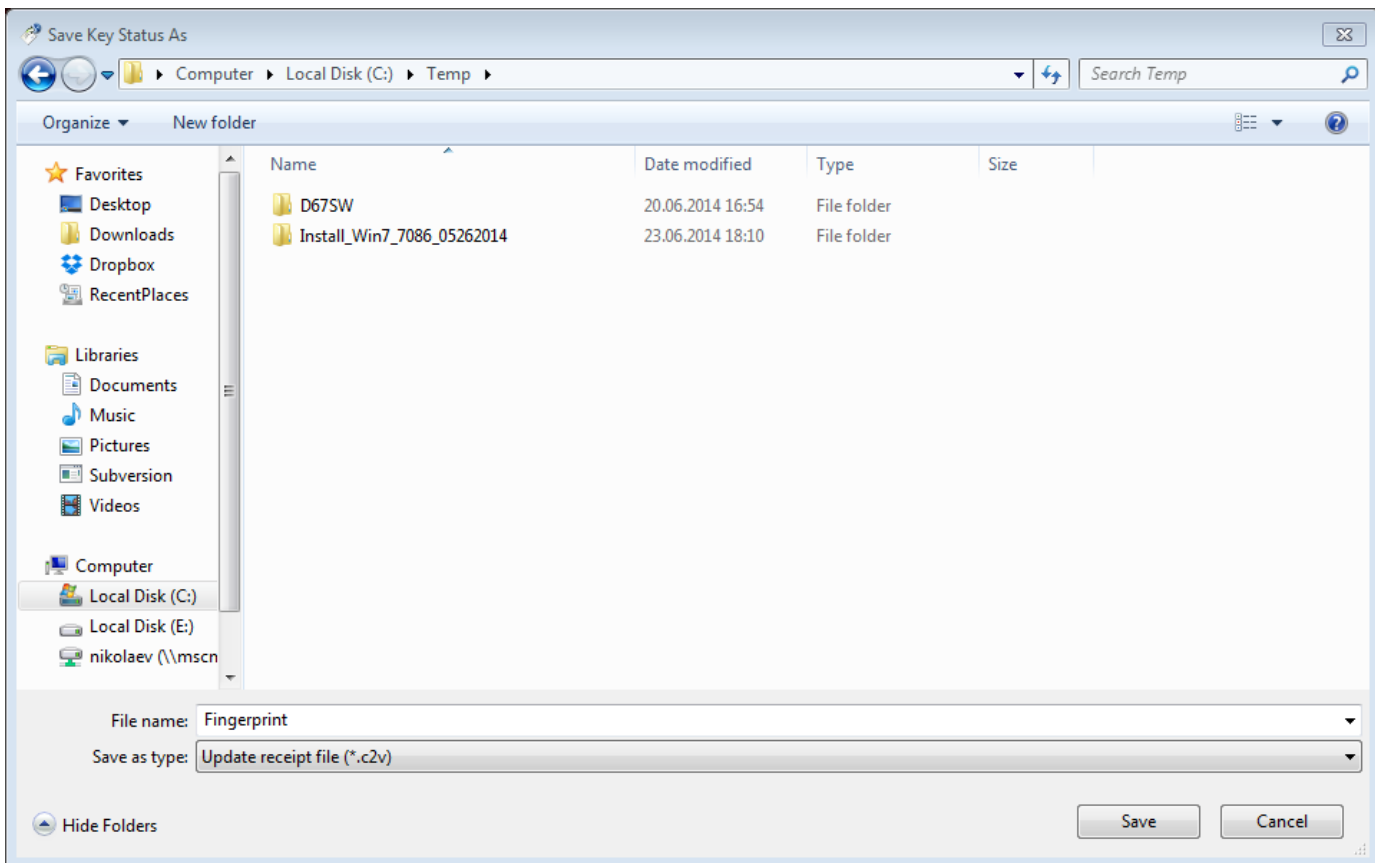
## Collecting the host fingerprint and information on an existing license

Open the "Collect Status Information" tab.

Select either "Update of existing protection key" to acquire information on the current license (both the hardware dongle or the software license) or "Installation of new protection key" to get information on the host system.



Press "Collect Information" and save the fingerprint or the information on the current license to a "\*.c2v" file:



Press "Save" to create the "\*.c2v" file.

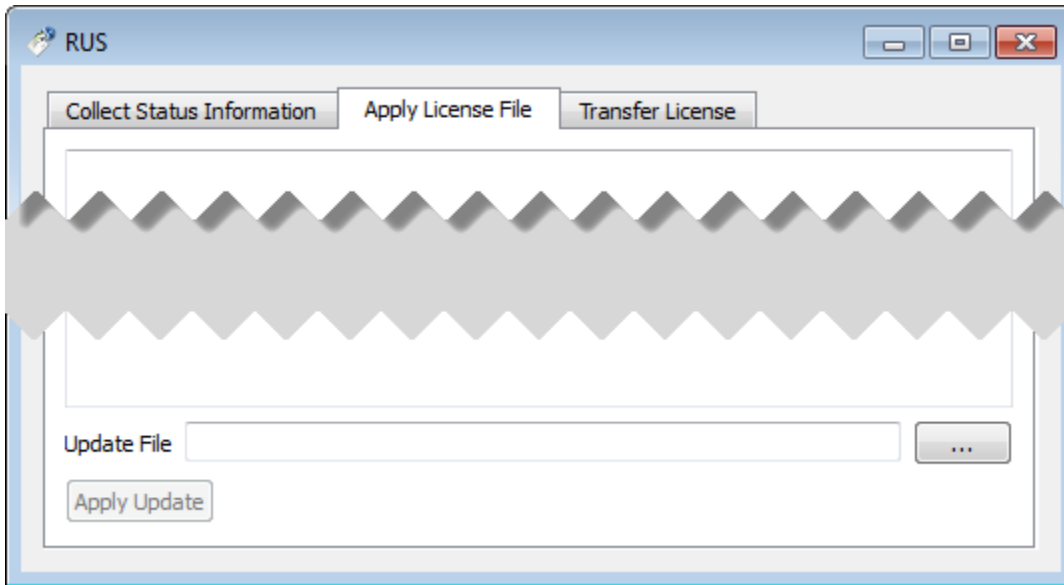
Send this file to STROMASYS to receive an update of the current license or a new license.

[Back to Table of Contents](#)

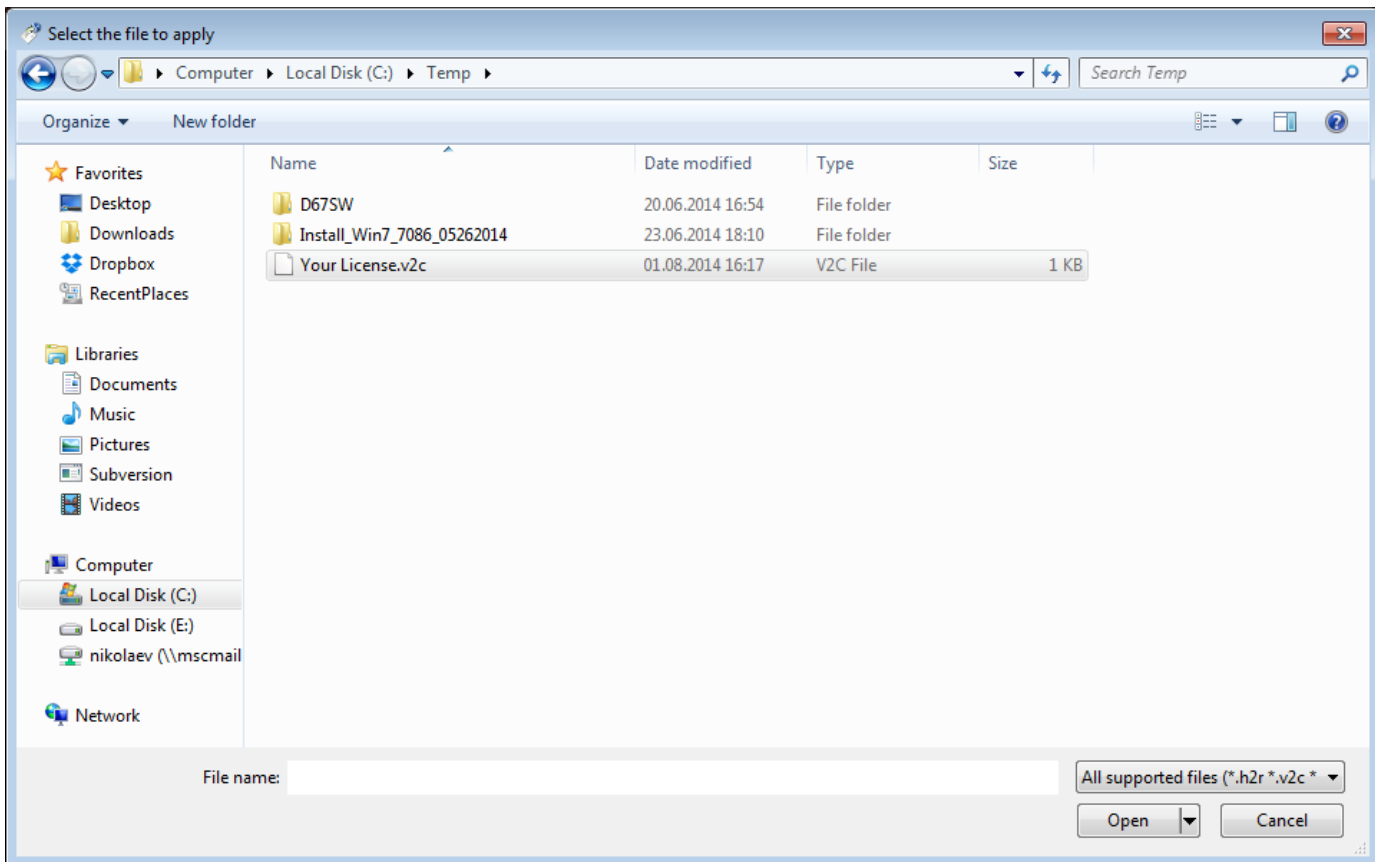
## Installation and update of a CHARON license

Open the "Apply License File" tab.

Press "...":



Select the license "\*.v2c" file received from STROMASYS:

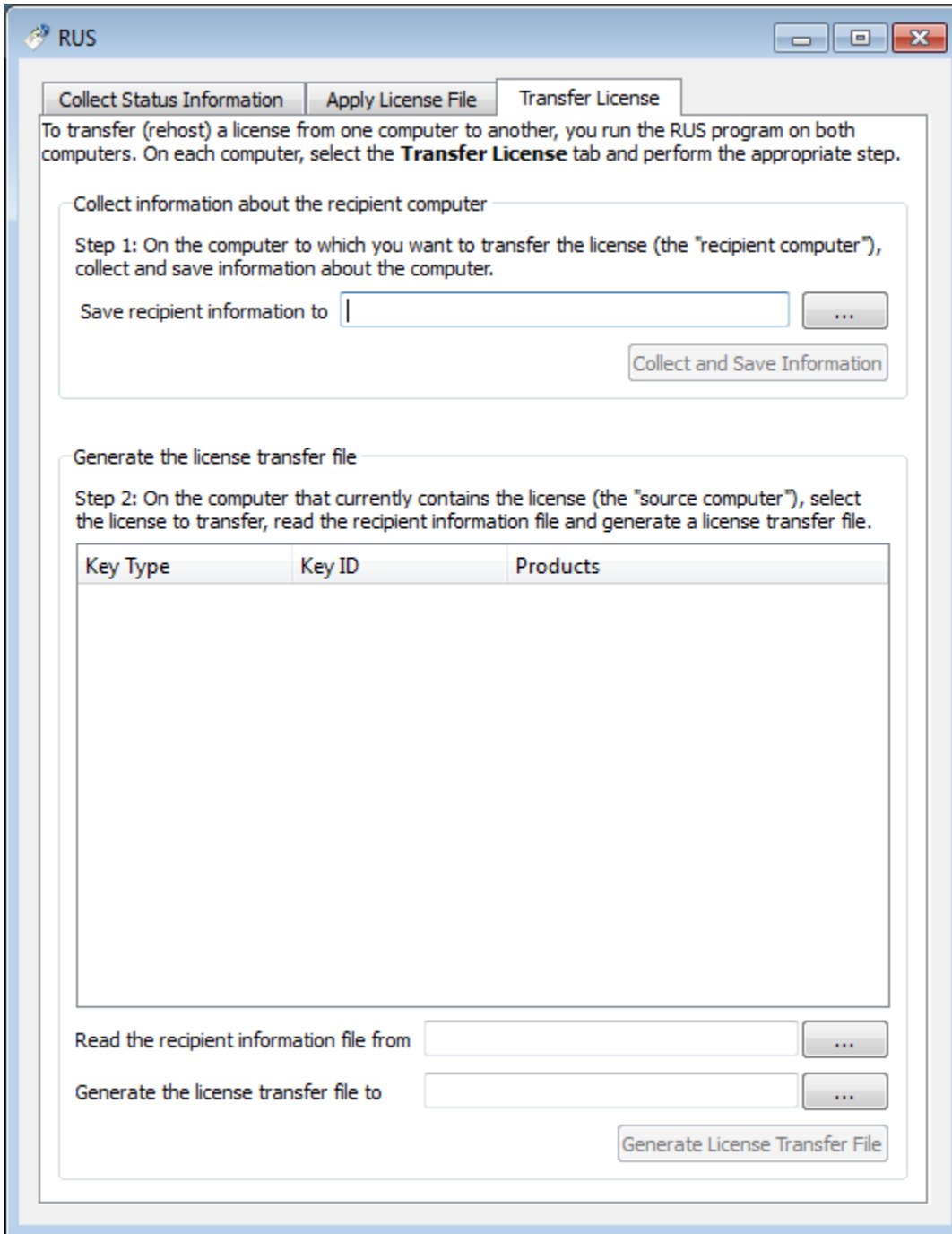


Press "Open" and then "Apply Update" in the main dialog box to apply the new license or to update an existing one.

[Back to Table of Contents](#)

## CHARON software license transfer

Open the "Transfer License" tab:



For transferring licenses from one host to another host (for example: "SourceHost" and "RecipientHost"), execute the following steps:

1. Collect the specific information about the RecipientHost to issue a transfer license. This is done by running the "License Update Service" utility on the "RecipientHost" (see above), choose the "Transfer License" tab and press "." beside the "Save Recipient Information" box.

In the popup choose a directory and file name for the RecipientHost information.  
Press "Collect and Save Information" to create the RecipientHost information file.

2. Copy the RecipientHost file to the SourceHost.

The RecipientHost file is an ASCII file, so use the "ascii" option in FTP transfers.

3. On "SourceHost", run the "License Update Service" utility, select the "Transfer License" tab and the particular license to transfer in the big textbox.

Press the "." beside the "Read the recipient information from file" textbox, choose the just transferred RecipientHost file, press "Open".  
Press the "." beside the "Generate the license transfer file to" textbox, choose the target directory and enter the desired name of the file (\*.h2h), press "Save".

Press "Generate License Transfer File" to create the license transfer file in the specified folder.

4. Copy the resulting "\*.h2h" file to the "RecipientHost".

The "\*.h2h" file is an ASCII file, so use the "ascii" option in FTP transfers.

5. On the "RecipientHost", apply the license transfer file (\*.h2h) the same way as a regular software license (see above).
6. Start any web browser on the "RecipientHost" and go to <http://localhost:1947> to access the "Sentinel HASP Admin Control Center" (ACC).
7. Ensure that the license appears in the "Sentinel Keys" menu.

[Back to Table of Contents](#)

## Software License Removal

After removing a Software License completely from a host, the license is stored in a specific transfer license file "\*.h2h" so it can be re-applied if needed.

Follow the license transfer procedure as described above to remove a Software License completely from a host. It is possible to use the fingerprint of the "SourceHost" (instead the one from the "RecipientHost") for the transfer procedure.

[Back to Table of Contents](#)

# mtd

## General Description

"mtd" is a command line utility to:

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

Open "cmd.exe" in the Start menu, to start the utility, and switch to the CHARON-AXP x64 utilities directory (C:\Program Files\CHARON\Utilities\_X.X.XXXX\x64").

[Back to Table of Contents](#)

## Usage

The following is an example of the usage syntax:

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

Options:

Parameter	Description
-l <file name>	Creates a log file. The name is "file name".
-r <number>	Specifies a number of attempts to read a damaged data block
-i	Ignore bad blocks and continue processing w/o interruption. It implies "-r 0"
-n	Do not rewind tape
-p	Disable progress reporting
-v	Enable verbose tracing of data transfer (implies "-p")

Example:

```
...> mtd -l tapel.txt -r 10 \\.\Tape0 "C:\Charon\Tapes\tapel.vtape"
```

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

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

Example:

```
...> mtd "C:\Charon\Tapes\tapel.vtape" \\.\Tape0
```

[Back to Table of Contents](#)

# CHARON Log Monitor and Dispatcher

## Table of Contents

- General Description
- Starting in background mode
- Installing as a service
- Starting the Log Monitor and Dispatcher service
- Stopping the Log Monitor and Dispatcher service
- Uninstalling the Log Monitor and Dispatcher service

[Back to Table of Contents](#)

## General Description

The Log Monitor & Dispatcher is a special program which monitors a guest LOG file produced by CHARON and executes a customizable script when it detects removal of a license.

It runs in the background (as a program or as a service) and periodically scans a specified LOG file. When it detects a message with the code 0000002A or 00000351 ("Detected removal of a license") it submits the NOLICENSE.BAT procedure (this BAT file must be created manually).

The Log Monitor & Dispatcher service is installed as EmulatorLogMonitor. By default it is installed in such a way that requires explicit actions to be started (either through a command line interface or using the standard ways of service management). For unattended execution, change the service's configuration so that Windows starts the service automatically.

**i** The tool requires a specific file "NOLICENSE.BAT" containing some specific instructions to be taken in situation of license absence. It is recommended you create this file in the folder (presumably) containing the LOG file.

When it is invoked by the "Log Monitor & Dispatcher", the current directory of the batch process is set to the same folder from which the "Log Monitor & Dispatcher" was previously installed as a service or from which it was started as a background process. This means that the user action file may, in principle, operate with relative paths and relocate (as part of the whole configuration, i.e. together with accompanying LOG file).

The user action file will not invoke interactive applications as it may run in an environment where interactive services do not work, for example: when "Log Monitor & Dispatcher" is installed as a service.

## Starting in background mode

In order to **start** the Log Monitor & Dispatcher as a background application:

1. Open "cmd.exe" from the "Start" menu.
2. Change current directory to the folder (presumably) containing the LOG file.
3. Start the Log Monitor & Dispatcher using the following command line as an example:

For single log file:

```
...> "C:\Program Files\CHARON\Utilities_X.X.XXXXX\X86\logmond" my_charon.log
```

For rotating log file:

```
...> "C:\Program Files\CHARON\Utilities_X.X.XXXXX\X86\logmond" -l <log-directory> -p <log-prefix>
```

where:

- <log-directory> is the directory where the rotating log files are stored
- <log-prefix> is the same as the "configuration\_name" value in corresponding CHARON configuration file (or "hw\_model", if "configuration\_name" is not specified).

**Example:**

```
...> "C:\Program Files\CHARON\Utilities_1.0.XXXXX\X86\logmond" -l "C:\My CHARON logs" -p "MY_VAX"
```

To stop the Log Monitor & Dispatcher application, open the Task Manager, find the "logmond.exe" process and terminate it.

[Back to Table of Contents](#)

## Installing as a service

In order to install the Log Monitor & Dispatcher as a background application:

1. Open "cmd.exe" from the "Start" menu in "Run as Administrator" mode.
2. Install Log Monitor & Dispatcher service using the following command line as an example:

```
...> "C:\Program Files\CHARON\Utilities_1.0.XXXXX\X86\logmond" -r  
...> "C:\Program Files\CHARON\Utilities_1.0.XXXXX\X86\logmond" -i
```

As result the "logmond" service has "Manual" service start type. Use Windows service management tools to change this mode to "Automatic" if it is required.

[Back to Table of Contents](#)

## Starting the Log Monitor and Dispatcher service

In order to start the Log Monitor & Dispatcher as a background application:

1. Open "cmd.exe" from the "Start" menu in "Run as Administrator" mode.
2. Start the Log Monitor & Dispatcher service using the following command line as an example:

```
...> "C:\Program Files\CHARON\Utilities_1.0.XXXXX\X86\logmond" -u
```

It is also possible to use Windows service management tools to start up the service.

As soon as "logmond" service is installed this way, and started, it monitors all CHARON instances (provided that these are installed as services).

It does NOT monitor emulator instances started from Launcher or manually from command line prompt, these cases can still be monitored with "logmond" invoked from command line prompt with name of the log file as an argument (see above).

[Back to Table of Contents](#)

## Stopping the Log Monitor and Dispatcher service

In order to stop the Log Monitor & Dispatcher as a background application:

1. Open "cmd.exe" from the "Start" menu in "Run as Administrator" mode.
2. Start the Log Monitor & Dispatcher service using the following command line as an example

```
...> "C:\Program Files\CHARON\Utilities_1.0.XXXXX\X86\logmond" -d
```

It is also possible to use Windows service management tools to stop the service.

[Back to Table of Contents](#)

## Uninstalling the Log Monitor and Dispatcher service

In order to uninstall the Log Monitor & Dispatcher as a background application:

1. Open "cmd.exe" from the "Start" menu in "Run as Administrator" mode.
2. Start the Log Monitor & Dispatcher service using the following command line as an example

```
...> "C:\Program Files\CHARON\Utilities_1.0.XXXXX\X86\logmond" -r
```

[Back to Table of Contents](#)



# IDLE Utility

## Table of Contents

- General Description
- Installation
- Deinstallation

[Back to Table of Contents](#)

## General Description

IDLE utility significantly reduces the CHARON-AXP host CPU usage whenever a OpenVMS system is running on CHARON-AXP is idle.

The utility stalls the emulated CPU when it detects an OpenVMS idle condition. While IDLE utility is running the emulated CPU consumes, on average, less host system CPU time.

 It is not recommended to use the IDLE utility in real-time process control environments.

This set of utilities is located in the "idle\_vms\_pkg.vdisk" disk file in the "C:\Program Files\CHARON\Virtual Disk Images\idle\_vms\_pkg\_v3.0".

[Back to Table of Contents](#)

## Installation

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

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

In the [.AXP] folder of this disk you will see the following files:

```
README.TXT
SRI-AXPVMS-IDLE-V0102-1.PCSI
VMS62TO71U2_PCSI-V0200.PCSI-DCX_AXPEXE
VMS62TO71U2_PCSI-V0200.TXT
```

If the OpenVMS version is less than 7.1 the following steps must be applied:

1. Copy the "VMS62TO71U2\_PCSI-V0200.PCSI-DCX\_AXPEXE" file to some directory on any spare disk and run this file from there:

```
$ COPY VMS62TO71U2_PCSI-V0200.PCSI-DCX_AXPEXE SYS$DISK:[000000.TEMP]
$ SET DEF SYS$DISK:[000000.TEMP]
$ RUN VMS62TO71U2_PCSI-V0200.PCSI-DCX_AXPEXE
```

2. Install the patch:

```
$ PRODUCT INSTALL VMS62TO71U2_PCSI /SOURCE=SYS$DISK:[000000.TEMP]
```

If the patch has been installed return to the "[000000.AXP]" directory of the "idle\_vms\_pkg.vdisk" and proceed with installation of the "Idle" utility itself:

```
$ PRODUCT INSTALL IDLE /SOURCE=<directory containing the IDLE kit>
```

Once IDLE utility is installed it starts to take effect immediately, reducing the host system CPU usage if OpenVMS system running on CHARON-AXP is idle. No reboot is required.

The utility is loaded automatically on reboot, no additional configuration or startup sequence is needed.

[Back to Table of Contents](#)

## Deinstallation

Issue the following command to deinstall IDLE utility:

```
$ PRODUCT REMOVE IDLE
```

The will utility stop working on next system reboot.

Please also refer to the supplied documents "README.TXT" and "VMS62TO71U2\_PCSI-V0200.TXT" for more details.

[Back to Table of Contents](#)

# HOSTPrint

## Table of Contents

- General Description
- Usage

[Back to Table of Contents](#)

## General Description

The HOSTPrint utility is used to print CHARON output to Windows printers. It receives data from COM2 port of emulated HP Alpha model via a TCP/IP socket and prints the data received on the default Windows printer (if no printer is specified at the utility command line) on the host computer.

Note that the second console line "TTA0" (COM2) is available only for 1 CPU models such as:

- HP AlphaServer 400
- HP AlphaServer 800
- HP AlphaServer 1000
- HP AlphaServer 1000A
- HP AlphaServer DS10
- HP AlphaServer DS10L
- HP AlphaServer DS15

[Back to Table of Contents](#)

## Usage

HOSTPrint calling is specified in the CHARON configuration file. The call implement one or more of the following parameters:

Parameter	Description
-host=<hostname>	Name of the host - source of the printing data.
-port=<connection port number>	Port on the host to get the information to print from.
-delay=<delay for automatically buffer flush in seconds>	Flushing delay, 0 - wait infinite, 5..10800 - timeout for flush
-printer=[PrinterDeviceName]	Host name for the printer used.  <u>Example 1:</u> <pre>-printer=[\\.\Microsoft Office Document Image Writer]</pre> <u>Example 2:</u> <pre>-printer=[\\print_server\MSCLPS]</pre>
-font=<default font face>	Default font name.
-fontsize=<default font size>	Default font size.

The two last parameters are only used for compatibility with older versions of the utility (HOSTprint allows changing font settings from a popup menu).


It is strongly recommended to use fixed-size fonts (by default the "Courier" font is used) to avoid any problems relevant to proper calculation of the printing line length.

Example of CHARON-AXP configuration file for the HOSTPrint usage:

```
load virtual_serial_line TTA0 port=10000 application="hostprint.exe -port=10000 -printer=[\\print_server\MSCPS2]
-font=\Courier New\ -fontsize=10"
set COM2 line=TTA0
```

After initialization, HOSTprint creates an icon in the Windows tray. The icon can have two colors:

Color	Description
<b>GREEN</b>	IDLE (or Ready) state
<b>YELLOW</b>	BUSY (processing) mode

\*:  To access the HOSTprint application popup menu, point the mouse cursor at its icon in the system tray menu and click the right button, preview the last page content then change the utility mode via the flush buffer delay or change the default font setting.

[Back to Table of Contents](#)

# CHARON-AXP for Windows configuration details

## Introduction

This chapter describes, in detail, all of the configuration parameters of the devices emulated by CHARON-AXP for Windows, 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]="C:\Charon\Disks\BootDisk.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 "C:\Charon\Disks\BootDisk.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
- Networking
- DEFPA PCI FDDI adapter
- PBXDA-xx series PCI serial adapters
- 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

### 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><u>Example:</u></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-xxxxxxxxx.log</pre> <p>xxxxxxxxx is an increasing decimal number starting from 000000000 to separate log files with the same time of creation (in case the log is being written faster than one log file per second).</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. If an existing directory is specified, CHARON-AXP automatically creates individual log files for each CHARON-AXP execution session. If the log parameter is omitted CHARON-AXP creates a log file for each CHARON-AXP execution session in the directory where the emulator was started.</p> <p>It is possible to overwrite the existing log file or extend it using the "log_method" parameter. See below for details.  The "log_method" parameter is effective only in the case where a single log file is specified, not a directory.</p> <p><b>Examples:</b></p> <pre>set session log="log.txt"</pre> <pre>set session log="C:\Charon\Logs"</pre> <p>If only a directory name is specified in the "log" parameter and the "configuration_name" parameter of the session is specified, the log file name will have the following format:</p> <pre>&lt;configuration_name&gt;-YYYY-MM-DD-hh-mm-ss-xxxxxxxxx.log</pre> <p>If only a directory name is specified in the "log" parameter and the "configuration_name" parameter is omitted, the log file name will have the following format:</p> <pre>&lt;hw_model&gt;-YYYY-MM-DD-hh-mm-ss-xxxxxxxxx.log</pre> <p>xxxxxxxxx is an increasing decimal integer, starting from 000000000 to separate log files with the same time of creation (in case the log is being created faster than one log file per second).</p> <div style="border: 1px solid orange; padding: 5px; margin-top: 10px;"> <p>Only existing directory can be used as a value of the "log" parameter.</p> </div>

## log\_method

<b>Parameter</b>	log_method
<b>Type</b>	Text string
<b>Value</b>	<ul style="list-style-type: none"> <li>• "overwrite" (default)</li> <li>• "append"</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><u>Example:</u></p> <div style="border: 1px solid black; padding: 5px; margin: 10px 0;"> <pre>set session log="log.txt" log_method="append"</pre> </div>

## log\_show\_messages

<b>Parameter</b>	log_show_messages
<b>Type</b>	Text string
<b>Value</b>	<ul style="list-style-type: none"> <li>• "all" (default)</li> <li>• "info"</li> <li>• "warning"</li> <li>• "error"</li> </ul> <p>Defines the message types to be shown. The parameter is a string of comma delimited words.</p> <p><u>Example:</u></p> <div style="border: 1px solid black; padding: 5px; margin: 10px 0;"> <pre>set session log_show_messages="error, warning"</pre> </div>

## log\_repeat\_filter

<b>Parameter</b>	log_repeat_filter
<b>Type</b>	Text string
<b>Value</b>	<p>Specifies if repeated messages should be filtered or not. Possible values are "on" (default) and "off". If the value is "on", immediately following messages with the same identifier and system error code are not listed in the log, but they are counted. When a different log message is generated, the repeat count of the earlier log message is reported with "The previous message has been repeated N times.", and the counter is cleared.</p> <p><u>Example:</u></p> <div style="border: 1px solid black; padding: 5px; margin: 10px 0;"> <pre>set session log_repeat_filter="on"</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>

## 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. Used for soft partitioning of the host CPU resources and/or for isolating host CPUs for other applications. By default the CHARON-AXP emulator instance allocates as many host CPUs as possible. The "affinity" parameter overrides that and allows explicit specification on which host CPU the instance must run on.</p> <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> <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><u>Example:</u></p> <pre style="border: 1px solid #ccc; padding: 5px; width: fit-content;">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="background-color: #d9e1f2;">HP Alpha Model</th> <th style="background-color: #d9e1f2;">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. 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><u>Example:</u></p> <pre style="border: 1px solid #ccc; padding: 5px;">set session n_of_io_cpus=2</pre>

## 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 style="border: 1px solid #ccc; padding: 5px;">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 style="border: 1px solid #ccc; padding: 5px;">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 style="border: 1px solid #ccc; padding: 5px;">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 style="border: 1px solid #ccc; padding: 5px;">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 style="border: 1px solid #ccc; padding: 5px;">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>

[Back to Table of Contents](#)

# Core Devices

## Table of Contents

- CPU
  - enabled
  - cpu\_architecture
  - cache\_size
- RAM
  - size
  - Example
- TOY
  - container
  - sync\_to\_host
  - Example
- ROM
  - container
  - system\_name
  - system\_serial\_number
  - dsrdb
  - version
- Virtual HP Alpha interval timer
  - clock\_period
  - Example for AlphaServer 400, DS, ES, GS
  - Example for AlphaServer 800, 1000, 1000A, 1200, 2000, 2100, 4000, 4100
- 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> <p>The ACE mode is disabled when the host system does not meet the minimum physical requirements for this operation. If the emulator appears not to run at its normal performance, check the log file for a change in the ACE mode and verify that sufficient host resources, especially memory, are available.</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>	Numeric
<b>Value</b>	<p>"cache_size" defines the amount of memory in megabytes allocated to the ACE cache.</p> <p>The default value of the cache_size set to 1GB (1024 MB).</p> <p><u>Example:</u></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> 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>Disable undesirable 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	Disable undesirable 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	Disable undesirable 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>														

## Example

```
set toy container="C:\Charon\my_virtual_system.dat"
```

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><u>Example:</u></p> <pre>set rom container="C:\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. dsrdb[0] stands for SMM - System Marketing Model.</p> <p><u>Example:</u></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><u>Example:</u></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).

## clock\_period

<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

## 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><u>Example:</u></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
    - Example of mapping a console line to a host physical serial line
  - virtual\_serial\_line
    - host
    - port
    - application
    - break\_on
    - stop\_on
    - log
    - Example 1
    - Notes on "virtual\_serial\_line" options
    - Example 2

[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 has an additional on-board serial line controller providing a serial line TTA0.

The regular console is already preloaded, so you need to specify just its mapping to the host resources, for example:

```
load physical_serial_line OPA0 line="COM1:"
```

In case of 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	<p>This type of mapping associates a COM port on a host system with an emulated HP Alpha serial line controller virtual "line".</p> <p>The COM port can be a physical hardware port or a logical COM port.</p>
virtual_serial_line	<p>This type of mapping associates a network connection on the host system with an emulated HP Alpha serial line controller virtual "line"</p>

**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 COM port on a host system in the form of "\\.\COMn" (or "COM&lt;n&gt;:")</p> <p><b>Example:</b></p> <pre>set OPA0 line="\\.\COM1"</pre>

### baud


<b>Parameter</b>	baud
<b>Type</b>	Numeric
<b>Value</b>	<p>Forces the baud rate of the corresponding COM port to be a specified value.</p> <p>The variety of supported values depends on the underlying physical communication resource (COM 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> <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 physical serial line will trigger a STOP condition. The STOP condition causes CHARON-AXP to exit.</p> <p>Specify the one of the following values: "F6" or "none" ("none" disables triggering a STOP condition).</p> <p><u>Example:</u></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. 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><u>Examples:</u></p> <pre>set OPA0 log="log.txt"</pre> <pre>set OPA0 log="C:\Charon\Logs"</pre> <p style="text-align: center;">Only existing directory can be used as a value of the "log" parameter.</p>

## Example of mapping a console line to a host physical serial line

```
load physical_serial_line OPA0
set OPA0 line="\\.COM1"
```

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


## application

<b>Parameter</b>	application
<b>Type</b>	Text string
<b>Value</b>	<p>An application (a terminal emulator is assumed in most cases) to be started on initialization of this serial line emulation. The specified application startup string may contain all required parameters.</p> <p><b>Example:</b></p> <pre>set OPA application = "putty.exe -load OPA0"</pre> <p>In this example the terminal emulator application: "putty" is started with the parameters "-load OPA0" telling it to load a specific saved session named "OPA0", (created separately) from the host registry.</p> <p>The "application" parameter is often combined with a "port" parameter:</p> <pre>set TTA0 port = 10003 application = "putty.exe -load TTA1"</pre>

## 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> <p><u>Example:</u></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 one of the following values: "F6" or "none" ("none" disables triggering a STOP condition). The commands are case insensitive.</p> <p><u>Example:</u></p> <pre>set OPA0 stop_on="F6"</pre> <p>The default value is "none".</p> <p>Setting "<b>F6</b>" 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><u>Examples:</u></p> <pre>set OPA0 log="log.txt "</pre> <pre>set OPA0 log="C:\Charon\Logs"</pre> <p>Only existing directory can be used as a value of the "log" parameter.</p>

## Example 1

Mapping a console line to a host TCP/IP port 10003 and starting a "Putty" terminal emulator with a configuration "OPA" directing it to connect to the port 10003:

```
load virtual_serial_line OPA0
set OPA port = 10003 application = "putty.exe -load OPA0"
```

## 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 2

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 HP Alphas. The sequential order in which the instances of CHARON-AXP are started makes no difference.

[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><u>Example:</u></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><u>Example:</u></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><u>Example:</u></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
```

 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 EWA bus=pci_0 device=6 function=0
```

**Example 2:** Loading multiple DE500BA's into slot 0, populating all 4 functions (gives 4 Ethernet ports)

```
load DE500BA/dec21x4x EWA bus=pci_0 device=6 function=0
load DE500BA/dec21x4x EWB bus=pci_0 device=6 function=1
load DE500BA/dec21x4x EWC bus=pci_0 device=6 function=2
load DE500BA/dec21x4x EWD bus=pci_0 device=6 function=3
```

**Example 3:** Loading mixture of KZPBA and DE500BA into slot 1, populating 2 functions out of 4

```
load KZPBA PKB bus=pci_0 device=7 function=0
load DE500BA/dec21x4x EWA bus=pci_0 device=7 function=1
```

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

[Back to Table of Contents](#)

## AlphaServer 2100 (3 PCI slots)

In addition to 3 PCI vacant slots there are 3 PCI positions occupied by on-board devices. All 6 PCI positions are listed in the following table in the order in which Alpha SRM console enumerates them.

Slot	pci_<N>	device	function	irq	Description	Preloaded Name
<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 EWA bus=pci_0 device=6 function=0
```

### Example 2: Loading multiple DE500BA's into slot 0, populating all 4 functions (gives 4 Ethernet ports)

```
load DE500BA/dec21x4x EWA bus=pci_0 device=6 function=0
load DE500BA/dec21x4x EWB bus=pci_0 device=6 function=1
load DE500BA/dec21x4x EWC bus=pci_0 device=6 function=2
load DE500BA/dec21x4x EWD bus=pci_0 device=6 function=3
```

### Example 3: Loading mixture of KZPBA and DE500BA into slot 1, populating 2 functions out of 4

```
load KZPBA PKB bus=pci_0 device=7 function=0
load DE500BA/dec21x4x EWA bus=pci_0 device=7 function=1
```

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

[Back to Table of Contents](#)

## AlphaServer 4000 (16 PCI slots)

In addition to 16 PCI vacant slots there are 2 PCI positions occupied by on-board devices. All 18 PCI positions are listed in the following table in the order in which Alpha SRM console enumerates them.

Slot	pci_<N>	device	function	irq	Description	Preloaded Name
<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
```

**i** In the above example device name is PKB as there is a built-in PK-like PCI SCSI Adapter located “closer” to CPU and therefore assigned name PKA.

[Back to Table of Contents](#)

## AlphaServer DS10 (4 PCI slots)

In addition to 4 PCI vacant slots there are 5 PCI positions occupied by on-board devices. All 9 PCI positions are listed in the following table in the order in which Alpha SRM console enumerates them.

Slot	pci_<N>	device	function	irq	Description	Preloaded Name
<i>PCI1 (bus=pci_1)</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 EWA bus=pci_0 device=14 function=0
```




**Example 2:** Loading multiple DE500BA's into slot 1, populating all 4 functions (gives 4 Ethernet ports)

```
load DE500BA/dec21x4x EWA bus=pci_0 device=14 function=0
load DE500BA/dec21x4x EWB bus=pci_0 device=14 function=1
load DE500BA/dec21x4x EWC bus=pci_0 device=14 function=2
load DE500BA/dec21x4x EWD bus=pci_0 device=14 function=3
```

**Example 3:** Loading mixture of KZPBA and DE500BA into slot 1, populating 2 functions out of 4

```
load KZPBA PKB bus=pci_0 device=14 function=0
load DE500BA/dec21x4x EWC bus=pci_0 device=14 function=1
```

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

[Back to Table of Content](#)

## AlphaServer DS10L (1 PCI slot)

In addition to 1 PCI vacant slots there are 5 PCI positions occupied by on-board devices. All 6 PCI positions are listed in the following table in the order in which Alpha SRM console enumerates them.

Slot	pci_<N>	device	function	irq	Description	Preloaded Name
<i>PCI1 (bus=pci_1)</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 EWA bus=pci_0 device=17 function=0
```

**Example 2:** Loading multiple DE500BA's into slot 1, populating all 4 functions (gives 4 Ethernet ports)

```
load DE500BA/dec21x4x EWA bus=pci_0 device=17 function=0
load DE500BA/dec21x4x EWB bus=pci_0 device=17 function=1
load DE500BA/dec21x4x EWC bus=pci_0 device=17 function=2
load DE500BA/dec21x4x EWD bus=pci_0 device=17 function=3
```

**Example 3:** Loading mixture of KZPBA and DE500BA into slot 1, populating 2 functions out of 4

```
load KZPBA PKB bus=pci_0 device=17 function=0
load DE500BA/dec21x4x EWC bus=pci_0 device=17 function=1
```

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

[Back to Table of Content](#)

## AlphaServer DS15 (4 PCI slots)

In addition to 4 PCI vacant slots there are 7 PCI positions occupied by on-board devices. All 11 PCI positions are listed in the following table in the order in which Alpha SRM console enumerates them.

Slot	pci_<N>	device	function	irq	Description	Preloaded Name
<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 EWA bus=pci_2 device=7 function=0
```

**Example 2:** Loading multiple DE500BA's into slot 2, populating all 4 functions (gives 4 Ethernet ports)

```
load DE500BA/dec21x4x EWA bus=pci_2 device=8 function=0
load DE500BA/dec21x4x EWB bus=pci_2 device=8 function=1
load DE500BA/dec21x4x EWC bus=pci_2 device=8 function=2
load DE500BA/dec21x4x EWD bus=pci_2 device=8 function=3
```

**Example 3:** Loading mixture of KZPBA and DE500BA into slot 3, populating 2 functions out of 4

```
load KZPBA PKC bus=pci_2 device=9 function=0
load DE500BA/dec21x4x EWA bus=pci_2 device=9 function=1
```

 In the above example device name is PKC as there are two built-in PK-like PCI SCSI Adapters located “closer” to CPU and therefore assigned names PKA and PKB.

[Back to Table of Content](#)

## AlphaServer DS20 (6 PCI slots)

In addition to 6 PCI vacant slots there are 5 PCI positions occupied by on-board devices. All 11 PCI positions are listed in the following table in the order in which Alpha SRM console enumerates them.

Slot	pci_<N>	device	function	irq	Description	Preloaded Name
<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 EWA bus=pci_3 device=1 function=0
```

**Example 2:** Loading multiple DE500BA's into slot 5, populating all 4 functions (gives 4 Ethernet ports)

```
load DE500BA/dec21x4x EWA bus=pci_3 device=1 function=0
load DE500BA/dec21x4x EWB bus=pci_3 device=1 function=1
load DE500BA/dec21x4x EWC bus=pci_3 device=1 function=2
load DE500BA/dec21x4x EWD bus=pci_3 device=1 function=3
```

**Example 3:** Loading mixture of KZPBA and DE500BA into slot 6, populating 2 functions out of 4

```
load KZPBA PKC bus=pci_3 device=2 function=0
load DE500BA/dec21x4x EWA bus=pci_3 device=2 function=1
```

 In the above example device name is PKC as there are two built-in PK-like PCI SCSI Adapters located "closer" to CPU and therefore assigned names PKA and PKB.

[Back to Table of Content](#)



## AlphaServer ES40 (10 PCI slots)

In addition to 10 PCI vacant slots there are 3 PCI positions occupied by on-board devices. All 13 PCI positions are listed in the following table in the order in which Alpha SRM console enumerates them.

Slot	pci_<N>	device	function	irq	Description	Preloaded Name
<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

## 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
```

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

```
set PKA container[602]="C:\My disks\vms_distribution.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> <ul style="list-style-type: none"> <li>• 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.</li> <li>• 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.</li> </ul> <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: connection_number = remote_scsi_id * 100 + lun_id</p> <p><b>Example:</b></p> <pre>set PKA port[600]=17060 host[600]="localhost:16070"</pre>

## container

<b>Parameter</b>	container[N] N is "XXYY" number, where XX = SCSI ID (0..15) and YY = LUN (00..07)
<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>"\\.\PhysicalDrive&lt;X&gt;", where X is 0, 1, ...               <p> 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.</p> <p><u>Example:</u></p> <pre>set PKA container[0]="\\.\PhysicalDrive1"</pre> </li> <li> <b>Physical disk by its WWID</b> <ul style="list-style-type: none"> <li>"\\.\PhysicalDrive(DevID=XXX-...-XXXX)"               <p> 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.</p> <p>DevID addresses the target physical disk by its WWID (hexadecimal 128-bit identifier assigned to the disk drive by its manufacturer/originator).</p> <p><u>Example:</u></p> <pre>set PKA container[100]="\\.\PhysicalDrive(DevID=6008-05F3-0005-2950-BF8E-0B86-A0C7-0001)"</pre> </li> </ul> </li> <li> <b>iSCSI disks</b> <ul style="list-style-type: none"> <li>"\\.\PhysicalDrive(iScsiTarget = &lt;iSCSI target&gt;, LUN = &lt;LUN number&gt;)"               <p>iScsiTarget addresses the disk by its iSCSI target name.</p> <p>LUN specifies LUN on connected iSCSI disk.</p> <p><u>Example:</u></p> <pre>set PKA container[200]="\\.\PhysicalDrive(iScsiTarget=iqn.2008-04:iscsi.charon-target-test1, LUN= 1)"</pre> </li> </ul> </li> <li> <b>Tape device</b> <ul style="list-style-type: none"> <li>"\\.\Tape&lt;X&gt;", where X is 0, 1, ...               <p><u>Example:</u></p> <pre>set PKA container[600]="\\.\Tape0"</pre> </li> </ul> </li> </ul> </li> </ul>

- **SCSI device unknown to Windows for direct mapping, for example, a SCSI disk or tape reader**

- "\\.\Scsi<N>:<X>:<Y>:<Z>"

The values of N, X, Y and Z can be collected using special utility "Host Device Check" included in the CHARON distributive - or manually by investigation of the devices connected to CHARON host in the "Device Manager" applet.

Parameter	Description
N	A logical number assigned by host operating system (Microsoft Windows) to logical or host's physical storage resource such as physical SCSI HBA
X	An internal SCSI bus number (usually 0) on host's physical SCSI HBA
Y	A SCSI ID of physical SCSI target device attached to host's physical SCSI HBA
Z	A logical unit number inside physical SCSI target device attached to host's physical SCSI HBA

Example:

```
set PKA container[202]="\\.\Scsi1:0:1:0"
```

- **Floppy drive**

- "\\.\A:"
- "\\.\B:"

Example:

```
set PKA container[300]="\\.\A:"
```

- **CDROM device**

- "\\.\CdRom<X>", where X is 0, 1, ...

Example:

```
set PKA container[400]="\\.\CdRom0"
```

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

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

Mapping may also include the full path (recommended), for example: "C:\My disks\vms\_distribution.iso"

Example:

```
set PKA container[600]="C:\My disks\vms_distribution.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 "MkDisk" utility. Data and OS disks 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: "C:\My disks\my\_boot\_disk.vdisk"

Example:

```
set PKA container[401]="tru64-v51-system.vdisk"
```

- **File representing the tape (tape image)**

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

The container-file to which an emulated tape drive is mapped is created automatically when the emulator starts, and when the device is accessed from within the guest operating system.

We recommend specifying the full path to the container-file, for example: "C:\MyTapes\backup.vtape"

**Example:**


```
set PKA container[500]="E:\Tapes\backup.vtape"
```

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

Guest-OS Operation	Emulator Action
Open device for writing	Create a container file if necessary; open for writing and lock the container file
Open device for reading	Create a container file if none exists. open for reading and lock 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. STROMASYS does not support writing multiple volumes to a single container file. Your backup procedure may need to be adapted to fulfill this restriction. See below for more information.

 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 mutli-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:...
```

- **Other type of drive, for example magneto-optical drive**

- "\\.\<N>:"

Example:


```
set PKA container[300]="\\.\Z:"
```

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

## media\_type

<b>Parameter</b>	media_type[N] N is "XXYY" number, where XX = SCSI ID (0..15) and YY = LUN (00..07)
<b>Type</b>	Text string
<b>Value</b>	Instructs CHARON-AXP to use the supplied value as the PRODUCT field in the SCSI INQUIRY data returned to a software running on virtual HP Alpha system in response to SCSI INQUIRY command.  If not specified, CHARON-AXP attempts to guess the SCSI INQUIRY data based on virtual SCSI device type and underlying container (which is specified in the corresponding container configuration parameter).  Initially is not specified.  <u>Example:</u>  <pre>set PKA media_type[0]="HSZ70"</pre>

## removable

<b>Parameter</b>	removable[N] N is "XXYY" number, where XX = SCSI ID (0..15) and YY = LUN (00..07)
<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".  <u>Example:</u>  <pre>set PKA removable[400]=true</pre>   Note that virtual SCSI tapes and CD-ROM devices are always reported as removable regardless of the "removable" configuration parameter.

## geometry

<b>Parameter</b>	<p>geometry [N]</p> <p>N is "XXYY" number, where XX = SCSI ID (0..15) and YY - LUN (00..07)</p>																
<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;][["/"&lt;B&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;"> <thead> <tr style="background-color: #d9e1f2;"> <th style="text-align: left;">Parameter</th> <th style="text-align: left;">Description</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">X</td> <td>The number of sectors per track</td> </tr> <tr> <td style="text-align: center;">Y</td> <td>The number of tracks per cylinder</td> </tr> <tr> <td style="text-align: center;">Z</td> <td> <p>The number of cylinders on the unit.</p> <p>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.</p> <p>This is an optional parameter.</p> </td> </tr> <tr> <td style="text-align: center;">B</td> <td> <p>The total size of the disk (in blocks) reported to the guest OS.</p> <p>If omitted it is calculated automatically.</p> <p>This is an optional parameter.</p> </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>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;"> <thead> <tr style="background-color: #d9e1f2;"> <th style="text-align: left;">Parameter</th> <th style="text-align: left;">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> <p>The size (in KB) of the space left on the tape when a warning to the OS is issued.</p> <p>If omitted, 64K is assumed.</p> </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	<p>The number of cylinders on the unit.</p> <p>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.</p> <p>This is an optional parameter.</p>	B	<p>The total size of the disk (in blocks) reported to the guest OS.</p> <p>If omitted it is calculated automatically.</p> <p>This is an optional parameter.</p>	Parameter	Description	image-size	The tape size in MB	early-warning-zone-size	<p>The size (in KB) of the space left on the tape when a warning to the OS is issued.</p> <p>If omitted, 64K is assumed.</p>
Parameter	Description																
X	The number of sectors per track																
Y	The number of tracks per cylinder																
Z	<p>The number of cylinders on the unit.</p> <p>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.</p> <p>This is an optional parameter.</p>																
B	<p>The total size of the disk (in blocks) reported to the guest OS.</p> <p>If omitted it is calculated automatically.</p> <p>This is an optional parameter.</p>																
Parameter	Description																
image-size	The tape size in MB																
early-warning-zone-size	<p>The size (in KB) of the space left on the tape when a warning to the OS is issued.</p> <p>If omitted, 64K is assumed.</p>																

## use\_io\_file\_buffering

<b>Parameter</b>	use_io_file_buffering[N] N is "XXYY" number, where XX = SCSI ID (0..15) and YY = LUN (00..07)
<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>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><u>Example:</u></p> <pre>set PKA use_io_file_buffering[603]=true</pre>

When a tape or disk image connected to an emulated KZPBA 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 "MkDisk" 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
  - storage\_controller\_path\_id
  - host\_bus\_location
  - wwid
  - container
  - media\_type
  - removable
  - geometry
  - use\_io\_file\_buffering
- Mapping to host resources
  - Fabric virtualization mode
  - Usage of "presentation mode"
  - Pass Through mode
    - Installation of EMULEX LightPulse PCI/PCI-X/PCIe FC adapter driver
    - Collecting the parameters for mapping
    - Configuration of KGPSA-CA in pass through mode
      - FCMGR utility description
      - Configuration steps using FCMGR utility

[Back to Table of Contents](#)

### General description

CHARON-AXP supports emulation of DEC-KGPSA-CA PCI Fibre Channel adapter.

Every instance of KGPSA-CA works in one of the three following modes:

- **Fabric virtualization mode** (mapping to disk images and host/SAN physical disks). This is default mode.
- **Usage of "presentation mode"** of external storage controllers to automatically configure KGPSA-CA disks correspondent to the LUNs provided by the external storages
- **CHARON PCI Pass Through mode** (using a specific CHARON PCI Pass Through driver to work through the EMULEX LightPulse PCI/PCI-X/PCIe FC adapter plugged into host)

[Back to Table of Contents](#)

### Loading KGPSA storage adapter

Syntax for loading KGPSA-CA storage adapter:

```
load KGPSA <name>
```

Example:

```
load KGPSA FGA
```

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

```
set FGA container[100]="C:\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:



## storage\_controller\_path\_id

<b>Parameter</b>	storage_controller_path_id
<b>Type</b>	Text String
<b>Value</b>	<p>Instructs the CHARON software to create a set of virtual FC devices and connect to the virtual system through the virtual KGPSA FC Adapter.</p> <p><b>i</b> In this mode WWID and UDID values are taken from the storage controller directly. Note that some storage controllers may not support acquiring UDID, in this case this way of mapping cannot be used in case of OpenVMS (Tru64 Unix is Ok)!</p> <p>Syntax:</p> <pre>load KGPSA &lt;controller name&gt; storage_controller_path_id = &lt;Storage controller path ID&gt;</pre> <p>where the "Storage controller path ID" parameter is a storage (for example SAN) controller path ID. This ID can be obtained from the "Host Device Check" utility ("Storage Controllers" section):</p>  <pre>load KGPSA FGA storage_controller_path_id = 5008-05F3-0005-2950-5008-05F3-0005-2951</pre> <p>It is also possible to use wildcards for specifying the ID number, for example:</p> <pre>load KGPSA FGA storage_controller_path_id = 5008-05F3-0005-2950-*-05F3-0005-*</pre> <p>In this case CHARON will collect the LUNs from all the matching paths.</p> <p>This syntax works disregarding whether Windows MPIO is active or not. If it is active, CHARON uses "active" path (because it works through Windows), if Windows MPIO is OFF, CHARON selects the first matching path (and warns you in the LOG file).</p> <p><b>i</b> Note that LUNs obtained by usage of "storage_controller_path_id" parameter is not re-enumerated on reboot, so this mode is possible to use in case of shared disks clusters.</p>

## host\_bus\_location

<b>Parameter</b>	host_bus_location
<b>Type</b>	Text String
<b>Value</b>	<p>Establish connection between virtual DEC-KGPSA-CA PCI FC adapter and physical EMULEX LightPulse PCI/PCI-X/PCle FC adapter (pass through mode)</p> <p>Syntax:</p> <pre>load KGPSA &lt;controller name&gt; host_bus_location="PCI bus X, device Y, function Z"</pre> <p>where X, Y, and Z describe location of physical EMULEX LightPulse PCI/PCI-X/PCle FC adapter in the host computer (see <a href="#">this section</a> for details).</p> <p><b>Example:</b></p> <pre>load KGPSA FGA host_bus_location="PCI bus 3, device 1, function 0"</pre>

## wwid

<b>Parameter</b>	<p>wwid[N]</p> <p>N is 0..32766 (no more than 255 units)</p>
<b>Type</b>	Text String
<b>Value</b>	<p>Sets WWID for emulated KGPSA adapter unit in case if the WWID can not be obtained directly from the device CHARON is mapped to.</p> <p>For example in case of disk images or local physical drives CHARON cannot get its WWID from the the host system, so it is automaticaly introduce some generated WWID. In case if this WWID is not acceptable it can be replaced with the custom one using the "wwid" parameter.</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**

- "\\.\PhysicalDrive<X>", where X is 0, 1, ...

**!** Be careful not to destroy all the information from the disk dedicated to CHARON-AXP by mistake. These disks must not be formatted by the host OS.

**i** In this mode WWID is generated by CHARON and UDID is defined by the index of the "container" parameter.  
**Example:**

```
set FGA container[0]="\\.\PhysicalDrive1"
```

In case "\\.\PhysicalDrive<N>" points to some external iSCSI or FC disk note that the disks re-enumeration may appear on CHARON host reboot, for example an external disk originally referenced as "\\.\PhysicalDrive3" may become "\\.\PhysicalDrive1" etc. Especially it is important in shared disks cluster configurations!

So it is strictly recommended to use "Mapping by WWID" / "Mapping iSCSI disks" ways of mapping in these situations (see below). The mapping by using "storage\_controller\_path\_id" (see above) is acceptable as well, since it is not re-enumerating LUNs on each re-boot or CHARON restart.

▪ **Physical disk by its WWID**

- "\\.\PhysicalDrive(DevID =XXXX-XXXX-XXXX-XXXX-XXXX-XXXX-XXXX-XXXX)"

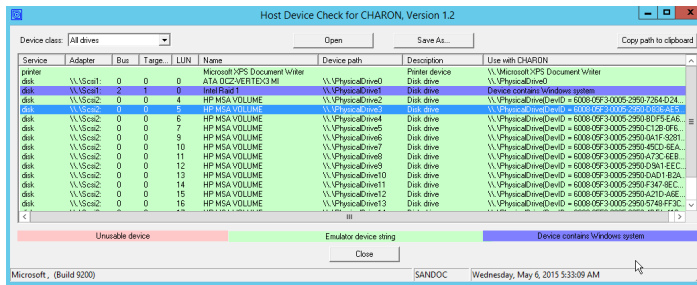
**!** Be careful not to destroy all the information from the disk dedicated to CHARON-AXP by mistake. These disks must not be formatted by the host OS.

DevID addresses the target physical disk by its WWID (hexadecimal 128-bit identifier assigned to the disk drive by its manufacturer/originator).

**i** In this mode WWID is generated by CHARON and UDID is defined by the index of the "container" parameter.  
**Example:**

```
set FGA container[100]="\\.\PhysicalDrive(DevID= 6008-05F3-0005-2950-BF8E-0B86-A0C7-0001)"
```

The WWID values can be obtained from "Host Device Check" utility ("All drives" section), for example:



- **iSCSI disks**

- "\.PhysicalDrive(iSCSITarget = <iSCSI target>, LUN = <LUN number>)"

iSCSITarget addresses the disk by its iSCSI target name. LUN specifies LUN on connected iSCSI disk.

**i** In this mode WWID is generated by CHARON and UDID is defined by the index of the "container" parameter.

Example:

```
set FGA container[200]="\\.PhysicalDrive(iSCSITarget= iqn.2008-04:iSCSI.charon-target-test1, LUN= 1)"
```

- **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 "MkDisk" utility. Data and OS disks 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: "C:\My disks\my\_boot\_disk.vdisk"

Example:

```
set FGA container[401]="my_dka401.vdisk"
```

**i** 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>	Instructs CHARON-AXP to use the supplied value as the PRODUCT field in the FC INQUIRY data returned to a software running on virtual HP Alpha system in response to FC INQUIRY command.  If not specified, CHARON-AXP attempts to guess the FC INQUIRY data based on virtual FC device type and underlying container (which is specified in the corresponding container configuration parameter).  Initially is not specified.  <u>Example:</u>  <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>	When set to "true", the removable configuration parameter instructs CHARON-AXP to report the corresponding virtual FC device as removable.  By default the removable configuration parameter is set to "false".  <u>Example:</u>  <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;]["/&lt;B&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>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 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 "MkDisk" 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

There are 3 modes of KGPSA-CA mapping to system resources:

Mode	Description	Pros	Cons
Fabric virtualization mode	Virtual KGPSA-CA PCI FC adapter is mapped to physical disks (both local, FC and iSCSI) and disk images   This is default mode, suitable for most cases	<ul style="list-style-type: none"> <li>High I/O performance using modern local SSD disks.</li> </ul>	<ul style="list-style-type: none"> <li>Requires CHARON-AXP reconfiguration and restart on any change to the mapped disk images and physical drives</li> <li>Disk images cannot be used in case of Tru64 cluster, but acceptable for OpenVMS cluster.</li> </ul>
"Presentation" mode	CHARON-AXP automatically creates/removes a set of virtual FC devices on a virtual KGPSA-CA FC adapter correspondent to available external SAN LUNs   This mode is recommended in case of working with SAN that may change LUN configuration - so CHARON-AXP will reflect it on-fly w/o reconfiguration and restarting.	<ul style="list-style-type: none"> <li>No reconfiguration and restart of CHARON-AXP is needed on change to the virtual FC drives provided as LUNs by the connected SAN.</li> <li>Effectively any SAN FC system could be used, that means newer, faster, larger, FC HBA controllers and device drivers supported by newer, faster host server platforms and operating systems - it offers significant performance capabilities and performance improvements.</li> </ul>	<ul style="list-style-type: none"> <li>Target configurations are restricted to the SAN models capable to work with HW Alphas (for example acquiring UDID must be supported in case of OpenVMS). This restriction comes from the fact that different SAN controllers could have different level of SCSI-3 protocol support, and if apparently some individual SAN model does not properly support complete set of SCSI-3 commands used by OpenVMS &amp; Tru64 KGPSA-CA drivers, then with high probability such SAN Controller will not be able to work with CHARON-AXP in "presentation" (as well as in Pass Through) Mode.</li> </ul>
Pass Through mode	This mode allows connection between virtual KGPSA-CA FC adapter and physical EMULEX LightPulse PCI/PCI-X/PCIe FC adapter plugged into host's PCI/PCI-X/PCIe slot   This mode is recommended for cluster setups that include HW Alphas	<ul style="list-style-type: none"> <li>High performance</li> <li>Support of shared disks and cluster configuration that includes HW Alphas</li> </ul>	<ul style="list-style-type: none"> <li>Cannot be used in VMware environment</li> <li>Obsolete PCI-X cards are not available anymore, HBA list is restricted to several EMULEX HBA models only. No way to use HBA from other vendors.</li> <li>Target configurations are restricted to SAN models capable to work with HW Alphas.</li> <li>As EMULEX LightPulse PCI/PCI-X/PCIe FC adapters have been never used on Alphas, formally there is a probability that some compatibility problems may be detected.</li> </ul>

All the described modes can be used in case of a cluster including HW Alphas and shared SAN, but pay attention that all the disks must look absolutely the same to the operating systems on all the cluster nodes, including CHARON: same WWID, same UDID (in case of OpenVMS), same size, same geometry etc.

## 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]="C:\My disks\my_dka401.vdisk"
set FGA container[100]="\\.\PhysicalDrive1"
set FGA container[200]="\\.\PhysicalDrive(DevID= 6008-05F3-0005-2950-BF8E-0B86-A0C7-0001)"
set FGA container[300]="\\.\PhysicalDrive(iSCSITarget= ign.2008-04:iSCSI.charon-target-test1, LUN= 1)"
```

**!** Use only "Mapping by WWID" and "Mapping iSCSI disks" syntax in case of the mapped physical disk located on some external disk controller (SAN, etc) since the "\\.\PhysicalDrive<N>" re-enumeration may happen on CHARON host reboot.

**i** Note that WWID and UDID is generated by CHARON in its mode. It may be important in case of shared disks cluster configurations.

See the "Configuration parameters" section for details.

## Usage of "presentation mode"

CHARON-AXP can utilize so called "presentation mode" of the storage controller it maps to.

In this mode CHARON-AXP creates a set of virtual FC devices and connects to the SAN system through the virtual KGPSA FC adapter.

The main benefit of this mode is a flexible way of CHARON-AXP virtual disks management depending on the mapped storage controller configuration. LUN parameters of any available to CHARON host storage controller specified in CHARON-AXP configuration file via "storage\_controller\_path\_id" parameter are automatically scanned and then assigned as 'virtual' FC drives which guest OS can see, no need to manually change/redefine/modify virtual LUN parameters in emulator.

For example if an extra disk has been added to the storage controller it will automatically appear as a new disk unit on the corresponding virtual KGPSA-CA virtual controller mapped to that storage controller.

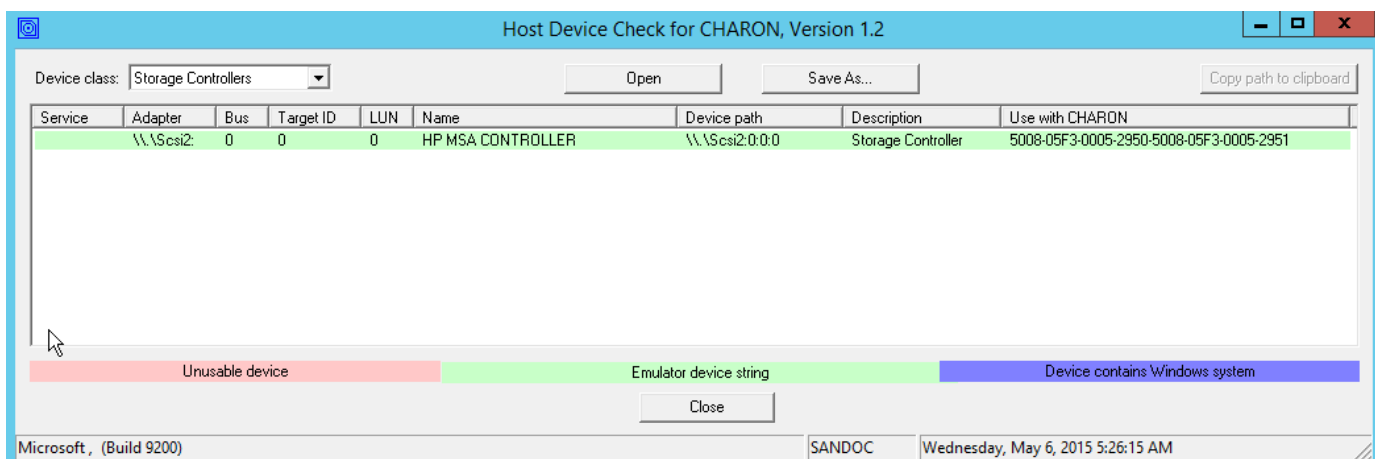
In "presentation mode" CHARON-AXP supports the following operations with FC disks (similar to ones supported in KGPSA FC Pass Through mode) w/o need of CHARON-AXP emulator restart:

- Dynamically allocation (retranslation) of new created SAN LUN to virtual Alpha configuration
- Scan and translation of SAN LUN parameters to virtual Alpha's KGPSA DGxx: virtual drive parameters
- Propagation of SAN LUN size change to virtual Alpha's KGPSA virtual drive parameters
- Automatically deallocation from virtual Alpha deleted SAN LUN

Example of configuration:

```
load KGPSA FGA storage_controller_path_id = 5008-05F3-0005-2950-5008-05F3-0005-2951
```

The ID number specified in the example above is a storage (for example SAN) controller path ID. This ID can be obtained from the "Host Device Check" utility ("Storage Controllers" section):



Once specified, all the disks attached to the storage are automatically mapped as disk units to CHARON.

It is also possible to use wildcards for specifying the ID number:

```
load KGPSA FGA storage_controller_path_id = 5008-05F3-0005-2950-* -05F3-0005-*
```

In this case CHARON will collect the LUNs from all the matching paths.

This syntax works disregarding whether Windows MPIO is active or not. If it is active, CHARON uses "active" path (because it works through Windows), if Windows MPIO is OFF, CHARON selects the first matching path (and warns you in the LOG file).

**Usage of LUNs in cluster configuration:**

Note that PATHs (on OpenVMS level) do not affect OpenVMS cluster. What really makes sense is LUN size, geometry, WWI, etc. CHARON obtains these properties from the LUN (within the limits established by Windows), so LUNs should appear similar to all cluster members.

**i** Note that LUNs obtained by usage of "storage\_controller\_path\_id" parameter is not re-enumerated on reboot, so this mode is possible to use in case of shared disks clusters.

**i** Note that the WWID and UDID are automatically obtained from disk server (SAN) in this mode - it is very convenient in case of shared disks cluster configurations. Also note that some storage controllers may not support acquiring UDID - in this case this way of mapping cannot be used in case of OpenVMS (Tru64 Unix is Ok)!

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="PCI bus X, device Y, function Z"
```

Example:

```
load KGPSA FGA host_bus_location="PCI bus 3, device 1, function 0"
```

The "host\_bus\_location" parameter addresses the host EMULEX LightPulse PCI/PCI-X/PCIe FC adapter in the following way:

Parameters	Description
"PCI bus X"	PCI bus number of the host EMULEX LightPulse PCI/PCI-X/PCIe FC adapter
"device Y"	PCI bus device number of the host EMULEX LightPulse PCI/PCI-X/PCIe FC adapter
"function Z"	The "function" parameter of the the host EMULEX LightPulse PCI/PCI-X/PCIe FC adapter

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



Also see the "Configuration parameters" section for details

[Back to Table of Contents](#)

### Installation of EMULEX LightPulse PCI/PCI-X/PCIe FC adapter driver

1. Install the EMULEX LightPulse PCI/PCI-X/PCIe FC adapter (see below for a list of supported models) to some spare PCI/PCI-X/PCIe slot of the host system
2. Boot a Windows operating system
3. Install the EMULEX LightPulse PCI/PCI-X/PCIe FC adapter driver from the following directory "C:\Program Files\CHARON\Drivers\EMULEX\_X.X.0.XXXXX" by choosing the "Install from a list or specific location (Advanced)" option and then selecting the "emulex\_lp\_ppt\_amd64.inf" file.
4. Direct Windows to use this custom driver instead of the default one:
  - a. Type "gpedit.msc" in the "Search programs and files" field under "Start" menu, press "Enter"
  - b. In the appearing applet choose: Administrative Templates -> System -> Device Installation -> Device Installation Restrictions -> Prevent Installation of Devices not described by other policy settings
  - c. Change the default "Not configured" to "Disabled"; press "Ok" to apply,
5. Reboot the host

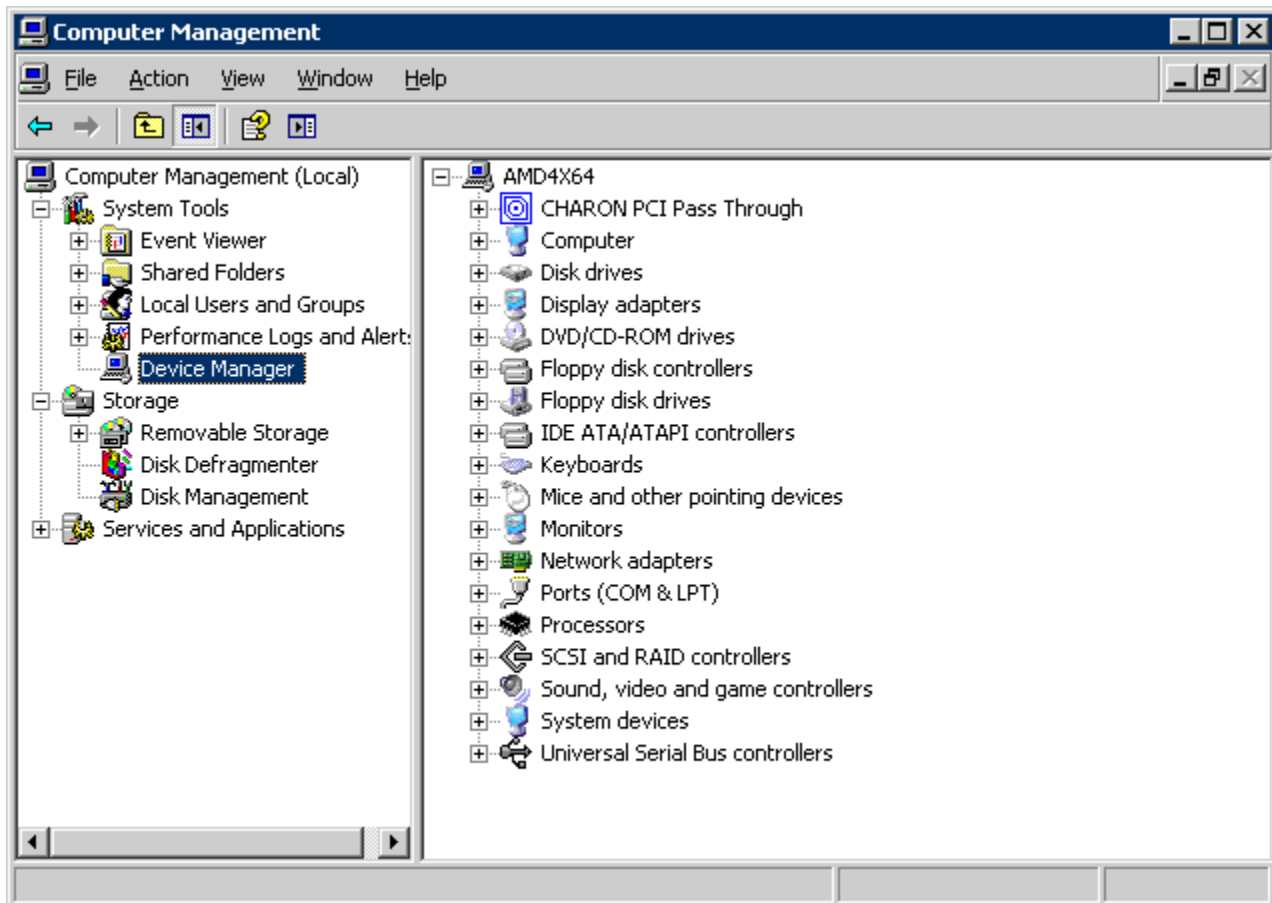
[Back to Table of Contents](#)

### Collecting the parameters for mapping

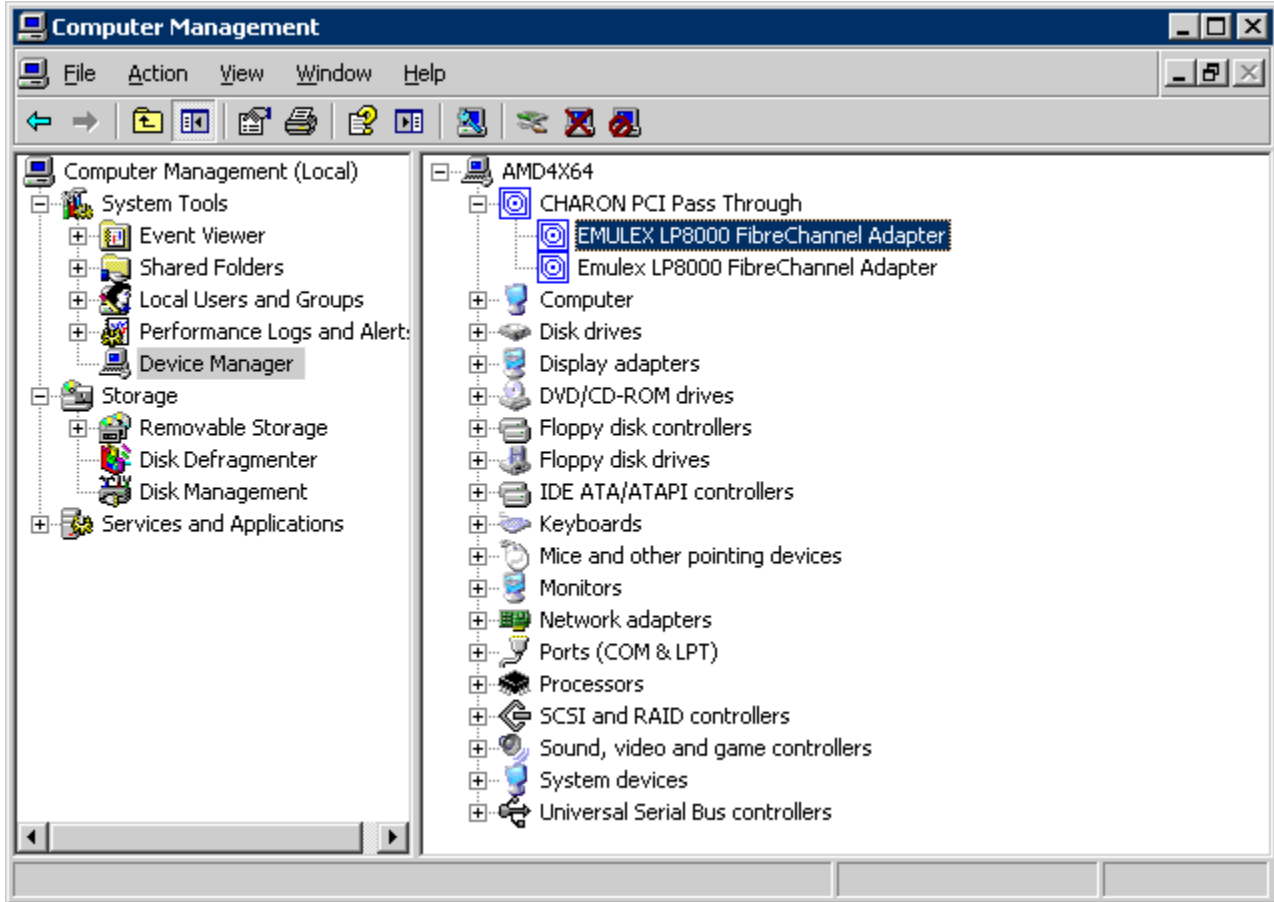
There are 2 ways of getting the required parameters for mapping virtual KGPSA-CA to EMULEX LightPulse PCI/PCI-X/PCIe FC adapter plugged into host's PCI/PCI-X/PCIe slot:

1. Usage of "HOST Device Check" utility. Follow the link for details.
2. Collecting the parameters directly using "Device Manager" applet

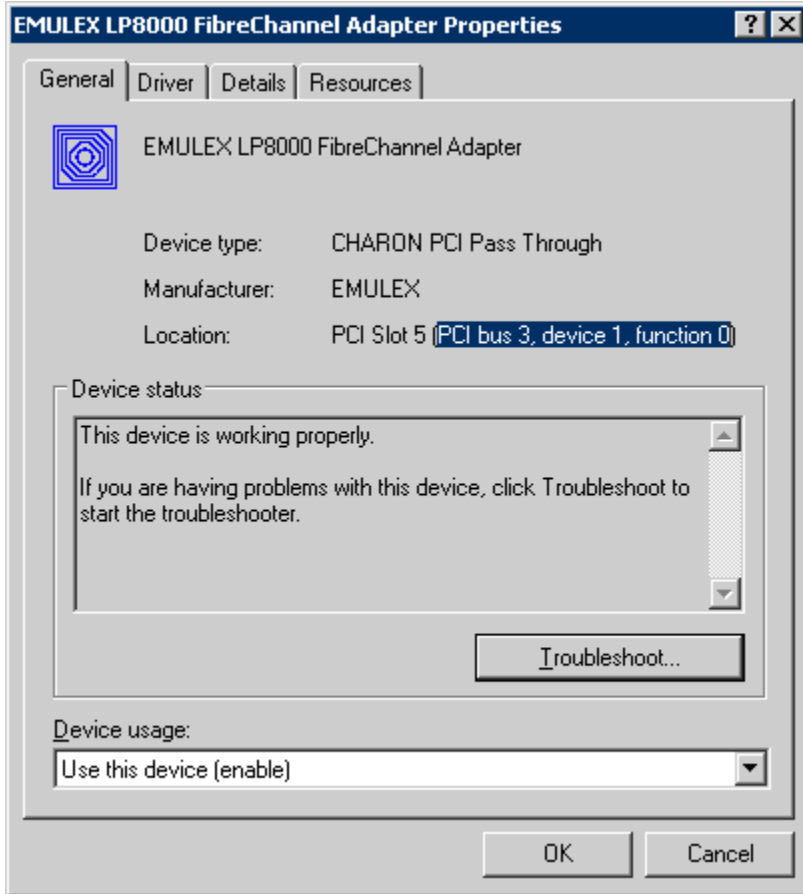
To collect the parameters directly using "Device Manager" applet, open "Computer Management" application and select "Device Manager":



In the right panel select desired physical EMULEX LP FibreChannel adapter under "CHARON PCI Pass Through":



Open its properties sheet by double-clicking on the selected adapter:



The "Location:" on the above picture gives X, Y, and Z for the "host\_bus\_location" parameter, for example:

```
load KGPSA FGA host_bus_location = "PCI bus 3, device 1, function 0"
```

**!** Non-US-EN installations of Windows may present "Location:" string in local language, but "host\_bus\_location" parameter requires English notation, so the words "PCI", "bus", "device", and "function" must be specified in English.

### ***Configuration of KGPSA-CA in pass through mode***

**FCMGR utility description**

To configure KGPSA-CA adapter in pass through mode a special SRM console utility "FCMGR" is used (it has the same functionality as the "WWIDMGR" utility of the native HP Alpha hardware).

It provides the following functionality:

Command	Description
<code>fc rescan {/verbose}</code>	Scans connected SAN using FC adapters, discovers FC ports, storage controllers, logical units and then builds volatile FC database.  The "/verbose" qualifier enables FC communication trace on console (for diagnostic and troubleshooting).
<code>fc show {adapter   port   device}</code>	Displays corresponding part of volatile FC database.
<code>fc set {boot   dump} udid &lt;X&gt;</code>	Fills the environment variables <code>wwid0..wwid3</code> and <code>n1..n4</code> to identify path(s) to logical unit with the specified UDID.  These variables are later used by "INIT" to create device database entries and by OpenVMS/Tru64 to get access to boot and dump disks.  This command does NOT make any change to other environment variables.
<code>fc clear</code>	Clears environment variables <code>wwid0..wwid3</code> and <code>n1..n4</code> , which automatically disable (but do NOT delete) device database entries representing FC attached devices.  This command does NOT affect volatile FC database.

**Example of usage:**

```
P00>>>fc show devices
  UDID:110 WWID:01000010:6000-1fe1-000b-6bf0-0009-9081-1283-0081 (ev:none)
      via adapter      via fc_port      con
[0]      pga0.0.0.5.1    5000-1fe1-000b-6bf1 yes (ev:none)
[1]      pga0.0.0.5.1    5000-1fe1-000b-6bf4 yes (ev:none)
  UDID:108 WWID:01000010:6000-1fe1-000b-6bf0-0009-0440-4014-0039 (ev:none)
      via adapter      via fc_port      con
[12]     pga0.0.0.5.1    5000-1fe1-000b-6bf1 yes (ev:none)
[13]     pga0.0.0.5.1    5000-1fe1-000b-6bf4 yes (ev:none)
```

**Configuration steps using FCMGR utility**

Once the configuration steps described above are done, start the CHARON-AXP and wait for the **P00>>>** prompt.

Please refer to the following example with two FC adapters PGA and PGB defined:

```

initializing ...

polling for units on kzpba0, slot 4, bus 0, hose 0 ...
  pka0.0.0.4.0          PKA0          Q-Logic/ISP PCI SCSI HBA

polling for units on kgpsa0, slot 5, bus 0, hose 0 ...
  pga0.0.0.5.0          PGA0          WWN 1000-0000-C92E-97C9
    fabric              WWN 2003-0060-6920-4682
    directory          WWN 20fc-0060-6920-4682
    port 021400        WWN 5000-1fe1-000b-6bf1
      lun 000000000000100    DEC      HSG80      V88F
    UDID:100          WWID:01000010:6000-1fe1-000b-6bf0-0009-0440-4014-0038
      lun 000000000000200    DEC      HSG80      V88F
    UDID:200          WWID:01000010:6000-1fe1-000b-6bf0-0009-9081-1283-0074
      lun 000000000000300    DEC      HSG80      V88F
    UDID:300          WWID:01000010:6000-1fe1-000b-6bf0-0009-9081-1283-007b
      lun 000000000000400    DEC      HSG80      V88F
    UDID:400          WWID:01000010:6000-1fe1-000b-6bf0-0009-9081-1283-0080
      lun 0000000000006c00    DEC      HSG80      V88F
    UDID:108          WWID:01000010:6000-1fe1-000b-6bf0-0009-0440-4014-0039
      lun 0000000000006d00    DEC      HSG80      V88F
    UDID:208          WWID:01000010:6000-1fe1-000b-6bf0-0009-0440-4014-003a
      lun 0000000000000000

    port 021500          WWN 5000-1fe1-000b-6bf4
      lun 000000000000100    DEC      HSG80      V88F
    UDID:100          WWID:01000010:6000-1fe1-000b-6bf0-0009-0440-4014-0038
      lun 000000000000200    DEC      HSG80      V88F
    UDID:200          WWID:01000010:6000-1fe1-000b-6bf0-0009-9081-1283-0074
      lun 000000000000300    DEC      HSG80      V88F
    UDID:300          WWID:01000010:6000-1fe1-000b-6bf0-0009-9081-1283-007b
      lun 000000000000400    DEC      HSG80      V88F
    UDID:400          WWID:01000010:6000-1fe1-000b-6bf0-0009-9081-1283-0080
      lun 0000000000006c00    DEC      HSG80      V88F
    UDID:108          WWID:01000010:6000-1fe1-000b-6bf0-0009-0440-4014-0039
      lun 0000000000006d00    DEC      HSG80      V88F
    UDID:208          WWID:01000010:6000-1fe1-000b-6bf0-0009-0440-4014-003a
      lun 0000000000000000

```

```
polling for units on kgpsal, slot 6, bus 0, hose 0 ...
pgb0.0.0.6.0      PGB0      WWN 1000-0000-C92D-8D00
fabric           WWN 2003-0060-6920-45ff
directory       WWN 20fc-0060-6920-45ff
port 011400     WWN 5000-1fel-000b-6bf2
  lun 0000000000000100    DEC    HSG80      V88F
UDID:100 WWID:01000010:6000-1fel-000b-6bf0-0009-0440-4014-0038
  lun 0000000000000200    DEC    HSG80      V88F
UDID:200 WWID:01000010:6000-1fel-000b-6bf0-0009-9081-1283-0074
  lun 0000000000000300    DEC    HSG80      V88F
UDID:300 WWID:01000010:6000-1fel-000b-6bf0-0009-9081-1283-007b
  lun 0000000000000400    DEC    HSG80      V88F
UDID:400 WWID:01000010:6000-1fel-000b-6bf0-0009-9081-1283-0080
  lun 00000000000006c00    DEC    HSG80      V88F
UDID:108 WWID:01000010:6000-1fel-000b-6bf0-0009-0440-4014-0039
  lun 00000000000006d00    DEC    HSG80      V88F
UDID:208 WWID:01000010:6000-1fel-000b-6bf0-0009-0440-4014-003a
  lun 0000000000000000
port 011500     WWN 5000-1fel-000b-6bf3
  lun 0000000000000100    DEC    HSG80      V88F
UDID:100 WWID:01000010:6000-1fel-000b-6bf0-0009-0440-4014-0038
  lun 0000000000000200    DEC    HSG80      V88F
UDID:200 WWID:01000010:6000-1fel-000b-6bf0-0009-9081-1283-0074
  lun 0000000000000300    DEC    HSG80      V88F
UDID:300 WWID:01000010:6000-1fel-000b-6bf0-0009-9081-1283-007b
  lun 0000000000000400    DEC    HSG80      V88F
UDID:400 WWID:01000010:6000-1fel-000b-6bf0-0009-9081-1283-0080
  lun 00000000000006c00    DEC    HSG80      V88F
UDID:108 WWID:01000010:6000-1fel-000b-6bf0-0009-0440-4014-0039
  lun 00000000000006d00    DEC    HSG80      V88F
UDID:208 WWID:01000010:6000-1fel-000b-6bf0-0009-0440-4014-003a
  lun 0000000000000000
port 011100     failed port login

... enter console

CHARON-AXP (AlphaServer ES40) emulator. Version 4.6
Copyright (C) 2015, STROMASYS (www.stromasys.com)

P00>>>
```

The next step is to configure paths for the FC storage:

```
P00>>>fc show devices

  UDID:100 WWID:01000010:6000-1fe1-000b-6bf0-0009-0440-4014-0038 (ev:none)
      via adapter      via fc_port      con
[0]      pga0.0.0.5.0      5000-1fe1-000b-6bf1 no (ev:none)
[1]      pga0.0.0.5.0      5000-1fe1-000b-6bf4 yes (ev:none)
[2]      pgb0.0.0.6.0      5000-1fe1-000b-6bf2 no (ev:none)
[3]      pgb0.0.0.6.0      5000-1fe1-000b-6bf3 yes (ev:none)
  UDID:200 WWID:01000010:6000-1fe1-000b-6bf0-0009-9081-1283-0074 (ev:none)
      via adapter      via fc_port      con
[4]      pga0.0.0.5.0      5000-1fe1-000b-6bf1 no (ev:none)
[5]      pga0.0.0.5.0      5000-1fe1-000b-6bf4 yes (ev:none)
[6]      pgb0.0.0.6.0      5000-1fe1-000b-6bf2 no (ev:none)
[7]      pgb0.0.0.6.0      5000-1fe1-000b-6bf3 yes (ev:none)
  UDID:300 WWID:01000010:6000-1fe1-000b-6bf0-0009-9081-1283-007b (ev:none)
      via adapter      via fc_port      con
[8]      pga0.0.0.5.0      5000-1fe1-000b-6bf1 no (ev:none)
[9]      pga0.0.0.5.0      5000-1fe1-000b-6bf4 yes (ev:none)
[10]     pgb0.0.0.6.0      5000-1fe1-000b-6bf2 no (ev:none)
[11]     pgb0.0.0.6.0      5000-1fe1-000b-6bf3 yes (ev:none)
  UDID:400 WWID:01000010:6000-1fe1-000b-6bf0-0009-9081-1283-0080 (ev:none)
      via adapter      via fc_port      con
[12]     pga0.0.0.5.0      5000-1fe1-000b-6bf1 no (ev:none)
[13]     pga0.0.0.5.0      5000-1fe1-000b-6bf4 yes (ev:none)
[14]     pgb0.0.0.6.0      5000-1fe1-000b-6bf2 no (ev:none)
[15]     pgb0.0.0.6.0      5000-1fe1-000b-6bf3 yes (ev:none)
  UDID:108 WWID:01000010:6000-1fe1-000b-6bf0-0009-0440-4014-0039 (ev:none)
      via adapter      via fc_port      con
[16]     pga0.0.0.5.0      5000-1fe1-000b-6bf1 yes (ev:none)
[17]     pga0.0.0.5.0      5000-1fe1-000b-6bf4 no (ev:none)
[18]     pgb0.0.0.6.0      5000-1fe1-000b-6bf2 yes (ev:none)
[19]     pgb0.0.0.6.0      5000-1fe1-000b-6bf3 no (ev:none)
  UDID:208 WWID:01000010:6000-1fe1-000b-6bf0-0009-0440-4014-003a (ev:none)
      via adapter      via fc_port      con
[20]     pga0.0.0.5.0      5000-1fe1-000b-6bf1 yes (ev:none)
[21]     pga0.0.0.5.0      5000-1fe1-000b-6bf4 no (ev:none)
[22]     pgb0.0.0.6.0      5000-1fe1-000b-6bf2 yes (ev:none)
[23]     pgb0.0.0.6.0      5000-1fe1-000b-6bf3 no (ev:none)

P00>>>fc set boot udid 400

P00>>>INIT

initializing ...

polling for units on kzpba0, slot 4, bus 0, hose 0 ...
  pka0.0.0.4.0      PKA0      Q-Logic/ISP PCI SCSI HBA
```

```

polling for units on kgpsa0, slot 5, bus 0, hose 0 ...
  pga0.0.0.5.0      PGA0      WWN 1000-0000-C92E-97C9
    fabric          WWN 2003-0060-6920-4682
      directory    WWN 20fc-0060-6920-4682
      port 021400  WWN 5000-1fe1-000b-6bf1
        lun 0000000000000100  DEC  HSG80  V88F
      UDID:100      WWID:01000010:6000-1fe1-000b-6bf0-0009-0440-4014-0038
        lun 0000000000000200  DEC  HSG80  V88F
      UDID:200      WWID:01000010:6000-1fe1-000b-6bf0-0009-9081-1283-0074
        lun 0000000000000300  DEC  HSG80  V88F
      UDID:300      WWID:01000010:6000-1fe1-000b-6bf0-0009-9081-1283-007b
        lun 0000000000000400  DEC  HSG80  V88F
      UDID:400      WWID:01000010:6000-1fe1-000b-6bf0-0009-9081-1283-0080
        lun 0000000000006c00  DEC  HSG80  V88F
      UDID:108      WWID:01000010:6000-1fe1-000b-6bf0-0009-0440-4014-0039
        lun 0000000000006d00  DEC  HSG80  V88F
      UDID:208      WWID:01000010:6000-1fe1-000b-6bf0-0009-0440-4014-003a
        lun 0000000000000000
      port 021500      WWN 5000-1fe1-000b-6bf4
        lun 0000000000000100  DEC  HSG80  V88F
      UDID:100      WWID:01000010:6000-1fe1-000b-6bf0-0009-0440-4014-0038
        lun 0000000000000200  DEC  HSG80  V88F
      UDID:200      WWID:01000010:6000-1fe1-000b-6bf0-0009-9081-1283-0074
        lun 0000000000000300  DEC  HSG80  V88F
      UDID:300      WWID:01000010:6000-1fe1-000b-6bf0-0009-9081-1283-007b
        lun 0000000000000400  DEC  HSG80  V88F
      UDID:400      WWID:01000010:6000-1fe1-000b-6bf0-0009-9081-1283-0080
        lun 0000000000006c00  DEC  HSG80  V88F
      UDID:108      WWID:01000010:6000-1fe1-000b-6bf0-0009-0440-4014-0039
        lun 0000000000006d00  DEC  HSG80  V88F
      UDID:208      WWID:01000010:6000-1fe1-000b-6bf0-0009-0440-4014-003a
        lun 0000000000000000

polling for units on kgpsal, slot 6, bus 0, hose 0 ...
  pgb0.0.0.6.0      PGB0      WWN 1000-0000-C92D-8D00
    fabric          WWN 2003-0060-6920-45ff
      directory    WWN 20fc-0060-6920-45ff
      port 011400  WWN 5000-1fe1-000b-6bf2
        lun 0000000000000100  DEC  HSG80  V88F
      UDID:100      WWID:01000010:6000-1fe1-000b-6bf0-0009-0440-4014-0038
        lun 0000000000000200  DEC  HSG80  V88F
      UDID:200      WWID:01000010:6000-1fe1-000b-6bf0-0009-9081-1283-0074
        lun 0000000000000300  DEC  HSG80  V88F
      UDID:300      WWID:01000010:6000-1fe1-000b-6bf0-0009-9081-1283-007b
        lun 0000000000000400  DEC  HSG80  V88F
      UDID:400      WWID:01000010:6000-1fe1-000b-6bf0-0009-9081-1283-0080
        lun 0000000000006c00  DEC  HSG80  V88F
      UDID:108      WWID:01000010:6000-1fe1-000b-6bf0-0009-0440-4014-0039
        lun 0000000000006d00  DEC  HSG80  V88F
      UDID:208      WWID:01000010:6000-1fe1-000b-6bf0-0009-0440-4014-003a
        lun 0000000000000000
      port 011500      WWN 5000-1fe1-000b-6bf3
        lun 0000000000000100  DEC  HSG80  V88F
      UDID:100      WWID:01000010:6000-1fe1-000b-6bf0-0009-0440-4014-0038
        lun 0000000000000200  DEC  HSG80  V88F
      UDID:200      WWID:01000010:6000-1fe1-000b-6bf0-0009-9081-1283-0074
        lun 0000000000000300  DEC  HSG80  V88F
      UDID:300      WWID:01000010:6000-1fe1-000b-6bf0-0009-9081-1283-007b
        lun 0000000000000400  DEC  HSG80  V88F
      UDID:400      WWID:01000010:6000-1fe1-000b-6bf0-0009-9081-1283-0080
        lun 0000000000006c00  DEC  HSG80  V88F
      UDID:108      WWID:01000010:6000-1fe1-000b-6bf0-0009-0440-4014-0039
        lun 0000000000006d00  DEC  HSG80  V88F
      UDID:208      WWID:01000010:6000-1fe1-000b-6bf0-0009-0440-4014-003a
        lun 0000000000000000
      port 011100      failed port login

```



```
... enter console

CHARON-AXP (AlphaServer ES40) emulator. Version 4.6
Copyright (C) 2015, STROMASYS (www.stromasys.com)

P00>>>SHOW DEV

sys0.0.0.0.0          SYS0          System ROOT Device
ewa0.0.0.3.0          EWA0          00-51-71-F5-8E-D8
pka0.0.0.4.0          PKA0          Q-Logic/ISP PCI SCSI HBA
pga0.0.0.5.0          PGA0          WWN 1000-0000-C92E-97C9
pgb0.0.0.6.0          PGB0          WWN 1000-0000-C92D-8D00
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          Virtual ATAPI - TEAC DW-224E-V
dka0.0.0.4.0          DKA0          Virtual SCSI Disk (C)SRI
dga400.1001.0.5.0     $1$DGA400    DEC      HSG80      V88F
dga400.1002.0.5.0     $1$DGA400    DEC      HSG80      V88F
dgb400.1003.0.6.0     $1$DGA400    DEC      HSG80      V88F
dgb400.1004.0.6.0     $1$DGA400    DEC      HSG80      V88F

P00>>>BOOT $1$DGA400

dga400.1001.0.5.0: failed to open device
(boot dga400.1002.0.5.0)
jumping to bootstrap code

    OpenVMS (TM) Alpha Operating System, Version V7.3-2
    © Copyright 1976-2003 Hewlett-Packard Development Company, L.P.
%SMP-I-CPUTRN, CPU #02 has joined the active set.
%SMP-I-CPUTRN, CPU #03 has joined the active set.
%SMP-I-CPUTRN, CPU #01 has joined the active set.
Please enter date and time (DD-MMM-YYYY HH:MM)
```

[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="\\.\CdRom0"
```

[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>	<p>Specifies the name of ATAPI or SATA CD/DVD-ROM drive attached to the host system.</p> <p>The supported values are of the form "\\.\CdRomN", where N is 0, 1, 2...</p> <p>In most cases 0 is the only meaningful value for N, because usually the host system has only one CD/DVD-ROM drive.</p> <p>By default it is left unspecified.</p>

## Example

```
set ide container="\\.\CdRom0"
```

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](#)

# 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
  - suspend\_msg\_on\_mac\_change
  - legacy\_mode
  - 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 Windows host system.

All the emulated controllers are loaded and configured in the same way.

 The Ethernet adapter in the Windows 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>

**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"

Load "packet\_port" to connect network adapter to the host hardware network card (or to a virtual network interface).

Example:

```
load packet_port/chnetwrk NDIS interface = "connection:Charon"
```

### 3. Connect the loaded "packet\_port" to the loaded virtual network adapter

Connect the network adapter to the "packet\_port" by setting the interface name.

Example:

```
set NIC interface = NDIS
```

[Back to Table of Contents](#)

## Configuration parameters

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

### interface

<b>Parameter</b>	interface
<b>Type</b>	Text String
<b>Value</b>	Name of the corresponding instance of the "packet_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> <div style="border: 1px solid black; padding: 2px; width: fit-content; margin: 5px 0;">XX-XX-XX-XX-XX-XX</div> <p>or</p> <div style="border: 1px solid black; padding: 2px; width: fit-content; margin: 5px 0;">XX:XX:XX:XX:XX:XX</div> <p><u>Example:</u></p> <div style="border: 1px solid black; padding: 2px; width: fit-content; margin: 5px 0;">set EWA station_address="AF:01:AC:78:1B:CC"</div>

## 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>Assigns 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><b>Example:</b></p> <pre>set EWA adapter_mode="100BaseT-HD"</pre> <div style="border: 1px solid red; padding: 10px; margin-top: 10px;"> <p>Please note that this parameter controls only the setting which the emulated network adapter reports to the guest operating system running on CHARON. It does not change any settings of the host adapter mapped to the virtual adapter! In case if the settings of the host adapter have to be changed please refer to the host Operating System User's Guide for details on how to do it.</p> <p>In case if "Auto" value is set CHARON collects the settings from the mapped host network adapter directly, assuming that all speeds above 100Mbps are represented as 100Mbps (maximum value). The resulting duplex setting corresponds to the duplex setting of the mapped host network adapter.</p> </div>	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 = "connection:Charon"
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
```

Please note that EIX0\_MODE environment variable controls only the setting which the emulated network adapter reports to the guest operating system running on CHARON. It does not change any settings of the host adapter mapped to the virtual adapter! In case if the settings of the host adapter have to be changed please refer to the host Operating System User's Guide for details on how to do it.

In case if "Auto" value is set CHARON collects the settings from the mapped host network adapter directly, assuming that all speeds above 100Mbps are represented as 100Mbps (maximum value). The resulting duplex setting corresponds to the duplex setting of the mapped host network adapter.

[Back to Table of Contents](#)

## Packet Port

The CHARON-specific "packet\_port" interface establishes a connection between an Ethernet adapter in the Windows 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 NDIS
```

"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. The leading keyword "connection:" is mandatory.</p> <p>Syntax:</p> <pre>set &lt;name&gt; interface="connection:&lt;adapter&gt;"</pre> <p><b>Example:</b></p> <pre>set NDIS interface="connection:Charon"</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><u>Example:</u></p> <pre>set NDIS 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><u>Example:</u></p> <pre>set NDIS 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><u>Example:</u></p> <pre>set NDIS 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 NDIS port_pending_tx_number=128</pre>

## suspend\_msg\_on\_mac\_change

<b>Parameter</b>	suspend_msg_on_mac_change
<b>Type</b>	Boolean
<b>Value</b>	<p>To avoid confusion arising from non critical errors during a MAC address change, logging is by default suppressed (default value is "true").</p> <p>To enable tracing during a MAC address change set this parameter to "false"</p> <p><b>Example:</b></p> <pre>set NDIS suspend_msg_on_mac_change=false</pre>

## legacy\_mode

<b>Parameter</b>	legacy_mode
<b>Type</b>	Boolean
<b>Value</b>	<p>If this parameter is set to "true" CHARON-AXP tries to use very first implementation of "packet_port" having certain restrictions such as inability to automatically change MAC address on given interface etc.</p> <p>This setting may be needed if CHARON-AXP runs in VMware environment for example.</p> <p>In other situations please do not change its default value ("false").</p> <p><b>Example:</b></p> <pre>set NDIS legacy_mode=true</pre> <div style="border: 1px solid black; padding: 10px; margin-top: 10px;"> <p>This parameter is required only if you do not want to allow promiscuous traffic for the VMware adapter/port. In this mode, CHARON-AXP uses legacy MAC address change logic, so use of this option is not desirable.</p> <p>Thus for ESXi networking configuration:</p> <ol style="list-style-type: none"> <li>1. Create the port with e1000 adapter;</li> <li>2. Enable the promiscuous traffic;</li> <li>3. do not use legacy_mode option in Charon's configuration file</li> </ol> </div>

## Example

```
load DE500BA/dec21x4x EWA
load packet_port/chnetwrk EWA0 interface="connection:Charon"
set EWA interface=EWA0
```

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](#)

# DEFPA PCI FDDI adapter

## Table of Contents

- General description
- Loading DEFPA PCI FDDI adapter
- Configuration parameters
  - host\_bus\_location
- Installation of DEFPA PCI FDDI adapter
- Mapping to host DEFPA PCI FDDI adapter

[Back to Table of Contents](#)

## General description

CHARON-AXP supports emulation of DEFPA PCI FDDI adapter via CHARON PCI Pass Through mode (using a specific CHARON PCI Pass Through driver).

CHARON PCI Pass Through mode enables connection between the virtual DEFPA PCI FDDI adapter and the physical DEFPA PCI FDDI adapter plugged into a hosting server PCI bus.

[Back to Table of Contents](#)

## Loading DEFPA PCI FDDI adapter

Syntax for loading DEFPA PCI FDDI adapter:

```
load defpa <name>
```

Example:

```
load defpa FDDI
```

By default DEFPA 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 DEFPA PCI FDDI adapter emulation has only one configuration parameter:

### host\_bus\_location

<b>Parameter</b>	host_bus_location								
<b>Type</b>	Text String								
<b>Value</b>	<p>Establish connection between virtual DEFPA PCI FDDI adapter and physical DEFPA PCI FDDI adapter installed on CHARON-AXP host (pass through mode)</p> <p>Syntax:</p> <pre>load defpa &lt;controller name&gt; host_bus_location="PCI bus X, device Y, function Z"</pre> <p>where:</p> <table border="1"> <thead> <tr> <th>Parameter</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>"PCI bus X"</td> <td>PCI bus number of the host DEFPA PCI FDDI adapter</td> </tr> <tr> <td>"device Y"</td> <td>PCI bus device number of the host DEFPA PCI FDDI adapter</td> </tr> <tr> <td>"function Z"</td> <td>The "function" parameter of the the host DEFPA PCI FDDI adapter</td> </tr> </tbody> </table> <p><b>Example:</b></p> <pre>load defpa FDDI host_bus_location="PCI bus 3, device 1, function 0"</pre>	Parameter	Description	"PCI bus X"	PCI bus number of the host DEFPA PCI FDDI adapter	"device Y"	PCI bus device number of the host DEFPA PCI FDDI adapter	"function Z"	The "function" parameter of the the host DEFPA PCI FDDI adapter
Parameter	Description								
"PCI bus X"	PCI bus number of the host DEFPA PCI FDDI adapter								
"device Y"	PCI bus device number of the host DEFPA PCI FDDI adapter								
"function Z"	The "function" parameter of the the host DEFPA PCI FDDI adapter								

[Back to Table of Contents](#)

## Installation of DEFPA PCI FDDI adapter

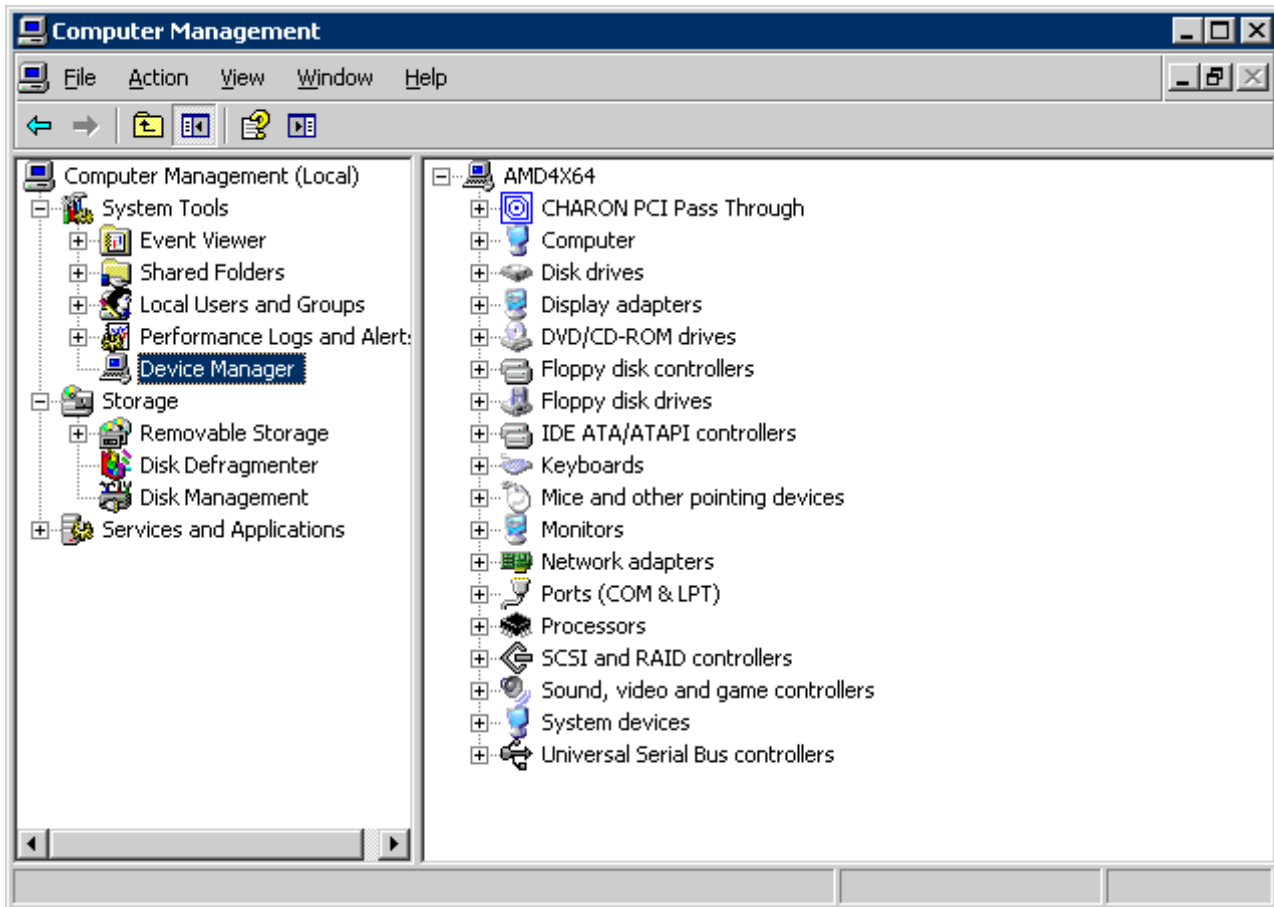
1. Install the DEFPA PCI FDDI adapter to some spare PCI slot of the host system
2. Boot a Windows operating system
3. Open "Computer Management"
4. Select "Device Manager"
5. In the right window select the desired physical FDDI adapter connected to the system, right-click the mouse button, and the corresponding menu will appear.
6. From the menu select "Update driver...". Windows will show "Hardware Upgrade Wizard"
7. Select "No, not this time", click "Next"
8. Select "Install from a list or specific location (Advanced)", click "Next"
9. Select "Don't search. I will choose the driver to install", click "Next"
10. Click "Have Disk...". Windows shows dialog "Install From Disk"
11. Instead of "A:\", click "Browse" and select path to the folder in which driver's INF file is located (typically "C:\Program Files\CHARON\Drivers\DEFPA\_X.X.0.XXXXX"), select "defpa\_ppt\_amd64.inf", and click "Open"
12. The "Hardware Upgrade Wizard" should have "CHARON DEFPA FDDI adapter". Select it, and click "Next"
13. There will be one or two more dialogs, but they are usual for device driver installation.
14. Reboot the host

Upon completion, a new device will appear in the device manager with the CHARON logo on it.

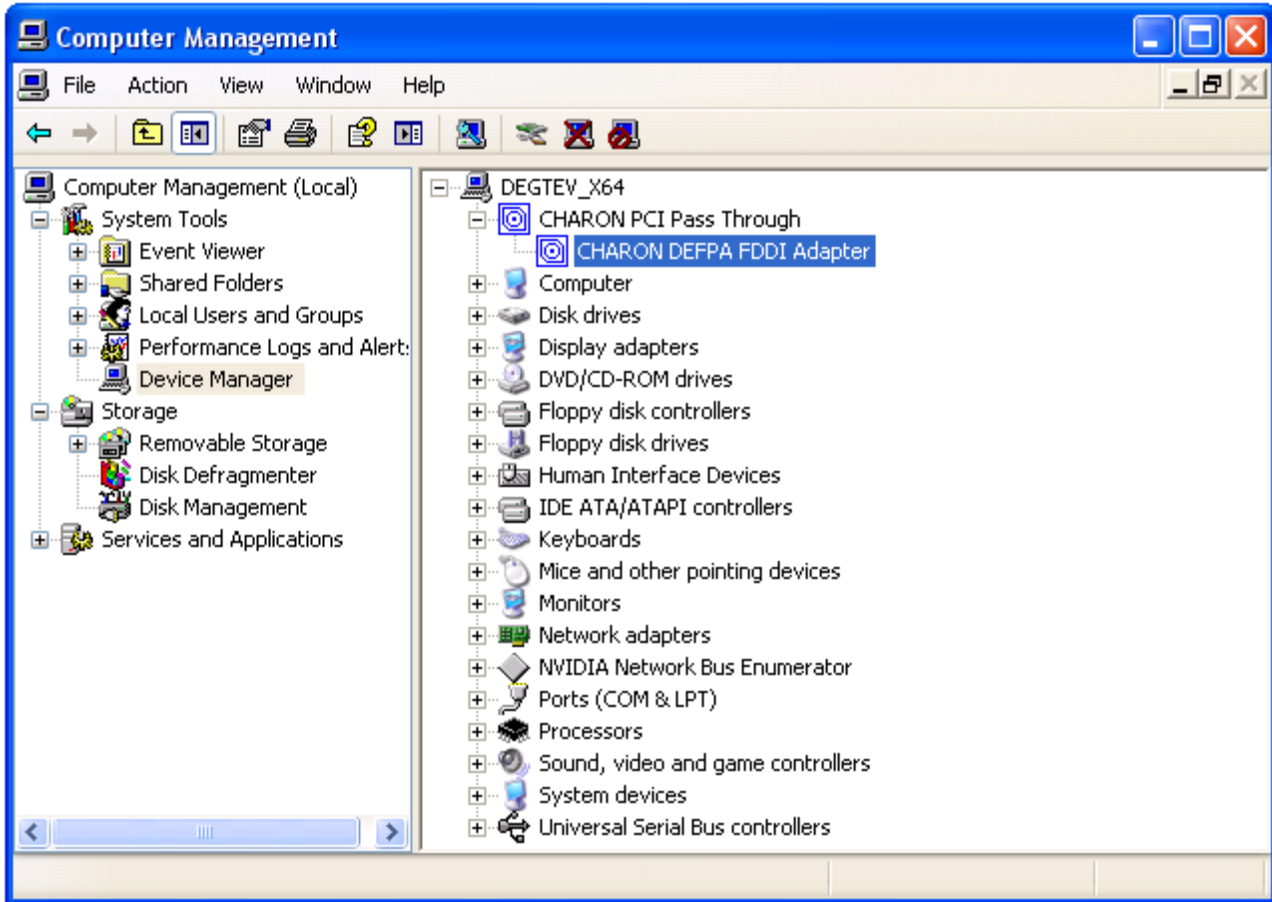
[Back to Table of Contents](#)

## Mapping to host DEFPA PCI FDDI adapter

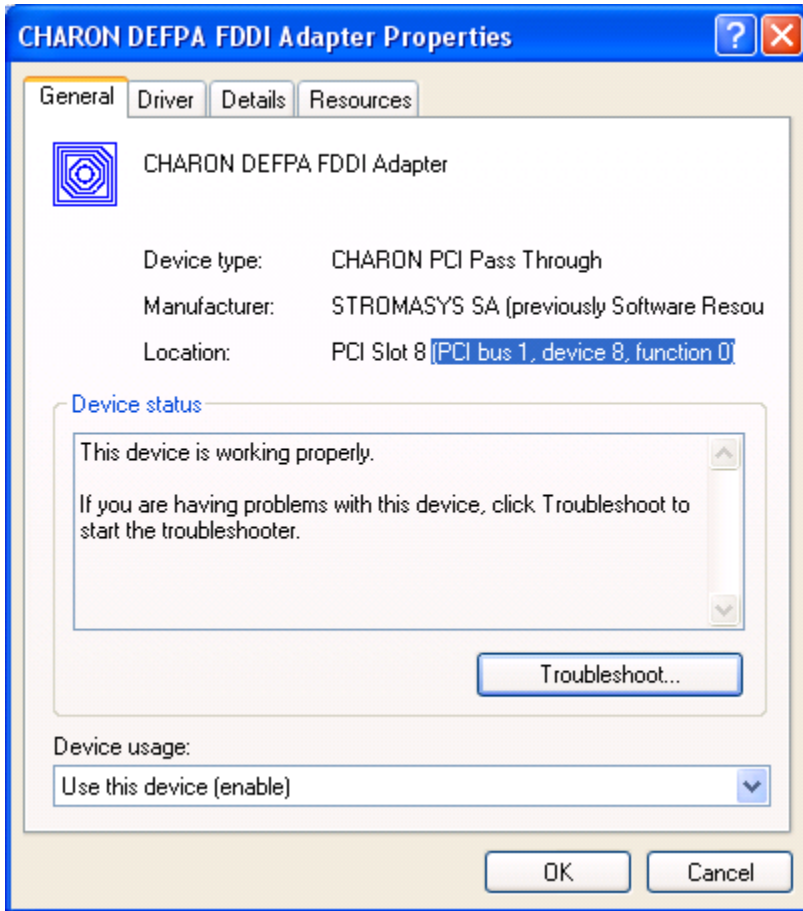
Open "Computer Management" application and select "Device Manager":



In the right panel select the installed DEFPFA PCI FDDI adapter:



Open its properties sheet by double-clicking on the selected adapter:



The "Location:" on the above picture gives X, Y, and Z for the "host\_bus\_location" parameter, for example:

```
set FDDI host_bus_location = "PCI bus 1, device 8, function 0"
```

**!** Non-US-EN installations of Windows may present "Location:" string in local language, but "host\_bus\_location" parameter requires English notation, so the words "PCI", "bus", "device", and "function" must be specified in English.

[Back to Table of Contents](#)



# PBXDA-xx series PCI serial adapters

## Table of Contents

- General description
- Loading PBXDA-xx PCI serial adapter
- Configuration parameters
  - host\_bus\_location
- Installation of DIGI AccelePort PCI serial adapter
- Mapping to host DIGI AccelePort PCI serial adapter

[Back to Table of Contents](#)

## General description

CHARON-AXP supports emulation of PBXDA-xx family PCI serial adapter based on DIGI AccelePort serial adapters via CHARON PCI Pass Through mode (using a specific CHARON PCI Pass Through driver).

CHARON PCI Pass Through mode enables connection between the virtual PBXDA-xx PCI serial adapter and the physical DIGI AccelePort PCI serial adapter plugged into a hosting server PCI bus.

**i** Note the model of DIGI AccelePort PCI serial adapter for emulation of a particular model of PBXDA-xx family PCI serial adapter:

DEC PBXDA-xx adapter	Name of the device to map to	Controller	Vendor ID	Device ID
PBXDA-BA	DIGI AccelePort 4r 920	ASIC PCI	114Fh	0026h
PBXDA-BB	DIGI AccelePort 8r 920	ASIC PCI	114Fh	0027h
PBXDA-AC	DIGI AccelePort Xem	ASIC PCI	114Fh	0004h
PBXDA-AC	DIGI AccelePort Xem	ASIC PCI	114Fh	0008h

[Back to Table of Contents](#)

## Loading PBXDA-xx PCI serial adapter

Syntax for loading PBXDA-xx PCI serial adapter:

```
load digi <name>
```

**Example:**

```
load digi SERIAL_A
```

By default PBXDA-xx 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 PBXDA-xx PCI serial adapter emulation has only one configuration parameter:

### host\_bus\_location

<b>Parameter</b>	host_bus_location								
<b>Type</b>	Text String								
<b>Value</b>	<p>Establish connection between virtual PBXDA-xx PCI serial adapter and physical DIGI AccelePort PCI serial adapter installed on CHARON-AXP host (pass through mode)</p> <p>Syntax:</p> <pre>load digi &lt;controller name&gt; host_bus_location="PCI bus X, device Y, function Z"</pre> <p>where:</p> <table border="1"> <thead> <tr> <th>Parameter</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>"PCI bus X"</td> <td>PCI bus number of the host DIGI AccelePort PCI serial adapter</td> </tr> <tr> <td>"device Y"</td> <td>PCI bus device number of the host DIGI AccelePort PCI serial adapter</td> </tr> <tr> <td>"function Z"</td> <td>The "function" parameter of the the host DIGI AccelePort PCI serial adapter</td> </tr> </tbody> </table> <p><u>Example:</u></p> <pre>load digi SERIAL_A host_bus_location="PCI bus 3, device 1, function 0"</pre>	Parameter	Description	"PCI bus X"	PCI bus number of the host DIGI AccelePort PCI serial adapter	"device Y"	PCI bus device number of the host DIGI AccelePort PCI serial adapter	"function Z"	The "function" parameter of the the host DIGI AccelePort PCI serial adapter
Parameter	Description								
"PCI bus X"	PCI bus number of the host DIGI AccelePort PCI serial adapter								
"device Y"	PCI bus device number of the host DIGI AccelePort PCI serial adapter								
"function Z"	The "function" parameter of the the host DIGI AccelePort PCI serial adapter								

[Back to Table of Contents](#)

## Installation of DIGI AccelePort PCI serial adapter

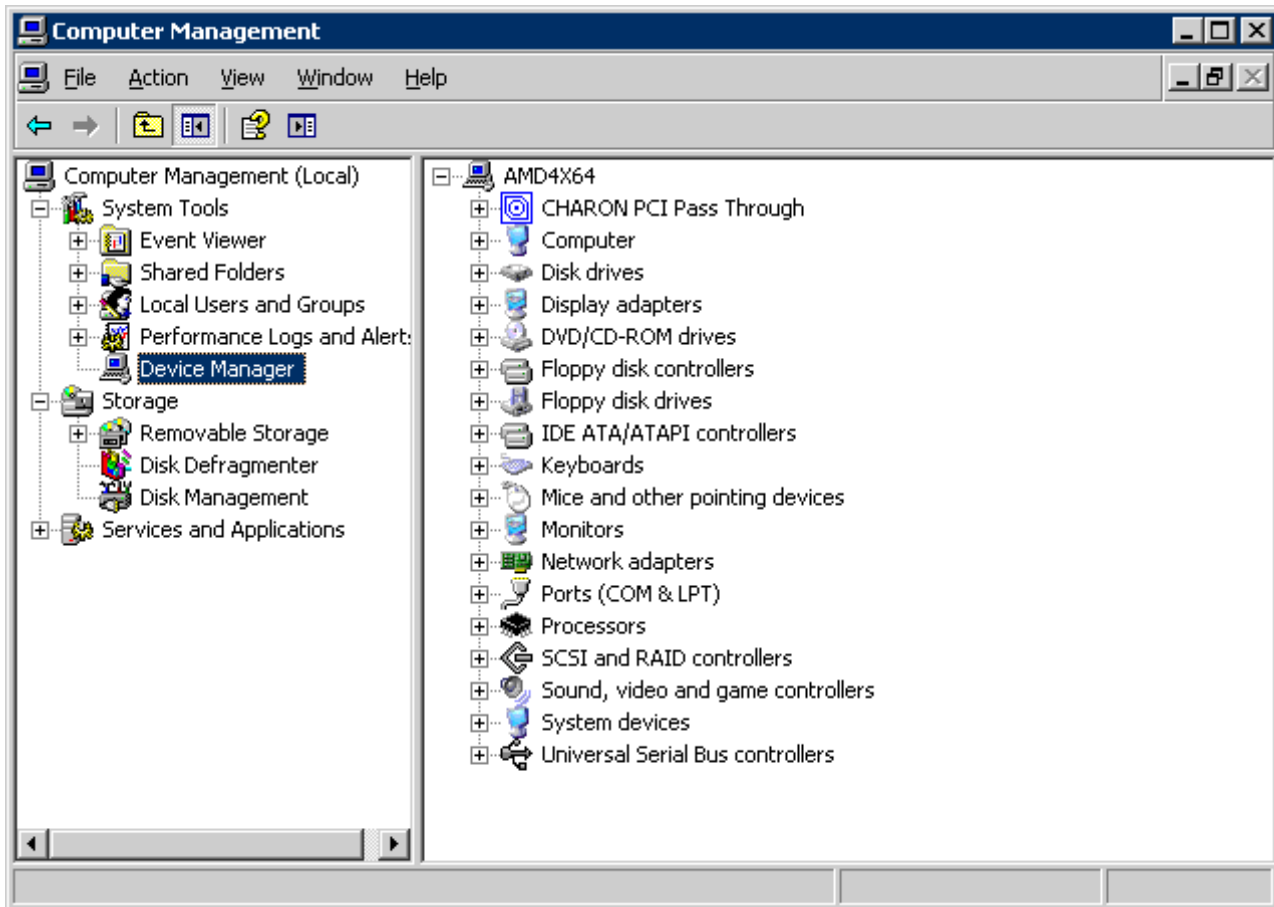
1. Install the DIGI AccelePort PCI serial adapter to some spare PCI slot of the host system
2. Boot a Windows operating system
3. Open "Computer Management"
4. Select "Device Manager"
5. In the right window select the desired physical serial adapter connected to the system, right-click the mouse button, and the corresponding menu will appear.
6. From the menu select "Update driver...". Windows will show "Hardware Upgrade Wizard"
7. Select "No, not this time", click "Next"
8. Select "Install from a list or specific location (Advanced)", click "Next"
9. Select "Don't search. I will choose the driver to install", click "Next"
10. Click "Have Disk...". Windows shows dialog "Install From Disk"
11. Instead of "A:\", click "Browse" and select path to the folder in which driver's INF file is located (typically "C:\Program Files\CHARON\Drivers\DIGI\_X.X.0.XXXXX"), select "digi\_ppt\_amd64.inf", and click "Open"
12. The "Hardware Upgrade Wizard" should have "CHARON DIGI adapter". Select it, and click "Next"
13. There will be one or two more dialogs, but they are usual for device driver installation.
14. Reboot the host

Upon completion, a new device will appear in the device manager with the CHARON logo on it.

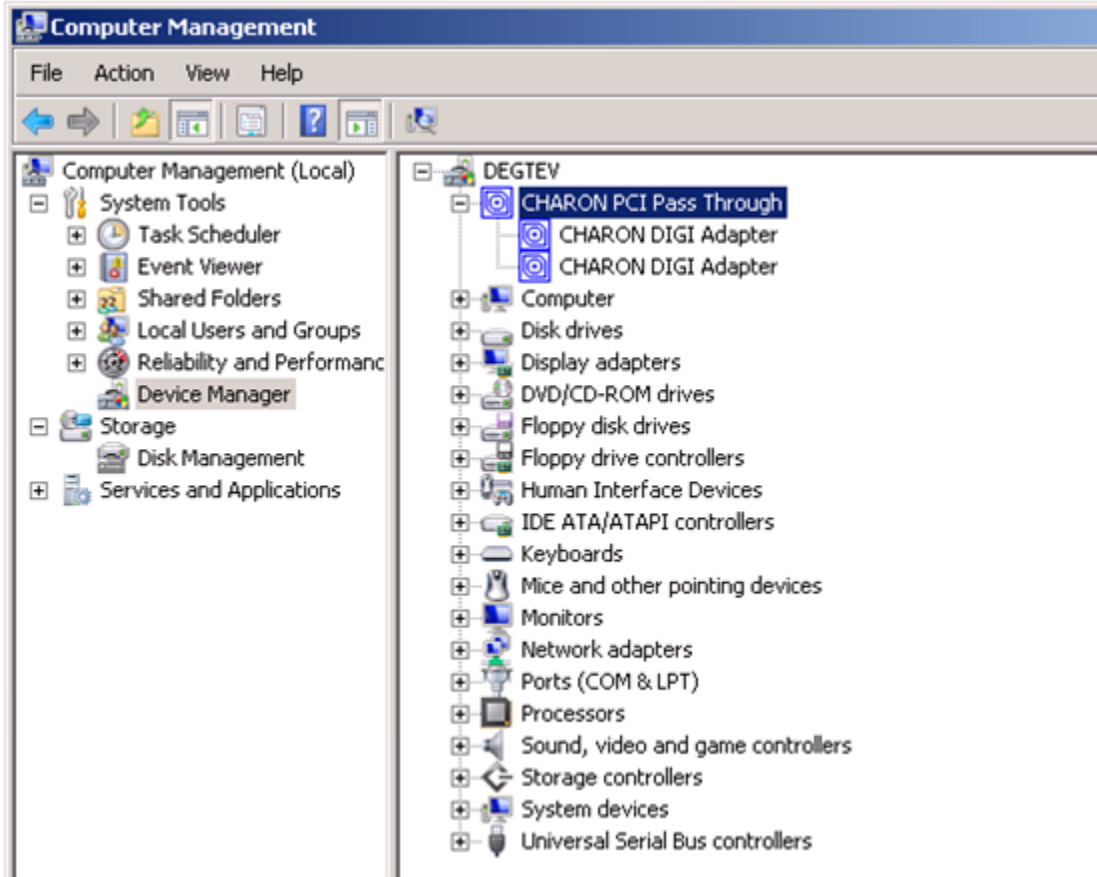
[Back to Table of Contents](#)

## Mapping to host DIGI AccelePort PCI serial adapter

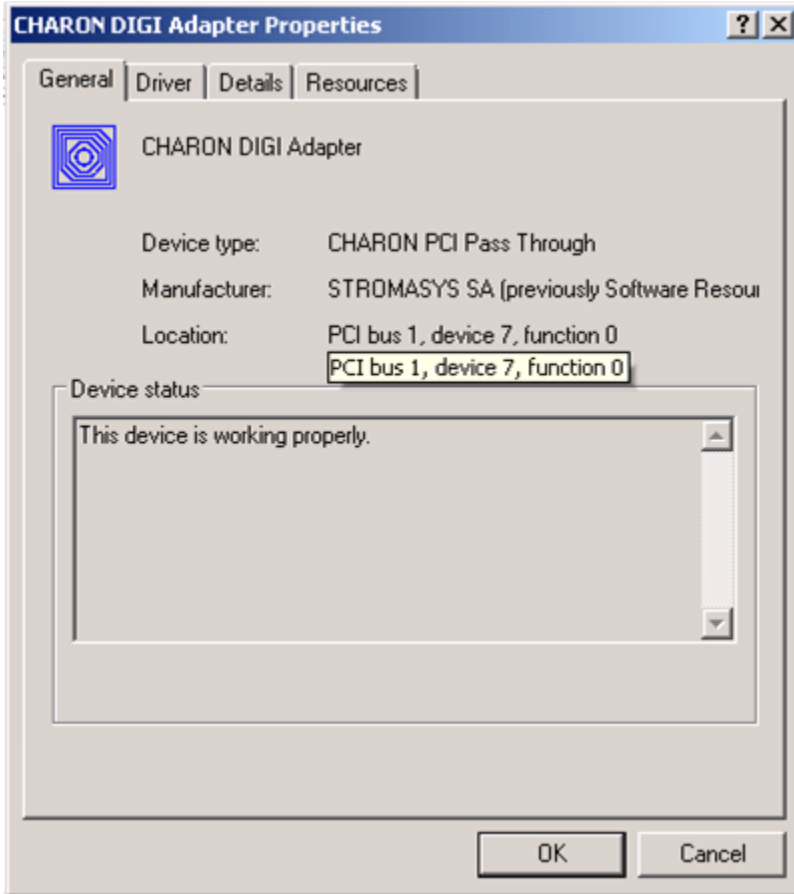
Open "Computer Management" application and select "Device Manager":



In the right panel select proper physical DIGI adapter:



Open its properties sheet by double-clicking on the selected adapter:



The "Location:" on the above picture gives X, Y, and Z for the "host\_bus\_location" parameter, for example:

```
set PBXDA host_bus_location = "PCI bus 1, device 7, function 0"
```

**!** Non-US-EN installations of Windows may present "Location:" string in local language, but "host\_bus\_location" parameter requires English notation, so the words "PCI", "bus", "device", and "function" must be specified in English.

[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) 1999-2014 STROMASYS
# All rights reserved.
#
# The software contained on this media is proprietary to and embodies
# the confidential technology of STROMASYS. Possession, use, duplication,
# or dissemination of the software and media is authorized only pursuant
# to a valid written license from STROMASYS.
#
#-----
#
# Sample configuration file for AlphaServer 800 machines.
#
#-----

set session hw_model = AlphaServer_800

#-----
#
# Select name of the instance to differentiate it among other instances
# running on the same host.
#
#-----

#set session configuration_name = AlphaServer_800

#-----
#
# Disable rotating LOG and enable single file LOG. Select either appending or
# overwriting it on every instance start. Then specify desired log file name
# and path to it.
#
#-----

#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 connection for the console serial line OPA0.
#
#-----

#load physical_serial_line OPA0 line="COM1:"
#load virtual_serial_line OPA0 port=10003
#load virtual_serial_line OPA0 port=10003 application="opa0.ht"
load virtual_serial_line OPA0 port=10003 application="putty -load OPA0"
#load virtual_serial_line OPA0 port=10003 application="c:\kea\user\opa0.ktc"

=====
#
# Select connection for the serial line TTA0.
#
#-----

```



```

#load physical_serial_line TTA0 line="COM2:"
#load virtual_serial_line TTA0 port=10000
#load virtual_serial_line TTA0 port=10000 application="tta0.ht"
#load virtual_serial_line TTA0 port=10000 application="putty -load TTA0"
#load virtual_serial_line TTA0 port=10000 application="c:\kea\user\tta0.ktc"

#=====
#
# 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="(disabled)"
#load packet_port/chnetwrk EWA0 interface="connection:<connection-name>"

#=====
#
# 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 host's disk drive.
#
#-----

#set PKA container[100]="\\.\PhysicalDrive0"
#set PKA container[100]="\\.\PhysicalDrive<N>"

#=====
#
# Uncomment to connect the emulator's GKA200 to an unknown SCSI device.
#
#-----

#set PKA container[200]="\\.\ScsiN:X:Y:Z"

#=====
#
# Uncomment to connect the emulator's DKA300 to host's CD/DVD-ROM drive.
#
#-----

#set PKA container[300]="\\.\CdRom0"
#set PKA container[300]="\\.\CdRom<N>"

```

```

=====
#
# Uncomment to connect the emulator's DKA400 to .ISO file (CD/DVD-ROM image).
#
#-----

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

=====
#
# Uncomment to connect the emulator's MKA500 to host's SCSI tape drive.
#
#-----

#set PKA container[500]="\\.\Tape0"
#set PKA container[500]="\\.\Tape<N>"

=====
#
# Uncomment to connect the emulator's MKA600 to .VTAPE file (tape image).
#
#-----

#set PKA container[600] = "<file-name>.vtape"

=====
#
# Uncomment to connect the emulator's DKA600 to host's 3.5" FDD. Uncomment
# the next line for the FDD to appear as DEC RX23 (otherwise it appears as
# DEC RX26).
#
# Either "A:" or "B:" may be used in container specification.
#
#-----

#set PKA container[600]="\\.\A:"
#set PKA media_type[600]="RX23"

=====
#
# 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 host's disk drive.
#
#-----

#set PKB container[100]="\\.\PhysicalDrive0"
#set PKB container[100]="\\.\PhysicalDrive<N>"

=====
#
# Uncomment to connect the emulator's GKB200 to an unknown SCSI device.
#
#-----

#set PKB container[200]="\\.\ScsiN:X:Y:Z"

```

```

=====
#
# Uncomment to connect the emulator's DKB300 to host's CD/DVD-ROM drive.
#
#-----

#set PKB container[300]="\\.\CdRom0"
#set PKB container[300]="\\.\CdRom<N>"

=====
#
# Uncomment to connect the emulator's DKB400 to .ISO file (CD/DVD-ROM image).
#
#-----

#set PKB container[400] = "<file-name>.iso"

=====
#
# Uncomment to connect the emulator's MKB500 to host's SCSI tape drive.
#
#-----

#set PKB container[500]="\\.\Tape0"
#set PKB container[500]="\\.\Tape<N>"

=====
#
# Uncomment to connect the emulator's MKB600 to .VTAPE file (tape image).
#
#-----

#set PKB container[600] = "<file-name>.vtape"

=====
#
# Uncomment to connect the emulator's DKB600 to host's 3.5" FDD. Uncomment
# the next line for the FDD to appear as DEC RX23 (otherwise it appears as
# DEC RX26).
#
# Either "A:" or "B:" may be used in container specification.
#
#-----

#set PKB container[600]="\\.\A:"
#set PKB media_type[600]="RX23"

=====
#
# 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 host's disk drive.
#
#-----

#set FGA container[100] = "\\.\PhysicalDrive<N>"

```

```
#####  
#  
# 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 = "PCI bus X, device Y, function Z"  
#set FGB host_bus_location = "PCI bus A, device B, function C"  
  
# this is the end of the configuration file #####
```

[Back to Top](#)

## HP AlphaServer 4000 configuration file

```

#
# Copyright (C) 1999-2014 STROMASYS
# All rights reserved.
#
# The software contained on this media is proprietary to and embodies
# the confidential technology of STROMASYS. Possession, use, duplication,
# or dissemination of the software and media is authorized only pursuant
# to a valid written license from STROMASYS.
#
#-----
#
# Sample configuration file for AlphaServer 4000 machines.
#
#-----

set session hw_model = AlphaServer_4000

#-----
#
# Select name of the instance to differentiate it among other instances
# running on the same host.
#
#-----

#set session configuration_name = AlphaServer_4000

#-----
#
# Disable rotating LOG and enable single file LOG. Select either appending or
# overwriting it on every instance start. Then specify desired log file name
# and path to it.
#
#-----

#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 connection for the console serial line OPA0.
#
#-----

#load physical_serial_line OPA0 line="COM1:"
#load virtual_serial_line OPA0 port=10003
#load virtual_serial_line OPA0 port=10003 application="opa0.ht"
load virtual_serial_line OPA0 port=10003 application="putty -load OPA0"
#load virtual_serial_line OPA0 port=10003 application="c:\kea\user\opa0.ktc"

=====
#
# 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="(disabled)"
#load packet_port/chnetwrk EWA0 interface="connection:<connection-name>"

#=====
#
# Load another optional DE500BA PCI Ethernet Adapter (EWB).
#
#-----

#load DE500BA/dec21x4x EWB interface=EWB0
#load packet_port/chnetwrk EWB0 interface="(disabled)"
#load packet_port/chnetwrk EWB0 interface="connection:<connection-name>"

#=====
#
# Load another optional DE500BA PCI Ethernet Adapter (EWC).
#
#-----

#load DE500BA/dec21x4x EWC interface=EWC0
#load packet_port/chnetwrk EWC0 interface="(disabled)"
#load packet_port/chnetwrk EWC0 interface="connection:<connection-name>"

#=====
#
# 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 host's disk drive.
#
#-----

#set PKA container[100]="\\.\PhysicalDrive0"
#set PKA container[100]="\\.\PhysicalDrive<N>"

#=====
#
# Uncomment to connect the emulator's GKA200 to an unknown SCSI device.
#
#-----

#set PKA container[200]="\\.\ScsiN:X:Y:Z"

#=====
#
# Uncomment to connect the emulator's DKA300 to host's CD/DVD-ROM drive.
#
#-----

#set PKA container[300]="\\.\CdRom0"
#set PKA container[300]="\\.\CdRom<N>"

```

```

=====
#
# Uncomment to connect the emulator's DKA400 to .ISO file (CD/DVD-ROM image).
#
#-----

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

=====
#
# Uncomment to connect the emulator's MKA500 to host's SCSI tape drive.
#
#-----

#set PKA container[500]="\\.\Tape0"
#set PKA container[500]="\\.\Tape<N>"

=====
#
# Uncomment to connect the emulator's MKA600 to .VTAPE file (tape image).
#
#-----

#set PKA container[600] = "<file-name>.vtape"

=====
#
# Uncomment to connect the emulator's DKA600 to host's 3.5" FDD. Uncomment
# the next line for the FDD to appear as DEC RX23 (otherwise it appears as
# DEC RX26).
#
# Either "A:" or "B:" may be used in container specification.
#
#-----

#set PKA container[600]="\\.\A:"
#set PKA media_type[600]="RX23"

=====
#
# 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 host's disk drive.
#
#-----

#set FGA container[100] = "\\.\PhysicalDrive<N>"

=====
#
# 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 = "PCI bus X, device Y, function Z"  
#set FGB host_bus_location = "PCI bus A, device B, function C"  
  
# this is the end of the configuration file #####
```

[Back to Top](#)

## HP AlphaServer DS20 configuration file

```

#
# Copyright (C) 1999-2014 STROMASYS
# All rights reserved.
#
# The software contained on this media is proprietary to and embodies
# the confidential technology of STROMASYS. Possession, use, duplication,
# or dissemination of the software and media is authorized only pursuant
# to a valid written license from STROMASYS.
#
#-----
#
# Sample configuration file for AlphaServer DS20 machines.
#
#-----

set session hw_model = AlphaServer_DS20

#-----
#
# Select name of the instance to differentiate it among other instances
# running on the same host.
#
#-----

#set session configuration_name = AlphaServer_DS20

#-----
#
# Disable rotating LOG and enable single file LOG. Select either appending or
# overwriting it on every instance start. Then specify desired log file name
# and path to it.
#
#-----

#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 connection for the console serial line OPA0.
#
#-----

#load physical_serial_line OPA0 line="COM1:"
#load virtual_serial_line OPA0 port=10003
#load virtual_serial_line OPA0 port=10003 application="opa0.ht"
load virtual_serial_line OPA0 port=10003 application="putty -load OPA0"
#load virtual_serial_line OPA0 port=10003 application="c:\kea\user\opa0.ktc"

=====
#
# 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="\.\CdRom0"

#-----
#
# Load optional DE500BA PCI Ethernet Adapter (EWA).
#
#-----

#load DE500BA/dec21x4x EWA interface=EWA0
#load packet_port/chnetwrk EWA0 interface="(disabled)"
#load packet_port/chnetwrk EWA0 interface="connection:<connection-name>"

#-----
#
# Load another optional DE500BA PCI Ethernet Adapter (EWB).
#
#-----

#load DE500BA/dec21x4x EWB interface=EWB0
#load packet_port/chnetwrk EWB0 interface="(disabled)"
#load packet_port/chnetwrk EWB0 interface="connection:<connection-name>"

#-----
#
# Load another optional DE500BA PCI Ethernet Adapter (EWC).
#
#-----

#load DE500BA/dec21x4x EWC interface=EWC0
#load packet_port/chnetwrk EWC0 interface="(disabled)"
#load packet_port/chnetwrk EWC0 interface="connection:<connection-name>"

#-----
#
# 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 host's disk drive.
#
#-----

#set PKA container[100] = "\.\PhysicalDrive0"
#set PKA container[100] = "\.\PhysicalDrive<N>"

#-----
#
# Uncomment to connect the emulator's GKA200 to an unknown SCSI device.
#
#-----

#set PKA container[200] = "\.\ScsiN:X:Y:Z"

```

```

=====
#
# Uncomment to connect the emulator's DKA300 to host's CD/DVD-ROM drive.
#
#-----

#set PKA container[300] = "\\.\CdRom0"
#set PKA container[300] = "\\.\CdRom<N>"

=====
#
# Uncomment to connect the emulator's DKA400 to .ISO file (CD/DVD-ROM image).
#
#-----

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

=====
#
# Uncomment to connect the emulator's MKA500 to host's SCSI tape drive.
#
#-----

#set PKA container[500] = "\\.\Tape0"
#set PKA container[500] = "\\.\Tape<N>"

=====
#
# Uncomment to connect the emulator's MKA600 to .VTAPE file (tape image).
#
#-----

#set PKA container[600] = "<file-name>.vtape"

=====
#
# Uncomment to connect the emulator's DKA600 to host's 3.5" FDD. Uncomment
# the next line for the FDD to appear as DEC RX23 (otherwise it appears as
# DEC RX26).
#
# Either "A:" or "B:" may be used in container specification.
#
#-----

#set PKA container[600] = "\\.\A:"
#set PKA media_type[600] = "RX23"

=====
#
# 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 host's disk drive.
#
#-----

#set PKB container[100] = "\\.\PhysicalDrive0"
#set PKB container[100] = "\\.\PhysicalDrive<N>"

=====
#
# Uncomment to connect the emulator's GKB200 to an unknown SCSI device.
#
#-----

#set PKB container[200] = "\\.\ScsiN:X:Y:Z"

```

```

=====
#
# Uncomment to connect the emulator's DKB300 to host's CD/DVD-ROM drive.
#
#-----

#set PKB container[300] = "\\.\CdRom0"
#set PKB container[300] = "\\.\CdRom<N>"

=====
#
# Uncomment to connect the emulator's DKB400 to .ISO file (CD/DVD-ROM image).
#
#-----

#set PKB container[400] = "<file-name>.iso"

=====
#
# Uncomment to connect the emulator's MKB500 to host's SCSI tape drive.
#
#-----

#set PKB container[500] = "\\.\Tape0"
#set PKB container[500] = "\\.\Tape<N>"

=====
#
# Uncomment to connect the emulator's MKB600 to .VTAPE file (tape image).
#
#-----

#set PKB container[600] = "<file-name>.vtape"

=====
#
# Uncomment to connect the emulator's DKB600 to host's 3.5" FDD. Uncomment
# the next line for the FDD to appear as DEC RX23 (otherwise it appears as
# DEC RX26).
#
# Either "A:" or "B:" may be used in container specification.
#
#-----

#set PKB container[600] = "\\.\A:"
#set PKB media_type[600] = "RX23"

=====
#
# 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 host's disk drive.
#
#-----

#set FGA container[100] = "\\.\PhysicalDrive<N>"

```

```
#####  
#  
# 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 = "PCI bus X, device Y, function Z"  
#set FGB host_bus_location = "PCI bus A, device B, function C"  
  
# this is the end of the configuration file #####
```

[Back to Top](#)

## HP AlphaServer ES40 configuration file

```

#
# Copyright (C) 1999-2014 STROMASYS
# All rights reserved.
#
# The software contained on this media is proprietary to and embodies
# the confidential technology of STROMASYS. Possession, use, duplication,
# or dissemination of the software and media is authorized only pursuant
# to a valid written license from STROMASYS.
#
#-----
#
# Sample configuration file for AlphaServer ES40 machines.
#
#-----

set session hw_model = AlphaServer_ES40

#-----
#
# Select name of the instance to differentiate it among other instances
# running on the same host.
#
#-----

set session configuration_name = My_AlphaServer_ES40

#-----
#
# Disable rotating LOG and enable single file LOG. Select either appending or
# overwriting it on every instance start. Then specify desired log file name
# and path to it.
#
#-----

#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 connection for the console serial line OPA0.
#
#-----

#load physical_serial_line OPA0 line="COM1:"
#load virtual_serial_line OPA0 port=10003
#load virtual_serial_line OPA0 port=10003 application="opa0.ht"
load virtual_serial_line OPA0 port=10003 application="putty -load OPA0"
#load virtual_serial_line OPA0 port=10003 application="c:\kea\user\opa0.ktc"

```

```

=====
#
# 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="\\.\CdRom0"

=====
#
# Load optional DE500BA PCI Ethernet Adapter (EWA).
#
#-----

load DE500BA/dec21x4x EWA interface=EWA0
#load packet_port/chnetwrk EWA0 interface="(disabled)"
load packet_port/chnetwrk EWA0 interface="connection:Charon"

=====
#
# Load another optional DE500BA PCI Ethernet Adapter (EWB).
#
#-----

#load DE500BA/dec21x4x EWB interface=EWB0
#load packet_port/chnetwrk EWB0 interface="(disabled)"
#load packet_port/chnetwrk EWB0 interface="connection:<connection-name>"

=====
#
# Load another optional DE500BA PCI Ethernet Adapter (EWC).
#
#-----

#load DE500BA/dec21x4x EWC interface=EWC0
#load packet_port/chnetwrk EWC0 interface="(disabled)"
#load packet_port/chnetwrk EWC0 interface="connection:<connection-name>"

=====
#
# 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] = "C:\Dev\test\performancecomparison-axp.vdisk"
set PKA container[100] = "C:\Dev\test\RZ22.vdisk"
set PKA container[200] = "C:\Dev\test\RZ23.vdisk"

=====
#
# Uncomment to connect the emulator's DKA100 to host's disk drive.
#
#-----

#set PKA container[100]="\\.\PhysicalDrive0"
#set PKA container[100]="\\.\PhysicalDrive<N>"

```

```

=====
#
# Uncomment to connect the emulator's GKA200 to an unknown SCSI device.
#
#-----

#set PKA container[200]="\\.\ScsiN:X:Y:Z"

=====
#
# Uncomment to connect the emulator's DKA300 to host's CD/DVD-ROM drive.
#
#-----

#set PKA container[300]="\\.\CdRom0"
#set PKA container[300]="\\.\CdRom<N>"

=====
#
# Uncomment to connect the emulator's DKA400 to .ISO file (CD/DVD-ROM image).
#
#-----

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

=====
#
# Uncomment to connect the emulator's MKA500 to host's SCSI tape drive.
#
#-----

#set PKA container[500]="\\.\Tape0"
#set PKA container[500]="\\.\Tape<N>"

=====
#
# Uncomment to connect the emulator's MKA600 to .VTAPE file (tape image).
#
#-----

set PKA container[600] = "C:\Dev\test\my_tape.vtape"

=====
#
# Uncomment to connect the emulator's DKA600 to host's 3.5" FDD. Uncomment
# the next line for the FDD to appear as DEC RX23 (otherwise it appears as
# DEC RX26).
#
# Either "A:" or "B:" may be used in container specification.
#
#-----

#set PKA container[600]="\\.\A:"
#set PKA media_type[600]="RX23"

=====
#
# 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] = "C:\Dev\test\RZ24.vdisk"

```

```
#####  
#  
# Uncomment to connect the emulator's $1$DGA100 to host's disk drive.  
#  
#####  
#set FGA container[100] = "\\.\PhysicalDrive<N>"  
  
#####  
#  
# 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 = "PCI bus X, device Y, function Z"  
#set FGB host_bus_location = "PCI bus A, device B, function C"  
  
# this is the end of the configuration file #####
```

[Back to Top](#)

## HP AlphaServer GS80 configuration file

```

#
# Copyright (C) 1999-2014 STROMASYS
# All rights reserved.
#
# The software contained on this media is proprietary to and embodies
# the confidential technology of STROMASYS. Possession, use, duplication,
# or dissemination of the software and media is authorized only pursuant
# to a valid written license from STROMASYS.
#
#-----
#
# Sample configuration file for AlphaServer GS80 machines.
#
#-----

set session hw_model = AlphaServer_GS80

#-----
#
# Select name of the instance to differentiate it among other instances
# running on the same host.
#
#-----

#set session configuration_name = AlphaServer_GS80

#-----
#
# Disable rotating LOG and enable single file LOG. Select either appending or
# overwriting it on every instance start. Then specify desired log file name
# and path to it.
#
#-----

#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 connection for the console serial line OPA0.
#
#-----

#load physical_serial_line OPA0 line="COM1:"
#load virtual_serial_line OPA0 port=10003
#load virtual_serial_line OPA0 port=10003 application="opa0.ht"
load virtual_serial_line OPA0 port=10003 application="putty -load OPA0"
#load virtual_serial_line OPA0 port=10003 application="c:\kea\user\opa0.ktc"

```

```

=====
#
# 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="\\.\CdRom0"

=====
#
# Load optional DE500BA PCI Ethernet Adapter (EWA).
#
#-----

#load DE500BA/dec21x4x EWA interface=EWA0
#load packet_port/chnetwrk EWA0 interface="(disabled)"
#load packet_port/chnetwrk EWA0 interface="connection:<connection-name>"

=====
#
# Load another optional DE500BA PCI Ethernet Adapter (EWB).
#
#-----

#load DE500BA/dec21x4x EWB interface=EWB0
#load packet_port/chnetwrk EWB0 interface="(disabled)"
#load packet_port/chnetwrk EWB0 interface="connection:<connection-name>"

=====
#
# Load another optional DE500BA PCI Ethernet Adapter (EWC).
#
#-----

#load DE500BA/dec21x4x EWC interface=EWC0
#load packet_port/chnetwrk EWC0 interface="(disabled)"
#load packet_port/chnetwrk EWC0 interface="connection:<connection-name>"

=====
#
# 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 host's disk drive.
#
#-----

#set PKA container[100]="\\.\PhysicalDrive0"
#set PKA container[100]="\\.\PhysicalDrive<N>"

=====
#
# Uncomment to connect the emulator's GKA200 to an unknown SCSI device.
#
#-----

```

```

#set PKA container[200]="\\.\ScsiN:X:Y:Z"

#-----
#
# Uncomment to connect the emulator's DKA300 to host's CD/DVD-ROM drive.
#
#-----

#set PKA container[300]="\\.\CdRom0"
#set PKA container[300]="\\.\CdRom<N>"

#-----
#
# Uncomment to connect the emulator's DKA400 to .ISO file (CD/DVD-ROM image).
#
#-----

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

#-----
#
# Uncomment to connect the emulator's MKA500 to host's SCSI tape drive.
#
#-----

#set PKA container[500]="\\.\Tape0"
#set PKA container[500]="\\.\Tape<N>"

#-----
#
# Uncomment to connect the emulator's MKA600 to .VTAPE file (tape image).
#
#-----

#set PKA container[600] = "<file-name>.vtape"

#-----
#
# Uncomment to connect the emulator's DKA600 to host's 3.5" FDD. Uncomment
# the next line for the FDD to appear as DEC RX23 (otherwise it appears as
# DEC RX26).
#
# Either "A:" or "B:" may be used in container specification.
#
#-----

#set PKA container[600]="\\.\A:"
#set PKA media_type[600]="RX23"

#-----
#
# 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 host's disk drive.
#
#-----

#set FGA container[100] = "\\.\PhysicalDrive<N>"

```



```
#####  
#  
# 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 = "PCI bus X, device Y, function Z"  
#set FGB host_bus_location = "PCI bus A, device B, function C"  
  
# this is the end of the configuration file #####
```

[Back to Top](#)

# CHARON-AXP for Windows deinstallation

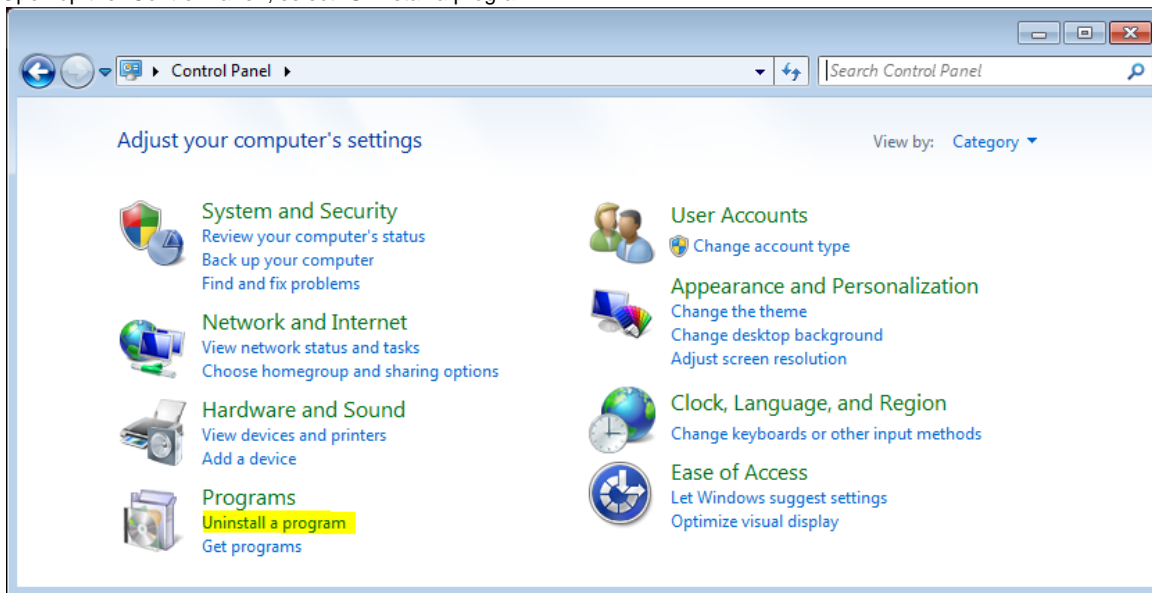
## Deinstallation procedure

Perform a clean shutdown of the running guests, stop all running CHARON-AXP instances and remove the corresponding CHARON-AXP services for the emulated HP Alpha models to be removed.

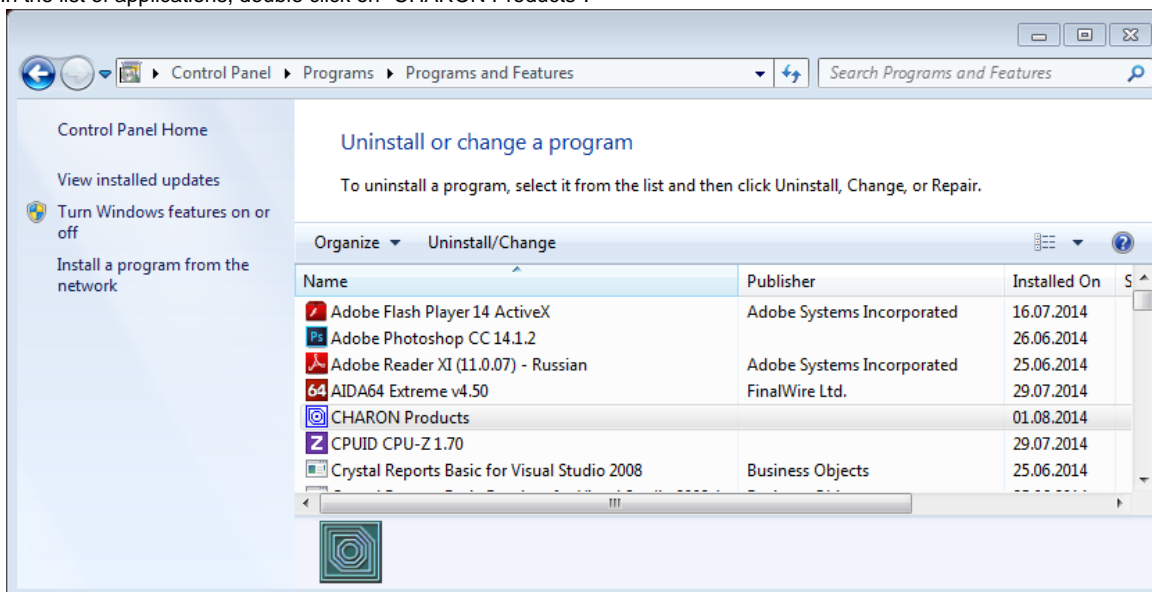
In case of a complete deinstallation:

- Remove all CHARON-AXP services.
- Using the "Network Control Center" utility, release all the network interfaces back to the CHARON host.
- Remove any specific CHARON drivers installed manually (if any).

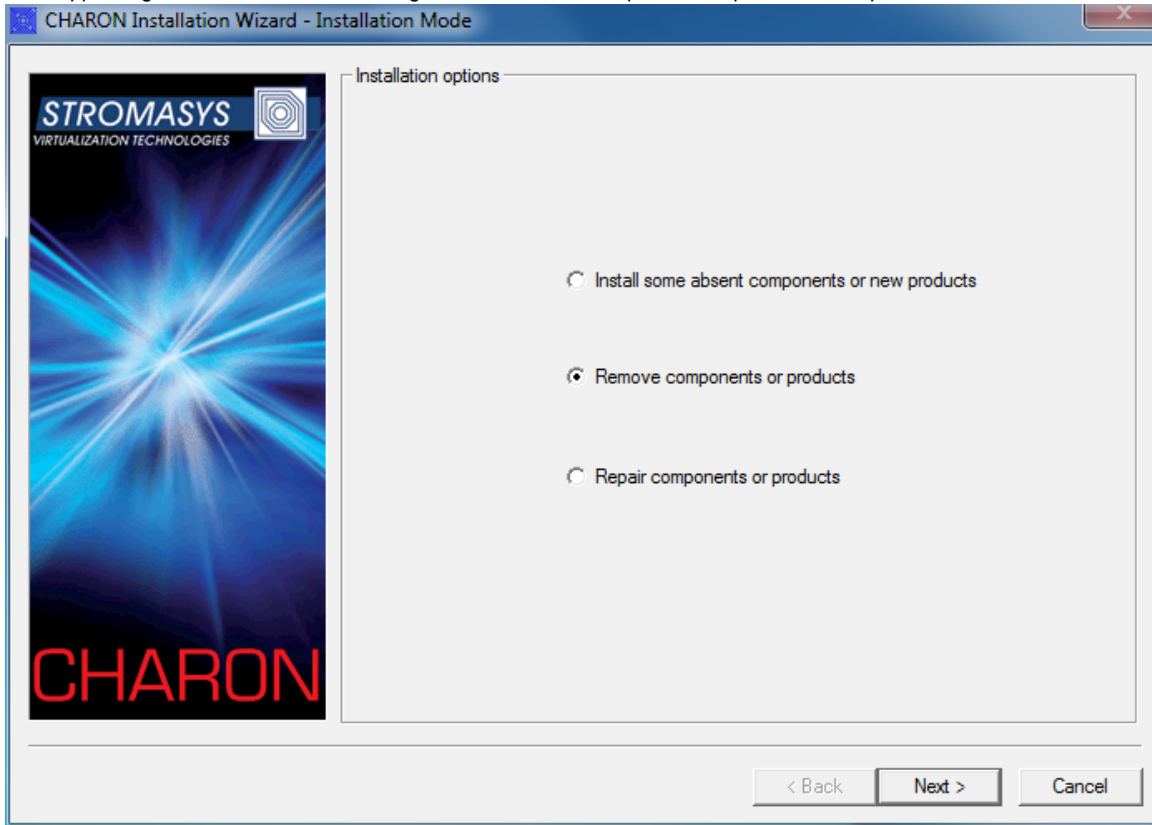
Open up the "Control Panel", select "Uninstall a program":



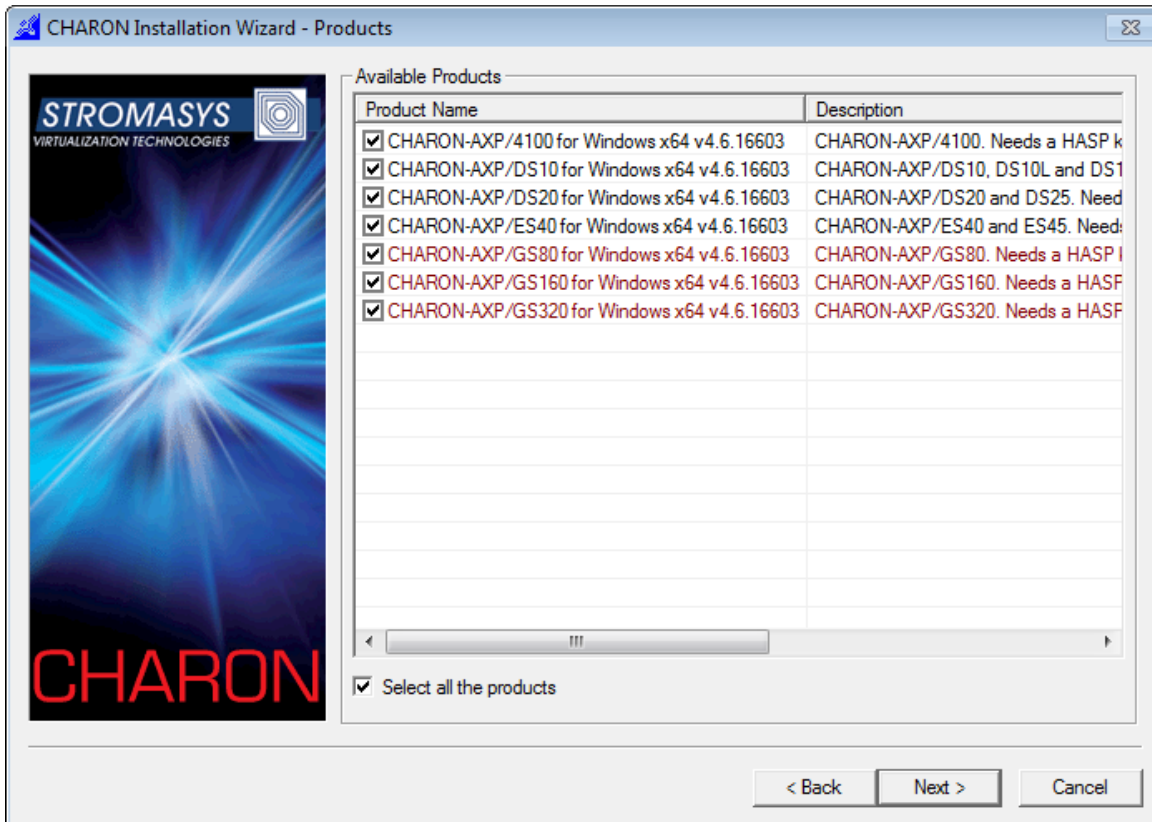
In the list of applications, double click on "CHARON Products":



In the appearing CHARON installation dialog, select "Remove components or products" and press "Next":

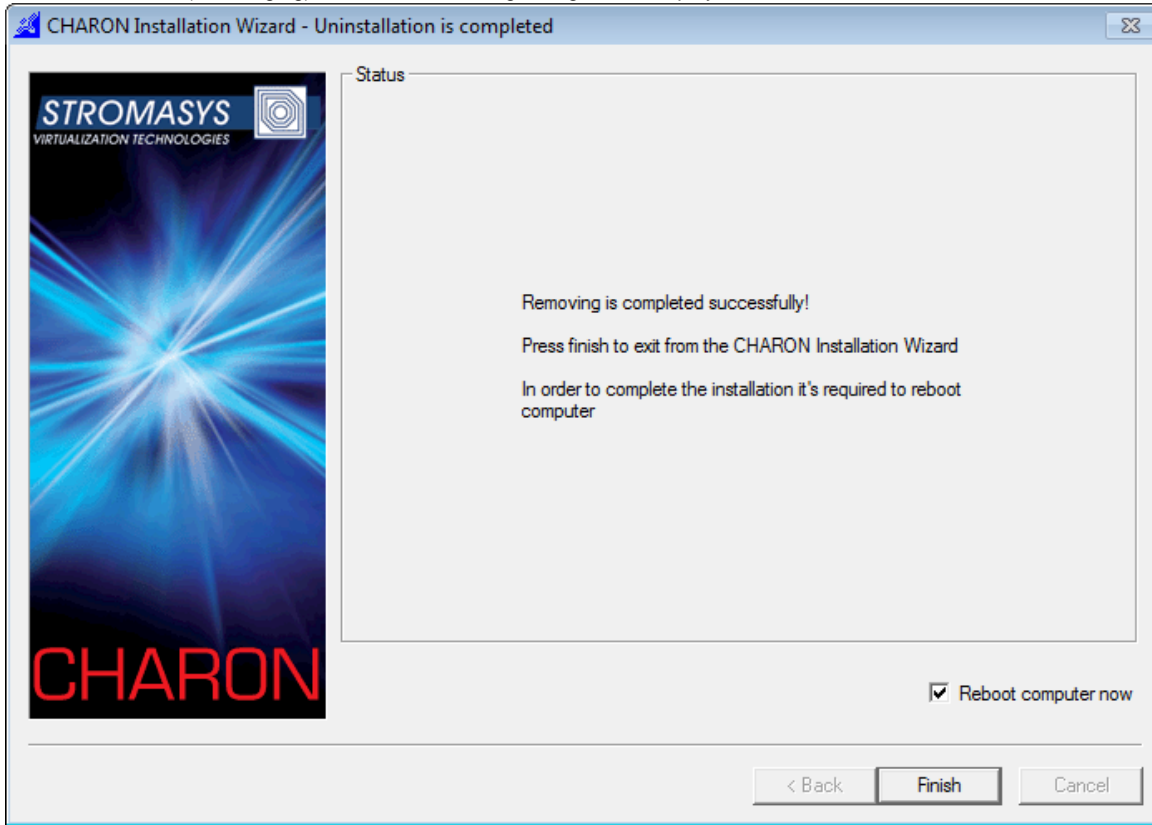


In the following dialog, select the CHARON products to be removed (or changed). Check "Select all the products" for deinstallation (or changing) of all installed CHARON products. Press "Next":





Once uninstallation (or changing) is done, the following dialog will be displayed:



Press "Finish". The CHARON host will reboot automatically.

[Back to Top](#)