



# **Charon-VAX and Charon-PDP V4.12 for Windows Users Guide**

# Contents

Introduction	3
About this guide	6
CHARON-VAX and CHARON-PDP for Windows installation	8
Running and managing CHARON-VAX and CHARON-PDP for Windows	35
CHARON-VAX and CHARON-PDP for Windows configuration	52
Migration to CHARON-VAX and CHARON-PDP for Windows	71
CHARON-VAX for Windows DSSI cluster	84
CHARON-VAX for Windows CI cluster	90
CHARON-VAX and CHARON-PDP for Windows licensing	94
CHARON-VAX and CHARON-PDP for Windows utilities	121
Host Device Check	122
Network Control Center	125
MkDisk	161
MkDskCmd	170
HASP View	173
License Update Service	175
License expiration check	179
mtd	190
CHARON Log Monitor and Dispatcher	192
HOSTPrint	195
VT-30TV	197
CHARON Guest Utilities for OpenVMS	198
Charon Report Utility	203
CHARON-VAX and CHARON-PDP for Windows configuration details	205
General Settings	206
Core Devices	216
Serial lines	223
Remote Management Console (RMC)	239
Disks and tapes	243
MSCP and TMSCP Controllers	244
SCSI Controllers	259
DSSI Subsystem	271
CI Subsystem	281
Networking	289
Specific Controllers	304
Sample configuration files	307
PDP-11/93 configuration file	308
VAX 4000 Model 108 configuration file	310
VAX 6310 configuration file	316
VAX 6610 configuration file	319
"configuration_name.icfg" configuration file	323
CHARON-VAX/PDP for Windows deinstallation	324
Appendixes	329
Log monitoring - logmond best practice	330

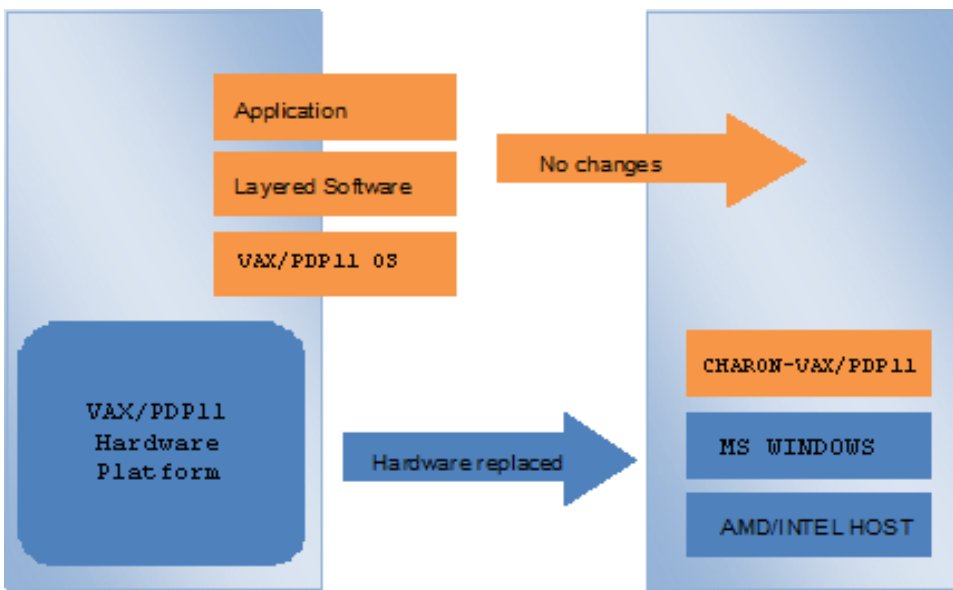
# Introduction

## Table of Contents

- General Description
- The principles of VAX and PDP11 Hardware Virtualization
  - Virtualized hardware
  - Host platform

## General Description

VAX and PDP11 Hardware Virtualization allows users of HP VAX and PDP11 (Previously known as DIGITAL VAX and PDP11) computers to move application software and user data to a modern Intel or AMD based x86 compatible platform without having to make changes to software and data. VAX and PDP11 Hardware Virtualization is a software solution that replaces VAX and PDP11 hardware.



This approach is best understood when the VAX and PDP11 Hardware Virtualization Software is viewed as a special interface between the old VAX and PDP11 software and a new hardware platform. Basically, the CHARON software presents a VAX and PDP11 hardware interface to the original VAX and PDP11 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 or 32-bit Intel or AMD) and continue to run for many more years.

The VAX and PDP11 virtualization software is designed to replace single and multi-CPU VAX and PDP11 computer systems, including:

- PDP11/93
- PDP11/94
- MicroVAX II
- MicroVAX 3600, 3900
- MicroVAX 3100 models 96 and 98
- VAXserver 3600, 3900
- VAX 4000 models 106, 108, 700, and 705
- VAXstation 4000 model 90
- VAX 6310
- VAX 6610, 6620, 6630, 6640, 6650 and 6660

# The principles of VAX and PDP11 Hardware Virtualization

## Virtualized hardware

CHARON-VAX and CHARON-PDP virtualize various VAX and PDP11 architectures and meet or exceed the performance level of these VAX and PDP11 systems when run on the recommended hardware platform. Our VAX and PDP11 emulator products are currently available in the following variants:

**CHARON-PDP** includes:

- PDP11/93
- PDP11/94

**CHARON-VAX** includes:

- MicroVAX II
- MicroVAX 3600
- MicroVAX 3900
- VAXserver 3600 (depending on license supports up to 512 MB of emulated memory)
- VAXserver 3900 (depending on license supports up to 512 MB of emulated memory)
- MicroVAX 3100 models 96 and 98
- VAX 4000 models 106, 108, 700 and 705
- VAXstation 4000 model 90
- VAX 6310
- VAX 6610
- VAX 6620
- VAX 6630
- VAX 6640
- VAX 6500
- VAX 6660

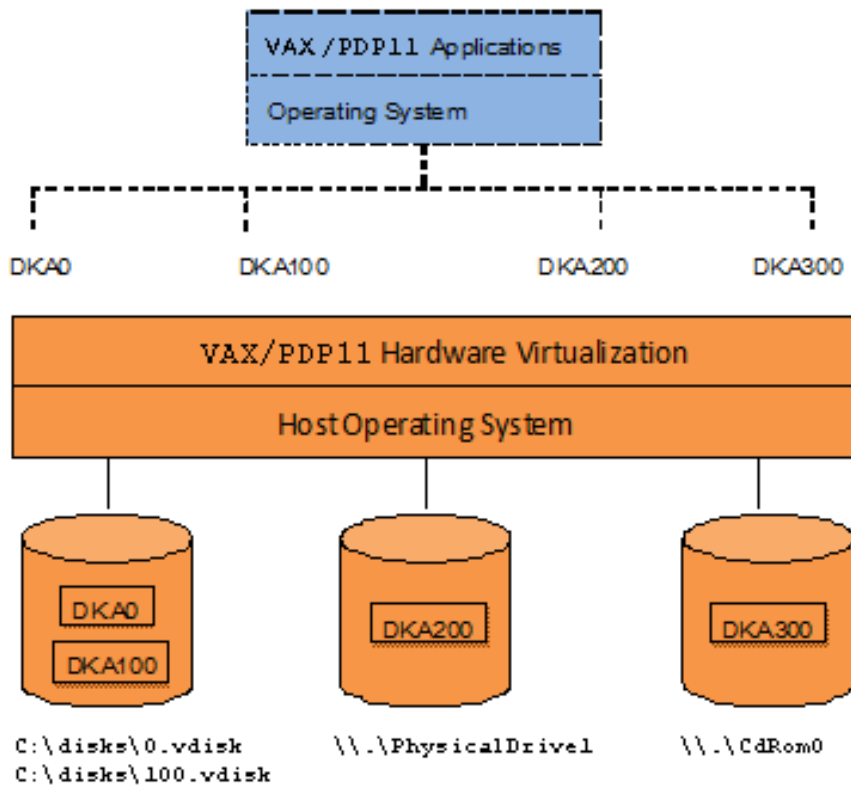
The following table explains which hardware boards CHARON virtualizes:

Subsystem	Covered VAX and PDP11 hardware
Serial Lines Controllers	UART, QUART, CXA16, CXB16, CXY08, DHQ11, DHV11, DZV11, DZQ11, DL11, DLV11, DZ11, DHW42-AA, DHW42-BA, DHW42-CA
XMI Disks/Tapes Controllers	KDM70
BI Disks Controllers	KDB50
QBUS Disks Controllers	RQDX3
QBUS/UNIBUS Tapes Controllers	TQK50, TUK50
SCSI Controllers	NCR53C94
DSSI Subsystem	SHAC, HSD50
CI Subsystem	CIXCD, HSJ50
Network Controllers	DEQNA, DESQA, DELQA, DEUNA, DELUA, DEMNA, DEBNI, PMADAA



## Host platform

The Virtualization Software presents standard VAX and PDP11 devices to the VAX and PDP11 operating system, allowing the OS to function as though it were still running on a VAX and PDP11 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 VAX and PDP11 OS as emulated SCSI disks attached to a 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.

## About this guide

### Table of contents

- Obtaining Documentation
- Obtaining Technical Assistance or General Product Information
  - Obtaining Technical Assistance
  - Obtaining General Product Information
- Conventions
- Definitions
- Related documents

## Obtaining Documentation

The latest released version of this manual and other related documentation are available on the Stromasys support website at [Product Documentation and Knowledge Base](#).

## Obtaining Technical Assistance or General Product Information

### Obtaining Technical Assistance

Several support channels are available to cover the Charon virtualization products.

**If you have a support contract with Stromasys**, please visit <http://www.stromasys.com/support/> for up-to-date support telephone numbers and business hours. Alternatively, the support center is available via email at [support@stromasys.com](mailto:support@stromasys.com).

If you purchased a Charon product through a Value-Added Reseller (VAR), please contact them directly.

### Obtaining General Product Information

If you require information in addition to what is available on the Stromasys [Product Documentation and Knowledge Base](#) and on the [Stromasys web site](#) you can contact the Stromasys team using <https://www.stromasys.com/contact/>, or by sending an email to [info@stromasys.com](mailto:info@stromasys.com).

For further information on purchases and the product best suited to your requirements, you can also contact your regional sales team by phone:

Region	Phone	Address
Americas	+1 919 239 8450	Stromasys LLC 871 Marlborough Ave, suite 100, Riverside CA 92507 USA
Europe, Middle-East and Africa	+41 22 794 1070	Avenue Louis-Casai 84 1216 Cointrin Switzerland

## Conventions

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>dsk0</i>	Italic monospace type, in interactive examples, indicates typed context dependent user input.

## Definitions

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

## Related documents

- [Charon-VAX/PDP V4.12 for Windows - User's Guide \(still internal!\)](#)
- [Charon-VAX V4.12 for Windows - Release Notes](#)
- [Charon-PDP V4.12 for Windows - Release Notes](#)

# CHARON-VAX and CHARON-PDP for Windows installation

## Table of contents

- Introduction
- Host Hardware Requirements
  - Number of CPU cores
  - CPU type and speed
  - Memory
  - Disk storage
  - Ethernet adapters
- Software Requirements
- Host system preparation
- Before installation
- Upgrading From Older Versions
- Note Network Interfaces Before Upgrading
- Distribution preparation
- Installation
- CHARON-VAX / CHARON-PDP home directory
- License installation
  - Regular HASP USB dongle
  - Network HASP USB dongle
  - Software license
- License validity verification
  - Troubleshooting
- Network configuration
  - Dedicating a Network Interface to Charon-VAX
  - Disabling TCP Chimney Offloading
  - Disable Network Interface Attributes Manually
- After installation

## Introduction

CHARON-VAX / CHARON-PDP products are distributed in form of a zip file that contains the CHARON-VAX / CHARON-PDP 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).

CHARON-VAX / CHARON-PDP installation consists of the following steps:

- Perform host system checks (hardware and software) to ensure the host platform meets minimum CHARON-VAX / CHARON-PDP installation requirements
- Optionally, create a specific CHARON username with administrative privileges
- Extract CHARON-VAX / CHARON-PDP installation procedure files from its archive
- Install CHARON-VAX / CHARON-PDP by running "InstallShell.exe" installer and following the instructions
- Install CHARON-VAX / CHARON-PDP license (hardware dongle or software license)
- Configure CHARON-VAX / CHARON-PDP host system, for example, network adapter, host OS services, etc.

# Host Hardware Requirements

## Number of CPU cores

Each CHARON-VAX / CHARON-PDP emulated CPU requires a corresponding physical core. Thus the total number of the host CPUs must exceed the number of emulated CPUs since some of the host CPUs must be dedicated to serving CHARON I/O operations and host operating system needs.

The following table shows the minimum and recommended number of CPUs required for VAX/PDP11 models:

CHARON-VAX / CHARON-PDP model	Minimal number of host CPU cores	Recommended number of host CPU cores
VAX 6610	2	4
VAX 6620	3	4
VAX 6630	4	6
VAX 6640	6	8
VAX 6650	8	12
VAX 6660	8	12
Other models (all single CPU models)	2	2

When starting, the CHARON-VAX software checks the available number of host CPU cores. Currently, this check is based on the maximum number of VAX CPUs that can be emulated, not on the number of the actually configured VAX CPUs. Therefore the number of host CPU cores recommended for the maximum number of emulated CPUs - as shown in the right column of the table above - must be available. If the available number of host CPU cores is below this number, CHARON-VAX will issue a warning message even if the requirements for the configured number of VAX CPUs are fulfilled. The CHARON-VAX software will work despite this warning if the requirements for the configured number of VAX CPUs are fulfilled.

**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-VAX / CHARON-PDP utilizes the LAHF instruction during VAX/PDP11 CPU emulation, please avoid using early (pre-2005) AMD64 and Intel 64 CPUs in 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 VAX/PDP11 performance. The minimal recommendation is at least 2.4 GHz.

## Memory

The minimum host memory size depends on the amount of VAX/PDP11 memory to be emulated and on the number of CHARON-VAX / CHARON-PDP 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 VAX/PDP11 memory emulated) per CHARON-VAX / CHARON-PDP instance.

**Please note** that the maximum amount of VAX memory that can be created in the CHARON-VAX/66x0 products and supported by OpenVMS/VAX is 3584 Mb.

For details, see the memory size specification.

Only in the case of a 32-bit host OS, when more than 1.5 GB VAX memory is configured, a 32 bit Windows host system must be configured for "large application memory". This is done by adding the /PAE and /3GB switches to the Windows boot.ini file.

To set the /PAE and /3GB startup switch in Windows:

1. Right-click My Computer and select Properties. The System Properties dialog box will appear.
2. Click the "Advanced" tab.
3. In the "Startup and Recovery" area, click "Settings". The "Startup and Recovery" dialog box will appear.
4. In the System startup area, click "Edit". This will open the Windows boot.ini file in Notepad.
5. In the [Operating Systems] section, add the following switches to the end of the startup line that includes the /fastdetect switch: /PAE and /3GB

6. Save the changes and close Notepad.
7. Click OK two times to close the open dialog boxes, and then restart the computer for the change to take effect.

## Disk storage

The total amount of disk space required for CHARON-VAX / CHARON-PDP can be calculated as a sum of all the disk/tape image sizes plus 500 MB for the CHARON software plus space required for the normal host OS.

## Ethernet adapters

The number of host network interfaces (NICs) should be the sum of NICs needed for the host, normally one, and the number of NICs that will be emulated.

## Software Requirements

- Microsoft Windows Server 2022 Standard and Datacenter Editions, 64 bit version
- Microsoft Windows Server 2019 Standard and Datacenter Editions, 64 bit version
- Microsoft Windows Server 2016 Standard and Datacenter Editions, 64 bit version
- Windows 10 Pro and Enterprise Editions, 32 bit and 64 bit version
- Windows 11 Professional and Enterprise Editions, 64 bit version
- VMware ESXi 5.5 and 6.x up to 6.7 (requires a supported Windows operating system running in the ESXi VM)
- Microsoft Hyper-V (requires a supported Windows operating system running in the Hyper-V VM)

Deprecated Windows Versions (Stromasys strongly discourages their use)

- Microsoft Windows Server 2012 R2 Standard and Datacenter Editions, 64 bit version (please notice that only R2 revision is supported)
- Microsoft Windows Server 2008 R2 (SP1) Standard and Enterprise Editions, 64 bit version (please notice that only R2 revision is supported)
- Microsoft Windows 7 Professional and Ultimate (SP1) Editions, 32 bit and 64 bit versions
- Microsoft Windows 8.1 Professional Edition, 32 and 64 bit versions

## Host system preparation

### Important notes:

- 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. Alternatively it is also possible to exclude all CHARON executables and material folders from scanning for viruses, but please note that scanning slows down CHARON host anyway - it may be critical for some installations requiring maximum performance.
- 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 shut down the operating system running on CHARON and stop all the running CHARON instances and services.

A network-wide license (red dongle or software license) requires the following operations to be performed:

- *On license server (where network license will reside)*: open port 1947 for both TCP and UDP
- *On the client*, if broadcast search for remote licenses is to be used, UDP traffic from port 1947 of the license server to ports 30000-65535 of the client must be permitted.
- *On both server and client*: set up default gateway

Please consult with Windows [online documentation](#) 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-VAX / CHARON-PDP11).

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 applet discovered.
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.

## Before installation

1. Login as the local system administrator ("Administrator") on the host system.
2. Create a special user for running CHARON-VAX / CHARON-PDP (optional). This user must have all the administrative privileges. Please consult with Windows [online documentation](#) for details.

**Please note** that the CHARON 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 remove any CHARON user from it, otherwise a [problem with virtual memory allocation](#) may appear on CHARON startup.

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-VAX / CHARON-PDP user having administrative privileges (if this option has been chosen).
4. Create a special directory for the CHARON-VAX / CHARON-PDP distribution kit and copy the provided files there.

## Upgrading From Older Versions

From version	Action
4.7 and below	First uninstall all CHARON products.
4.8 and above	<p>The version 4.11 can be installed without uninstalling the previous version(s).</p> <p><b>⚠</b> Before upgrading, please run the Network Control Center and take note of all network interfaces dedicated to Charon. After the upgrade, they will have to be dedicated again to Charon as they will all appear released to the host.</p> <p><b>⚠</b> Please note it is mandatory to update all existing CHARON Virtual Machine to make them run the latest version using the Virtual Machine Manager, VM Configuration tab - otherwise they will remain at the previous installed version.</p>




## Note Network Interfaces Before Upgrading

Before upgrading, please run the Network Control Center and take note of all network interfaces dedicated to Charon. After the upgrade, they will have to be dedicated again to Charon as they will all appear released to the host.

## Distribution preparation

- Extract the content of the distribution archive to the current directory or copy the content of the provided distribution directory to the current directory.
 

**⚠** The kit folder must not be the same as the target installation folder (Example: C:\Program Files\CHARON\Build\_20410) otherwise the installation procedure may report "Charon is in use".
- Run the "InstallShell.exe" file:

 HASP.def	06.05.2020 23:14
 InstallShell.exe	06.05.2020 23:14
 InstallShellCHS.dll	06.05.2020 23:14

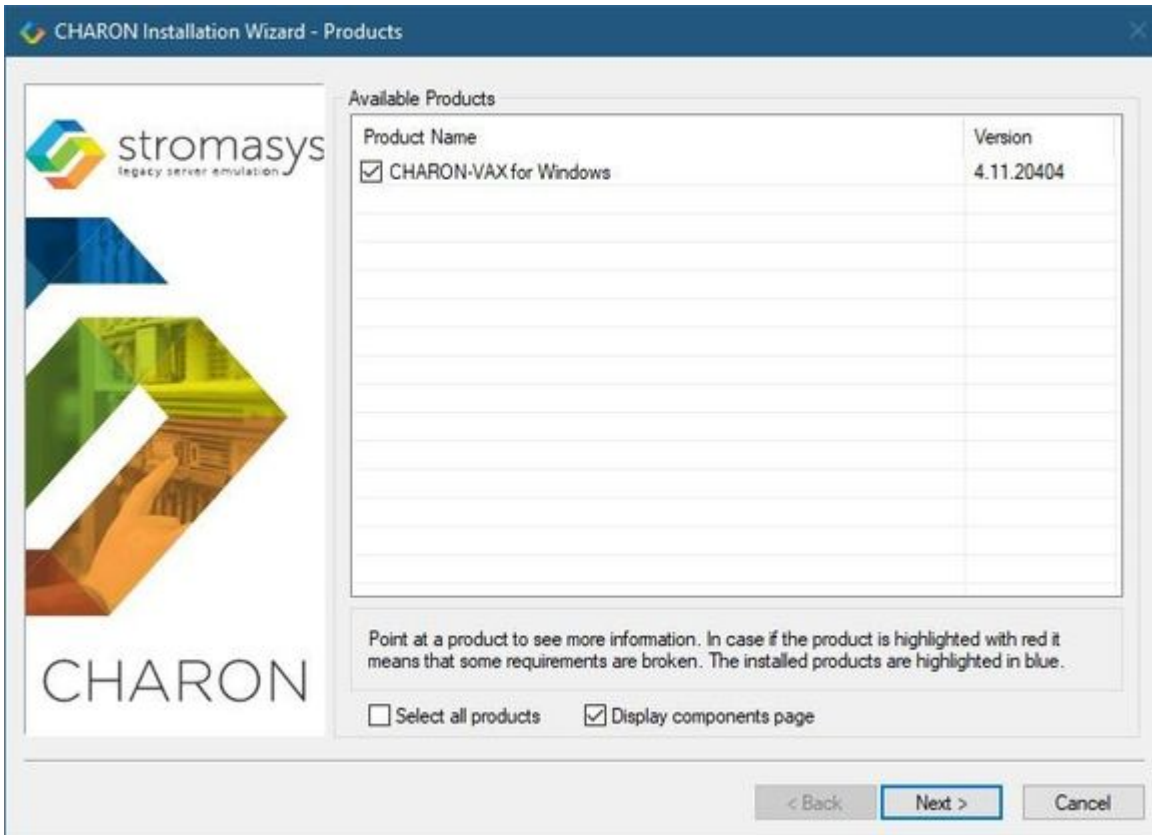


# Installation

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

All the following examples are given for CHARON-VAX product. Installation of CHARON-PDP/11 is the same except for the name of the product.

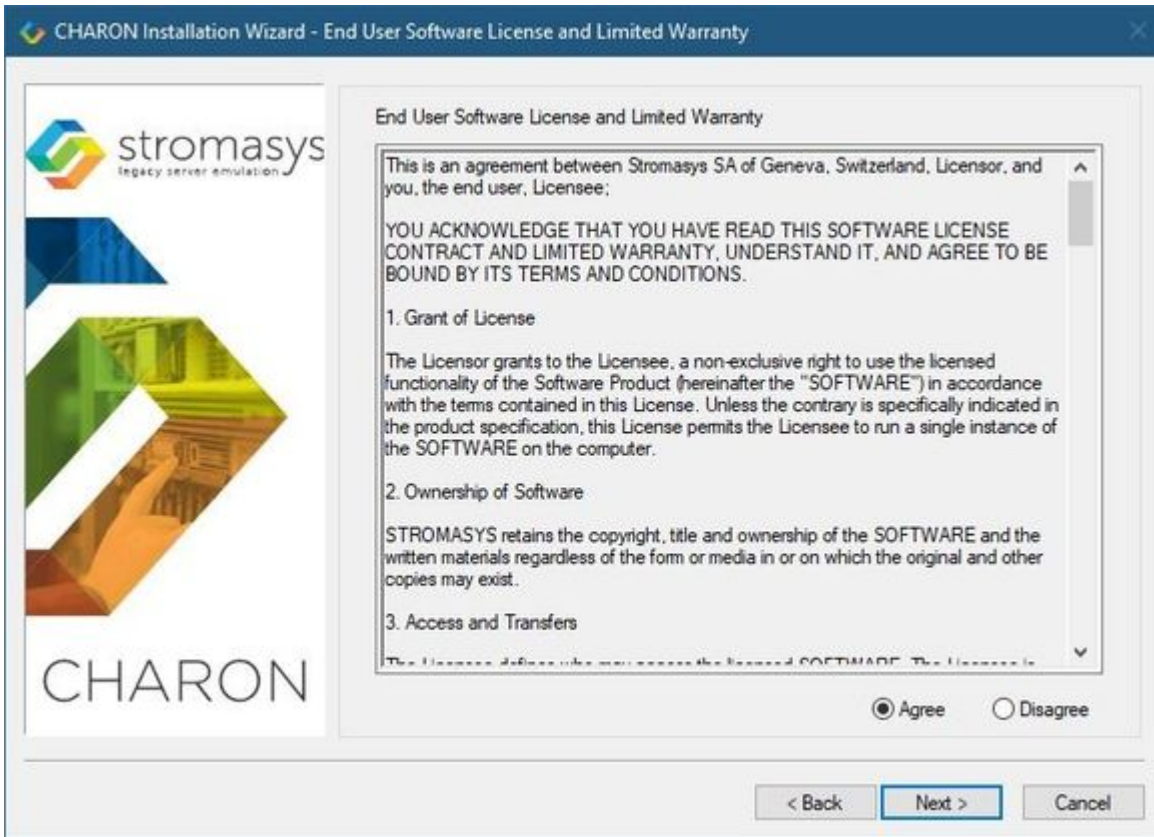
The first dialog lists the CHARON-VAX / CHARON-PDP product:



Select the CHARON-VAX product, or all products, by checking the corresponding box. Select the "Display components page" to include the products components page if you need it. Typically it is not required.

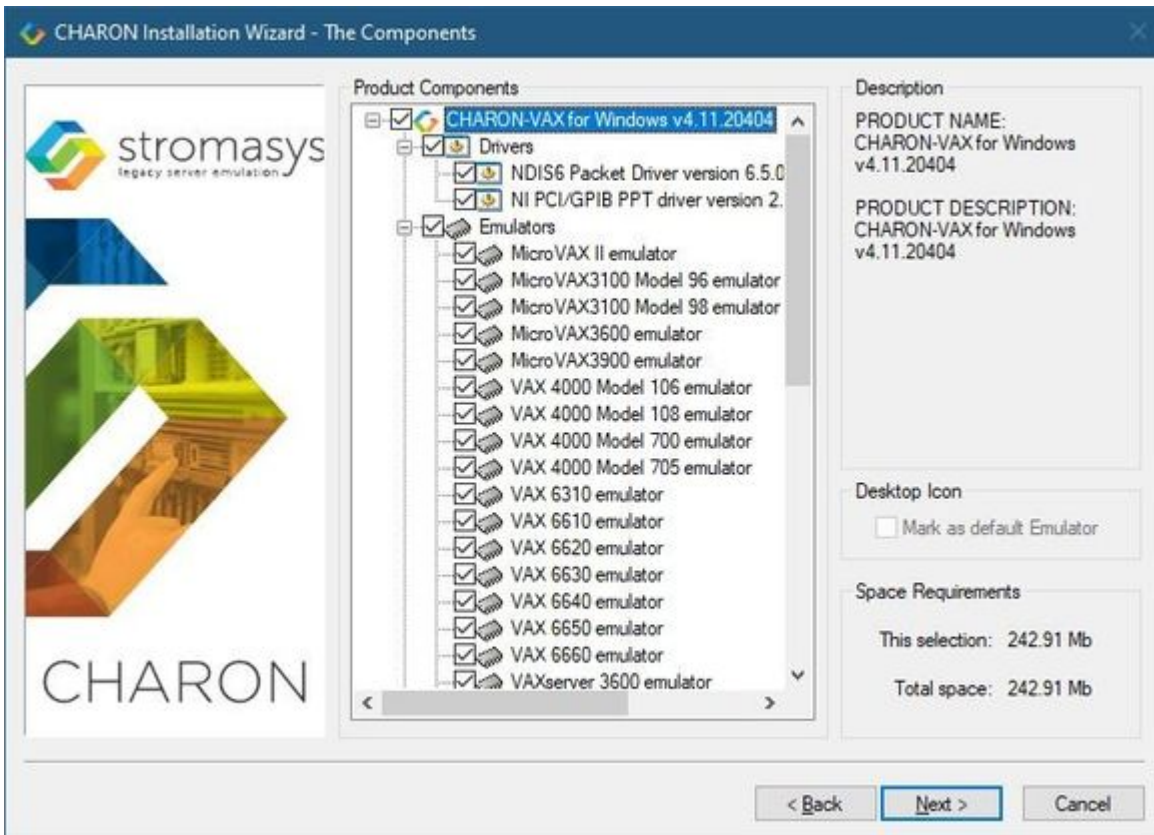
**Please note** that the CHARON products may be highlighted in red. This indicates that the CHARON host configuration does not meet CHARON requirements. To see what requirements are not met, point the mouse cursor on that product and a pop-up box with details will appear.

Press the "Next" button.

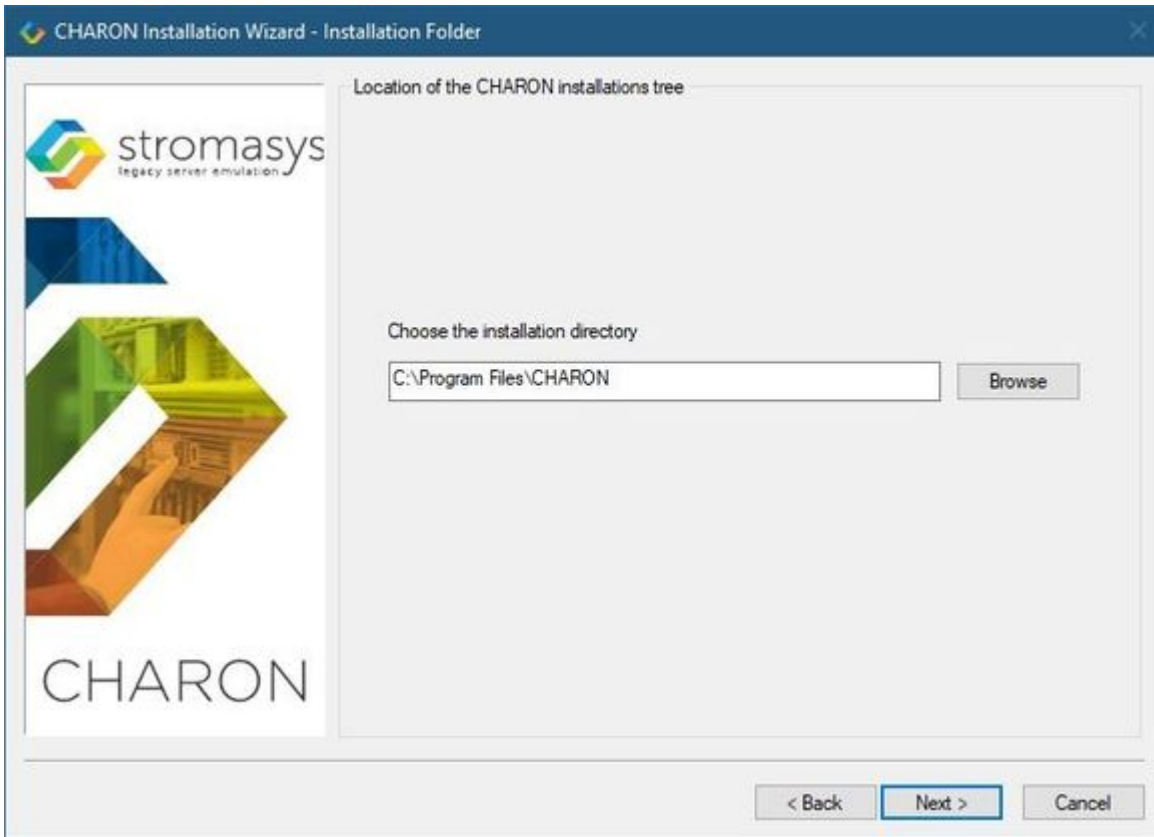


Read the license, check "Agree" if you agree and press the "Next" button.

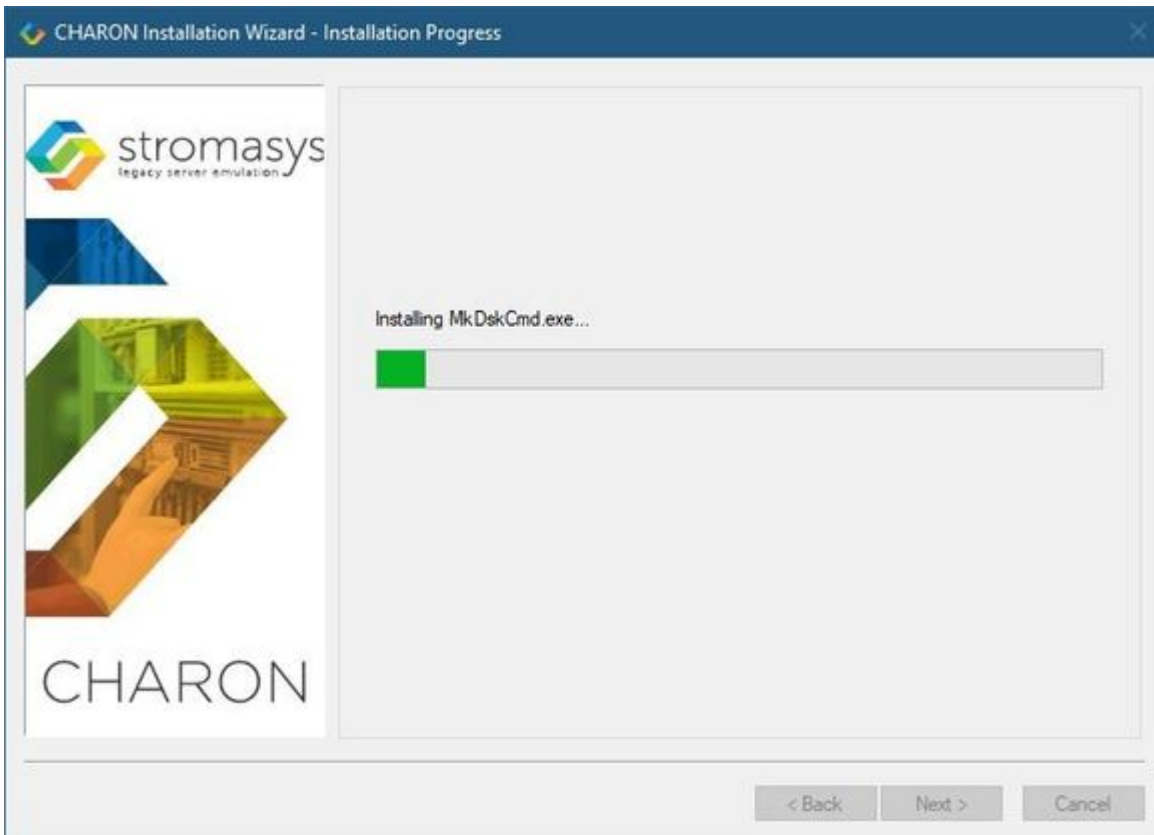
If "Display components page" option was selected, the setup procedure will display a dialog for the CHARON-VAX product components:



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



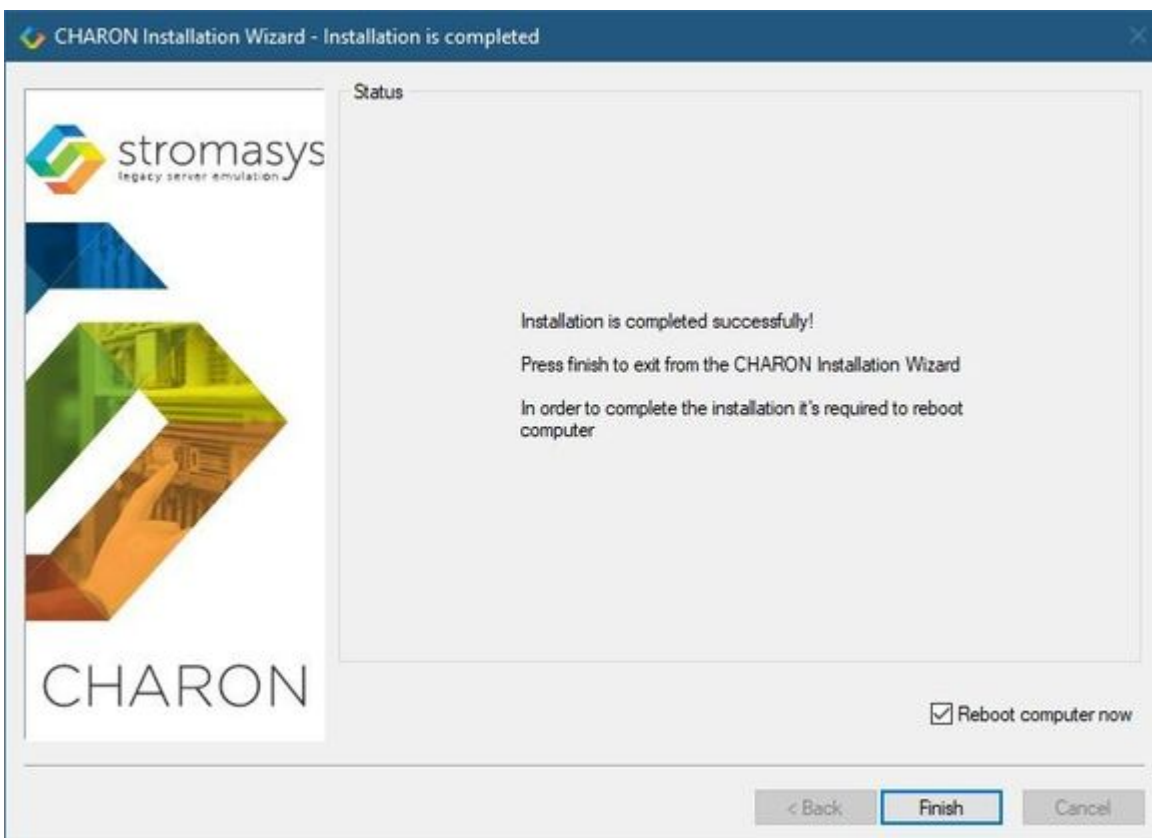
Select a directory to be used for CHARON installation or use the default. The path can either be entered manually or selected using the "Browse" button. Once done, press the "Next" button to start the installation of the selected components.



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 a new shortcut, "CHARON Virtual Machine Manager", created on the desktop during the installation procedure:



The CHARON Virtual Machine Manager is an essential utility for CHARON management. CHARON installation always keeps most up-to-date version of CHARON Virtual Machine Manager provided by all the installed kits.

**When upgrading** from V4.8, V4.9, V4.10 or V4.11, update all existing CHARON virtual machines to make them run the latest version - otherwise they will remain at the previous version. See the following chapters for details.

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



## CHARON-VAX / CHARON-PDP home directory

By default CHARON is installed in the "C:\Program Files\CHARON" directory. It has the following subdirectories:

Directory	Description
Build_XXX\x86 (and Build_XXX\x64 on x64 systems)	Contains product executables and libraries
Drivers	Contains the CHARON drivers
Chapi	Contains CHARON API files used for creating new emulated devices
ChMac	Chmac is a service allowing 32bit Charon VAX and PDP to change MAC address on 64 bit systems
InstallShell	Contains the CHARON installation procedure executables
Logs	Contains the CHARON installation log
NCC	Contains the Charon Network Control Center executable
Redistributables	Contains the redistributables needed for CHARON running (already installed, except the license alert script)
Virtual Machine Manager	Contains the most up-to-date version of CHARON Virtual Machine Manager
Virtual Machine Templates	Contains templates for creating CHARON virtual machines
Virtual Machines	Contains the CHARON virtual machines
Virtual Disk Images	Contains CHARON virtual disk images. For example " <a href="#">charoncp013.vdisk</a> " and some OpenVMS tools used for debugging (ovms_tools.vdisk)

The "Virtual Machine Templates" directory contains template configuration files to build CHARON Virtual Machine using the CHARON Virtual Machine Manager. We will focus our attention on this subject in the next chapter.

## License installation

### Regular HASP USB dongle

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

**Please note:**

- 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 or some other console-capable program must be used.
- Network HASP (red dongles) licenses have no restrictions with respect to remote access.
- Software licenses are always network licenses. Hence, they have no restrictions with respect to being displayed or accessed via a remote connection.
- Regular USB dongles are not supported with Microsoft Hyper-V.

### Network HASP USB dongle

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

If a remote license server is to be used:

- Copy the file "hasp\_install\haspdinst.zip" from the CHARON distribution to any directory on the server, for example "C:\Temp".
- Extract the content 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
```

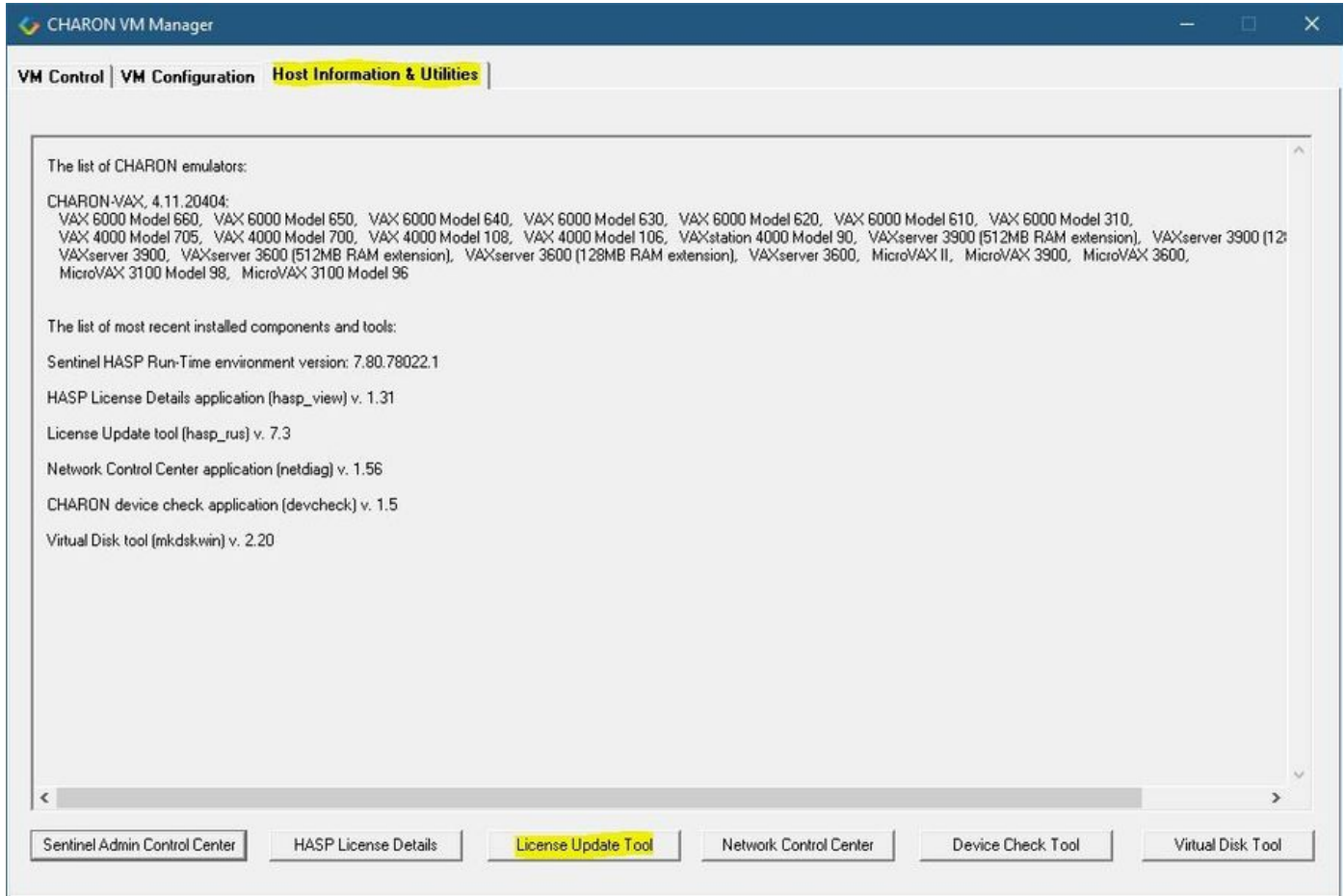
- Copy the file "hasp\_install\hasplib.zip" from the CHARON distribution to any directory on the server, for example "C:\Temp".
- Extract the contents of this archive to the same directory.
- Copy the file "haspvlib\_68704.dll" to "C:\Program Files (x86)\Common Files\Aladdin Shared\HASP" (in case of x64 host) or "C:\Program Files\Common Files\Aladdin Shared\HASP" (in case of x86 host)
- Connect the network HASP dongle to the server USB port.



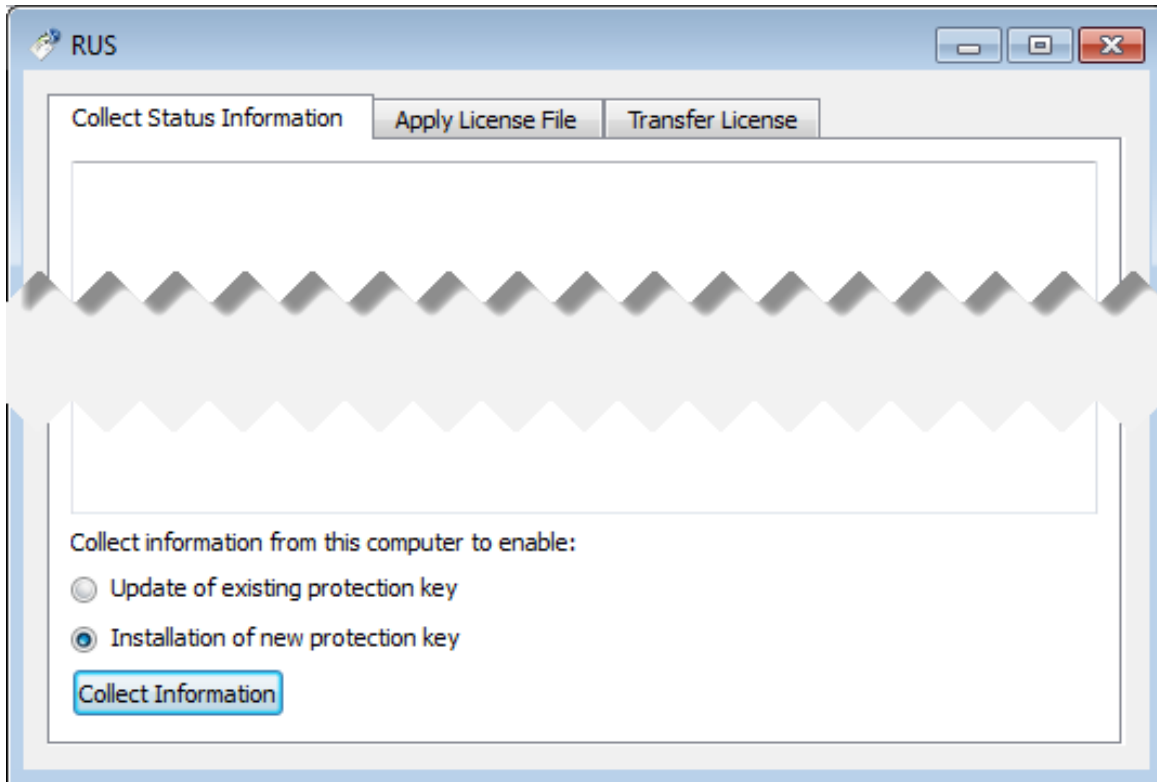
## Software license

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

1. Double click the "CHARON Virtual Machine Manager" icon on the desktop or select this utility in the tray menu; select the "Host Information and Utilities" section and press the "License Update Tool" button:

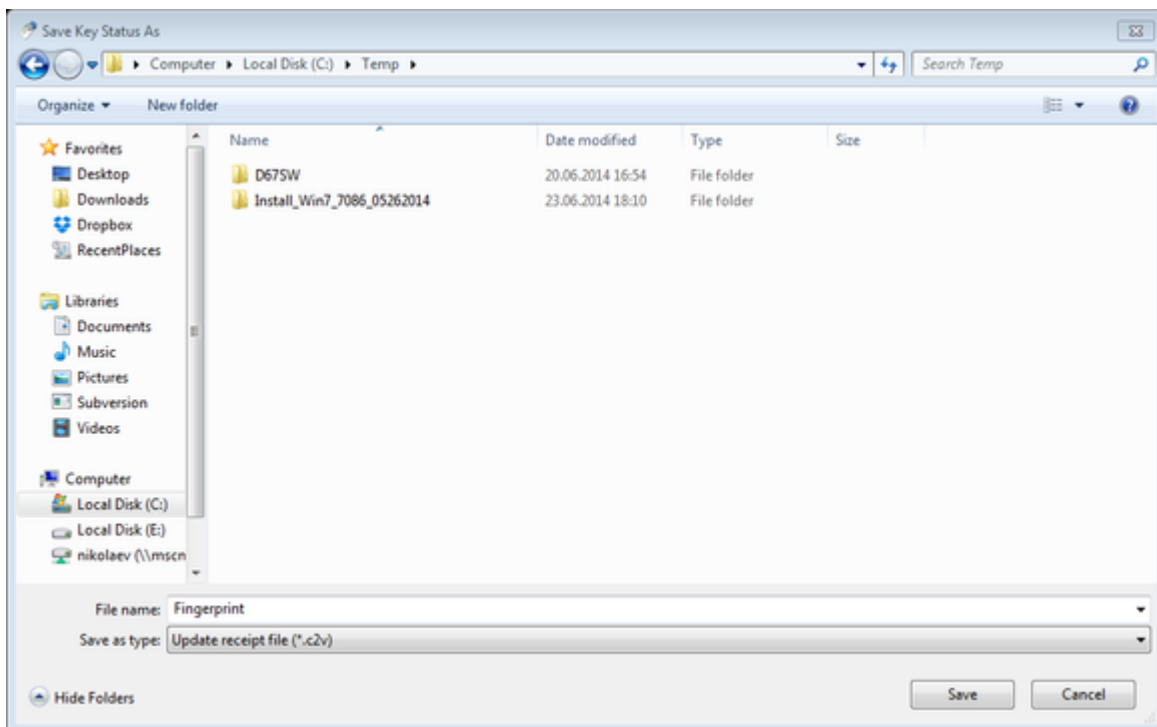


The following default dialog window will appear:



If you're installing a "Provisional" (demo) license, jump to step 5.

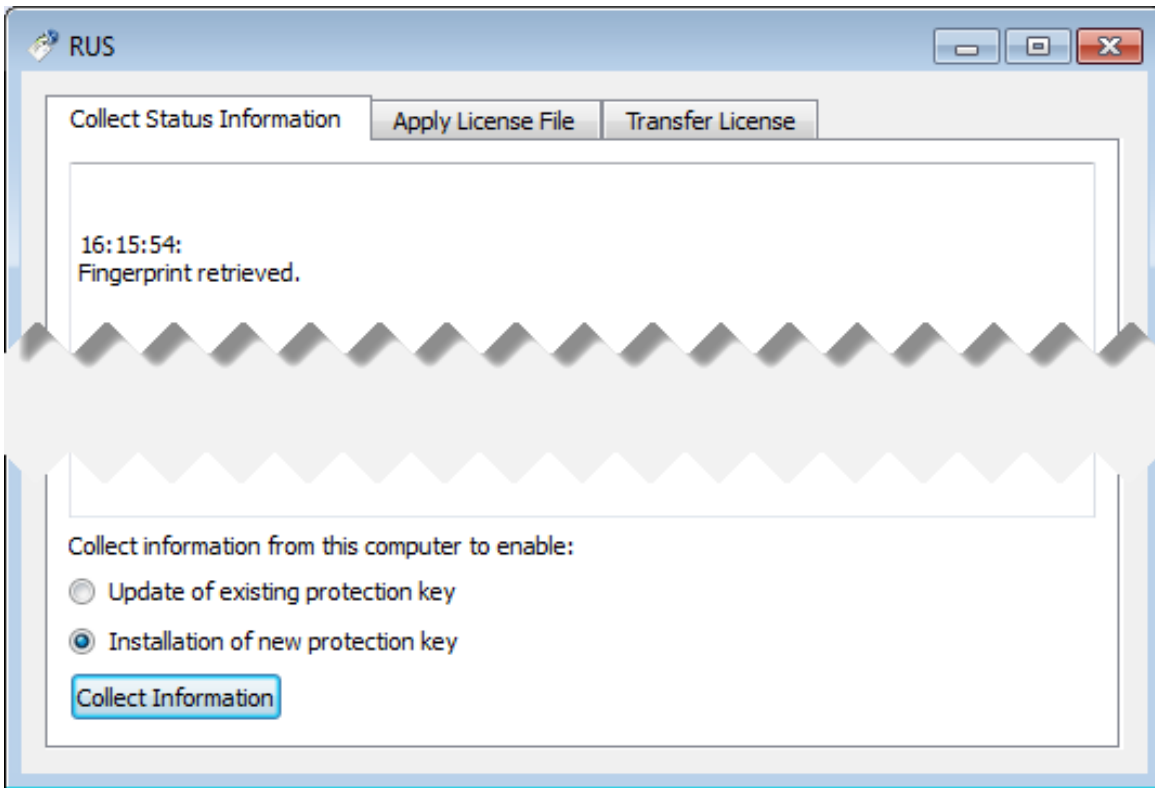
2. Ensure no license dongle is connected, select the "Installation of new protection key" radio-box and press the "Collect Information" button. The following window will appear:



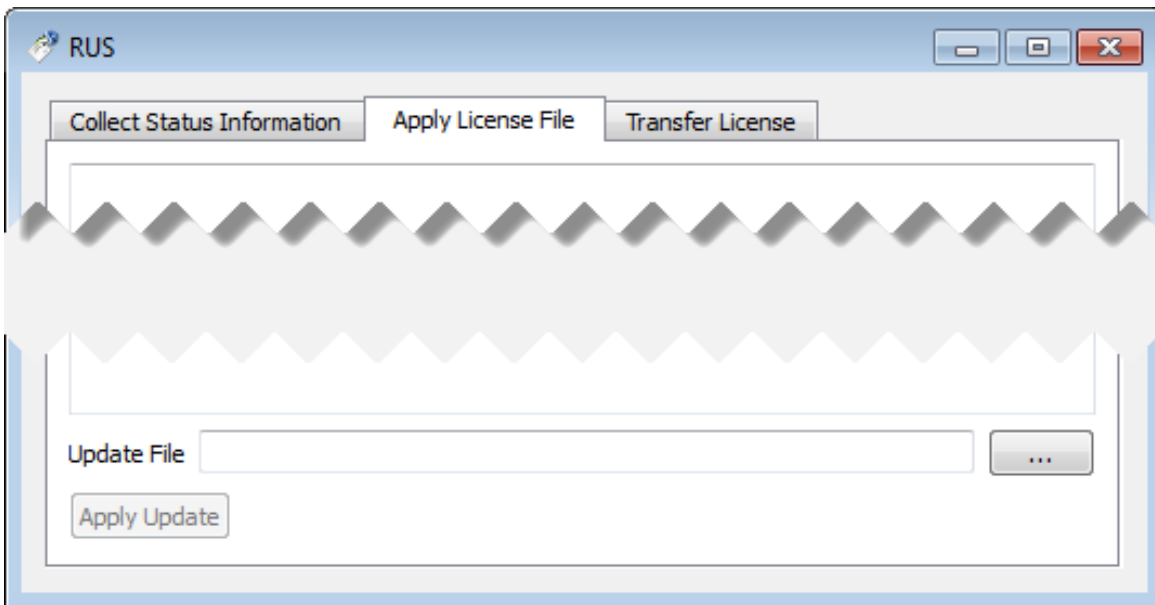
Specify the directory and name of the system "fingerprint" file to be created then press the "Save" button.



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

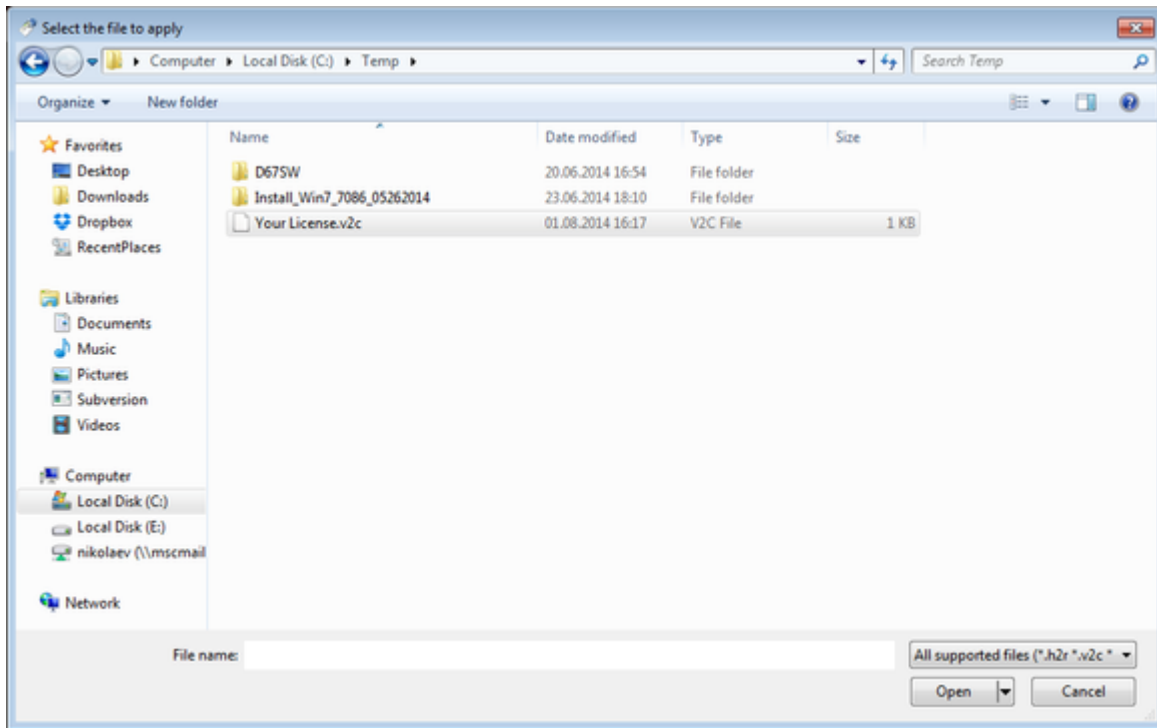


4. Send the resulting file to STROMASYS. In return STROMASYS will provide you with a ".v2c" file, for example "Your License.v2c".
5. Copy the received v2c file to CHARON host in the folder of your choice and select the "Apply License File" tab:



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

**Example:**

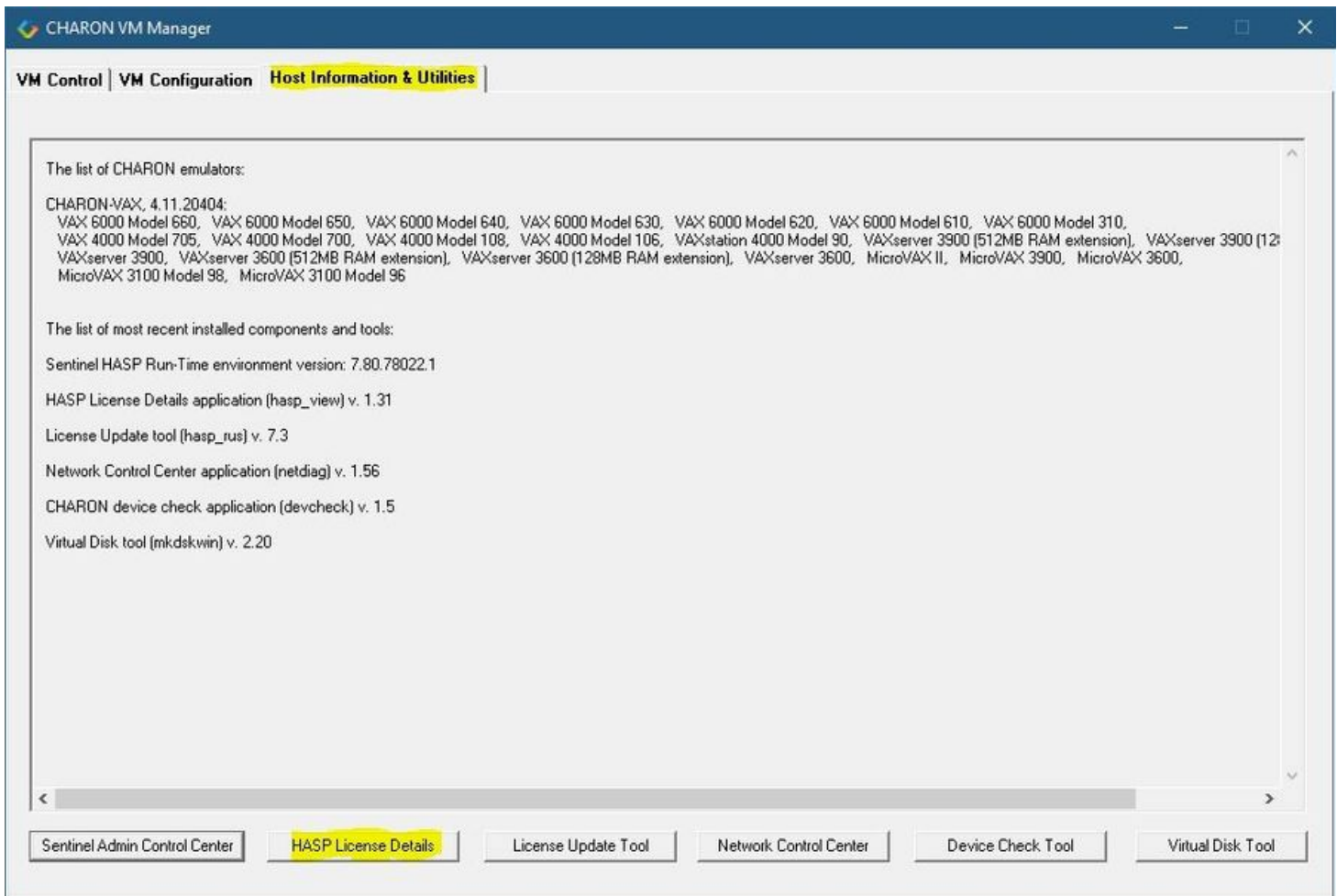


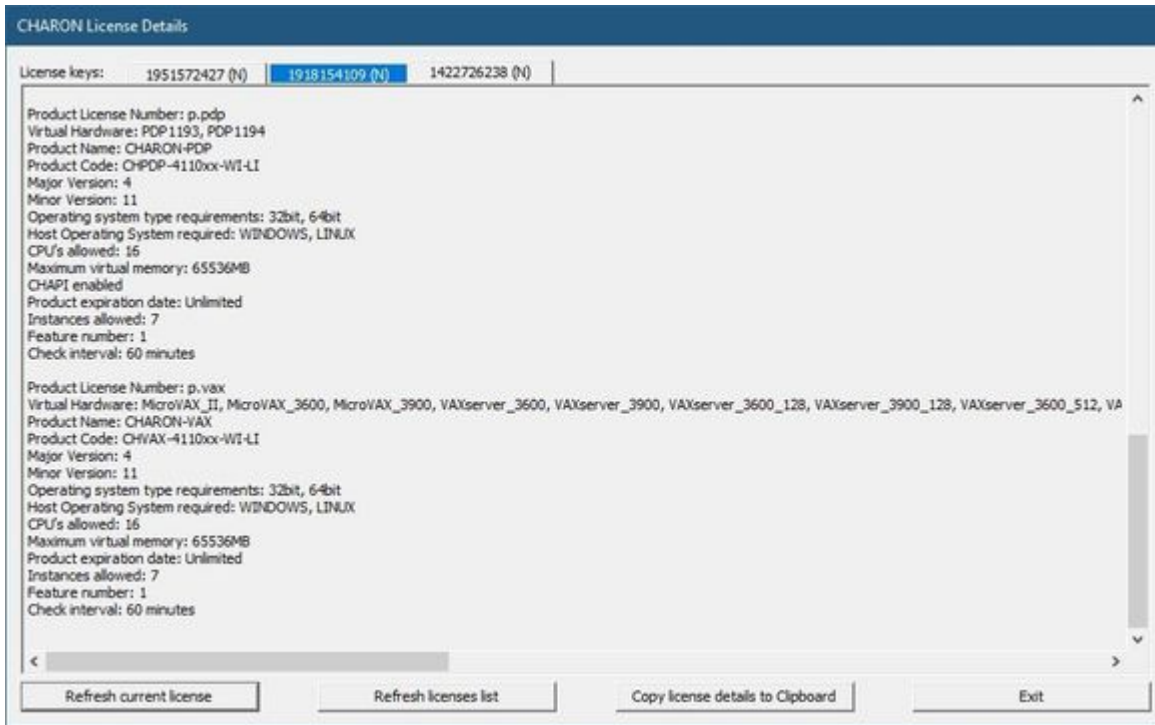
Press the "Open" button to apply the license.

- From the "CHARON Virtual Machine Manager" utility (see above) select the "Host Information and Utilities" section and press the "Sentinel Admin Control Center" button or (optionally) invoke the system default web browser and enter the URL <http://localhost:1947> to display the "**Sentinel Admin Control Center**" (ACC) web interface. This interface allows you to view and manage the CHARON licenses.
- Ensure that the software license is now visible in the "**Sentinel Keys**" section of the ACC.

## License validity verification

Check the CHARON license validity by invoking the "Charon License Details" utility to make sure that the CHARON license is visible and is correct. Double click the "CHARON Virtual Machine Manager" icon on the desktop or select this utility in the tray menu; select the "Host Information and Utilities" section and press the "HASP License Details" button:






- Verify the the license content is displayed correctly and no error message is shown
- Verify the content of the license looks correct. Check for example the license number, the major and minor versions, the minimum and maximum build numbers, the CHARON-AXP products and allowed hardware (CHARON-AXP models). More details on the license content can be found in the [CHARON-VAX / CHARON-PDP Licensing](#) chapter of this Guide.

## Troubleshooting

If the CHARON license content cannot be displayed by the "[CHARON License Details](#)" utility or it is incorrect, check that the license is available and correctly used:

1. From the "CHARON Virtual Machine Manager" utility (see above) select the "Host Information and Utilities" section and press the "Sentinel Admin Control Center" button or (optionally) invoke the system default web browser and enter the URL <http://localhost:1947> to display the "**Sentinel Admin Control Center**" (ACC) web interface.
2. Click on 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.

 To facilitate troubleshooting, Stromasys recommends enabling logging from the Sentinel Admin Control Center as described in this article: [Enabling logging in Sentinel Admin Control Center](#).

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

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

Software licenses can also be uninstalled using the method, described in the "[Removing CHARON-VAX Software Licenses](#)" chapter of this Guide.

For license servers accessible only via non-broadcast search it is also possible to disable access to network licenses if only a local license is to be used: Click on the "**Configuration**" link to open the "**Configuration for Sentinel Manager**" page.

Uncheck the "**Allow Access to Remote Licenses**" checkbox from the "**Access to Remote License Managers**" tab then press the "**Submit**" button to apply changes.

### Please note:

- Starting with Charon-AXP/VAX 4.9 for Linux and Charon-AXP/VAX version 4.8 for Windows the Charon emulator products do not follow the settings in the Sentinel ACC with respect to querying remote license servers and network visibility. They perform a **broadcast search** for network licenses even if this has been disabled in the Sentinel ACC. If this behavior has to be prevented for specific reasons, the network access of the system has to be temporarily restricted or disabled, for example by blocking the relevant traffic in a firewall. Another possibility would be to block access to the network license at the license server side.
- Note that such methods can negatively impact other functions of the system or, in the case of blocking access to a network license on the server, even the functions on other license clients.

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

### Example:

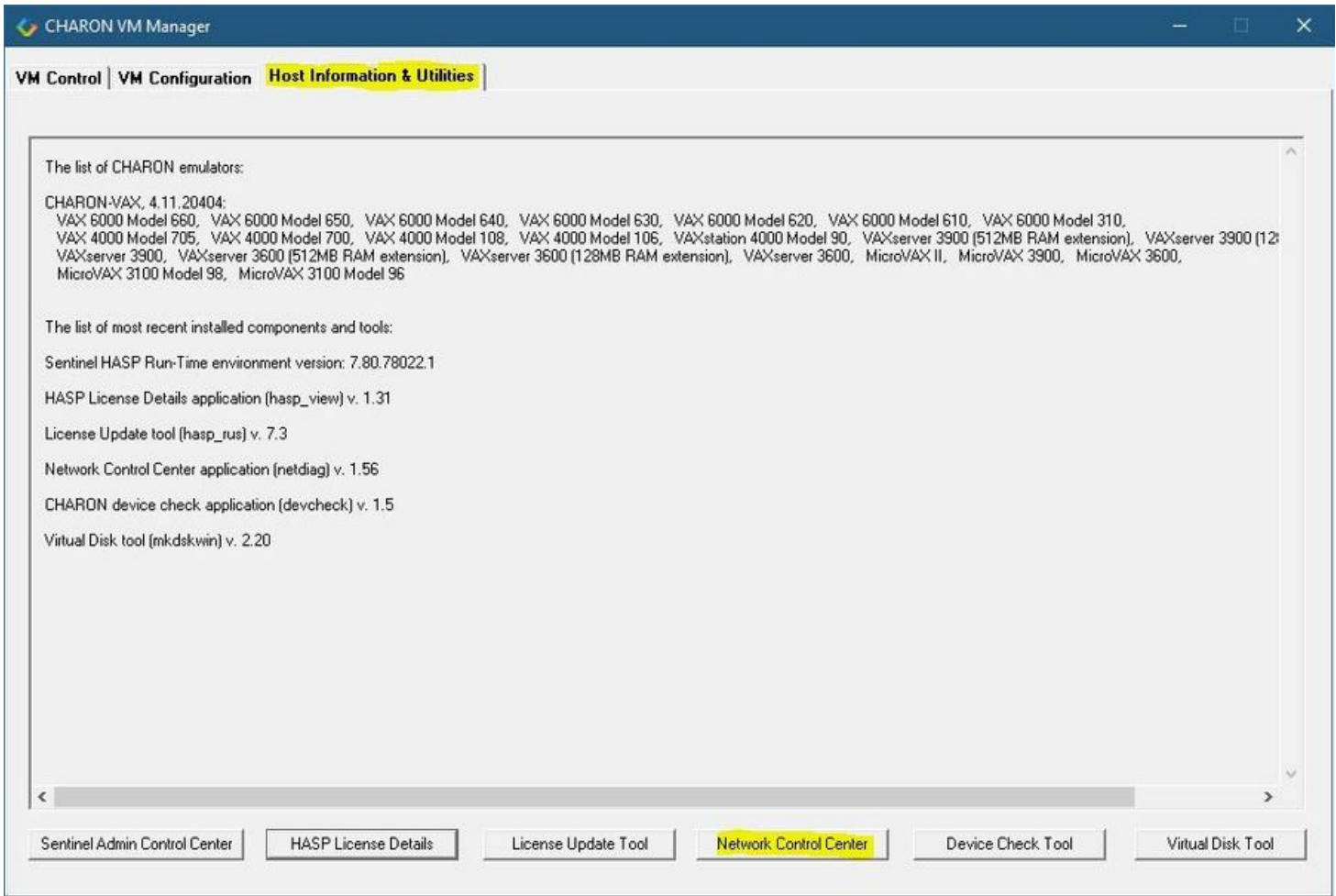
```
set session license_key_id="1877752571,354850588"
```

# Network configuration

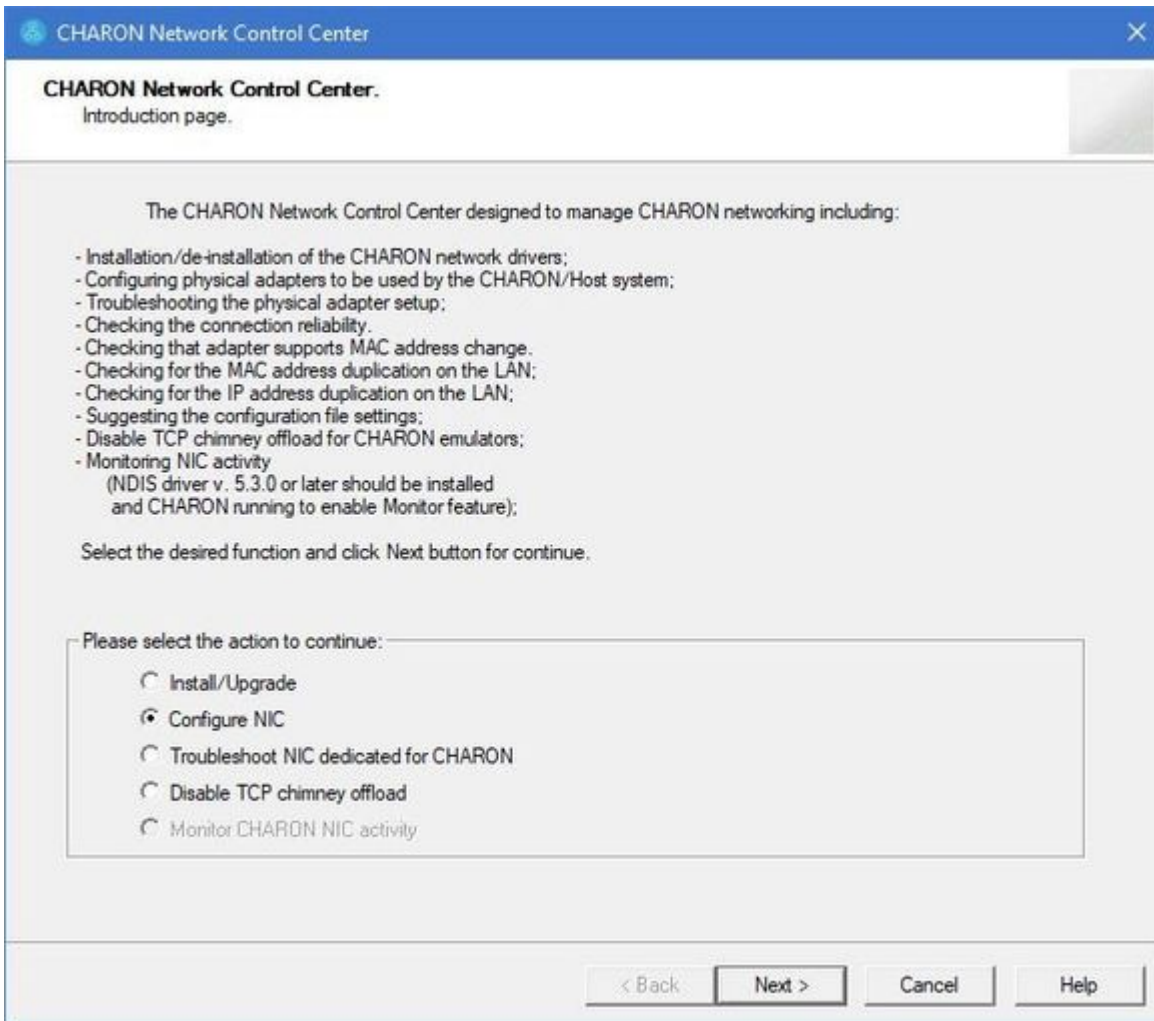
## Dedicating a Network Interface to Charon-VAX

If the CHARON virtual machine needs access to the network, one or more dedicated network interfaces cleared from any other protocols including TCP/IP will have to be defined.

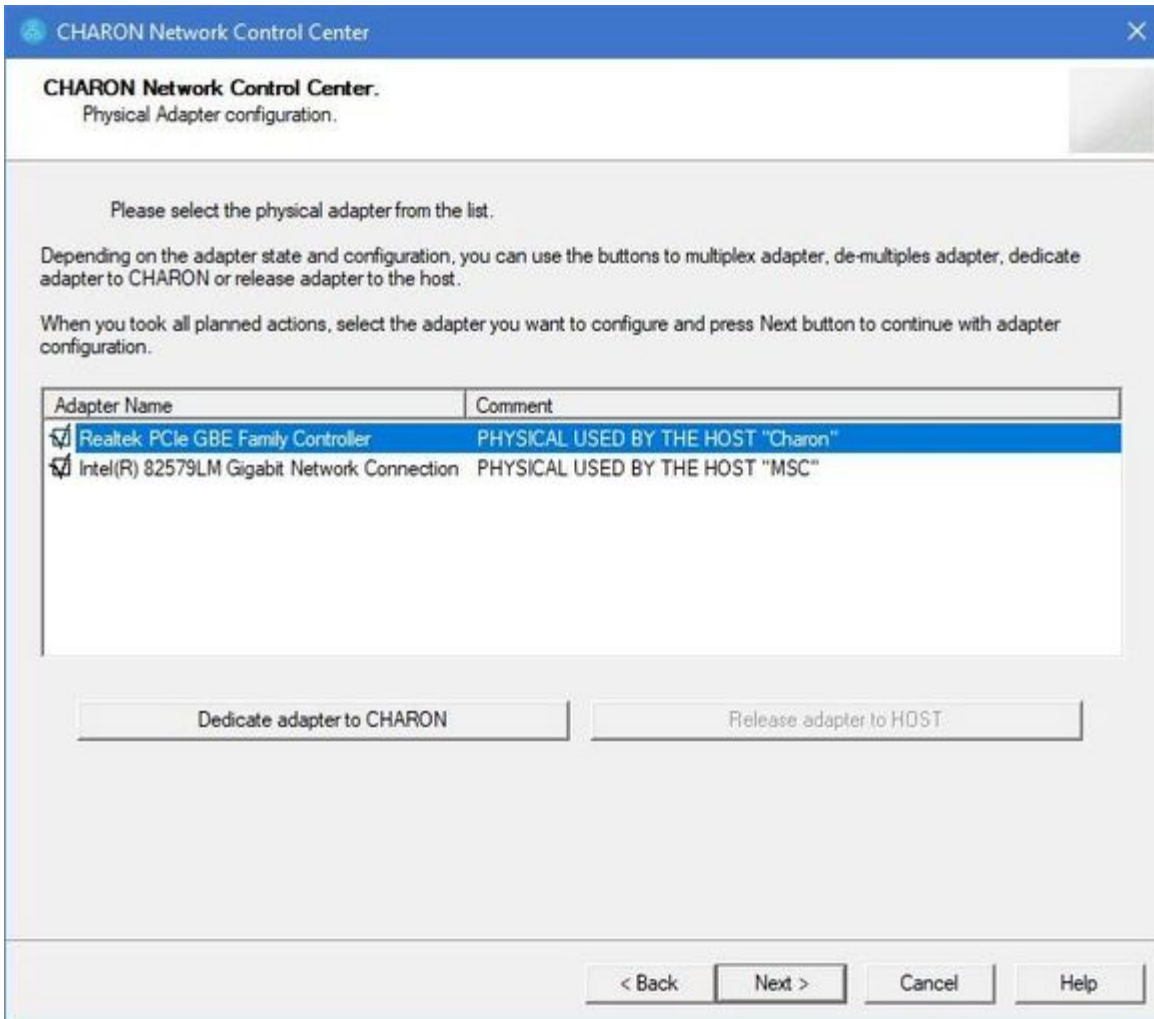
Configuring the network is done with the "Network Control Center" (NCC) utility. Double click the "CHARON Virtual Machine Manager" icon on the desktop or select this utility in the tray menu; select the "Host Information and Utilities" section and press the "Network Control Center" button:



In the appearing dialog, select "Configure NIC":



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

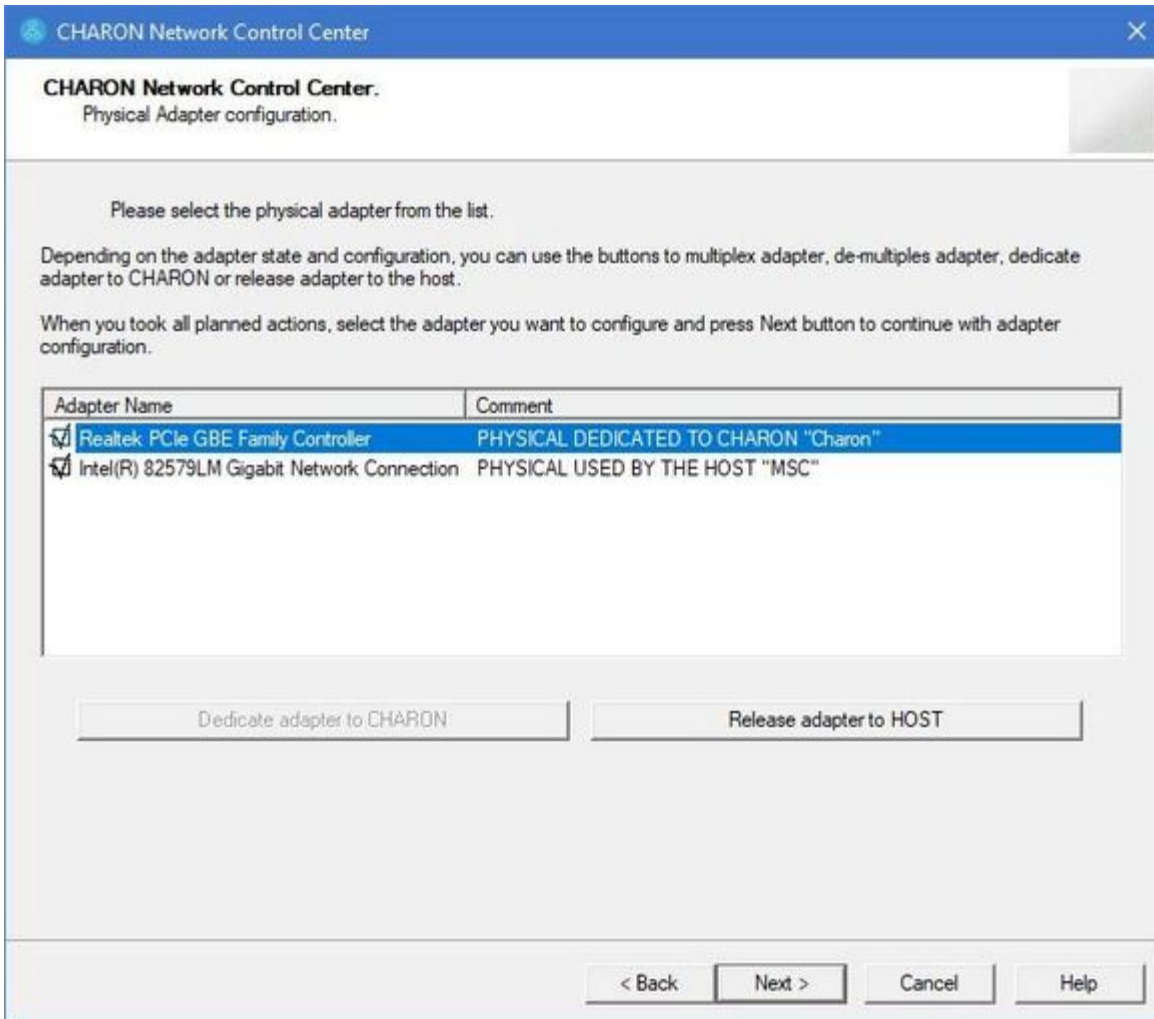


Select the interface to be dedicated to CHARON (in our example it is Realtek PCIe GBE Family Controller named "Charon") and press the "Dedicate adapter to CHARON" button.

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 configuring a regular network adapter and a VLAN one, so all the provided instructions are fully applicable for VLAN adapters as well.



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



**i** It is possible to release the interface back to the CHARON host by selecting the target interface and pressing the "Release adapter to HOST" button.

The next step displays the 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:

**CHARON Network Control Center.**  
Proposals for configuration file records.

The CHARON Network Control Center Config File suggestion step.

Please select the network interface you plan to use (depending on emulated model) and check the suggested configuration file settings with one you are using. The settings are suggested for primary adapter only.

For secondary adapter configuration and adapter options (see the your system's 'User Guide' for details).

QBUS systems (MicroVAX 3600, MicroVAX 4000 106/108, ...) - XQA	<input type="radio"/> DEQNA	<input type="radio"/> DELQA	<input type="radio"/> DESQA
SCSI systems (MicroVAX 3100 96/98, MicroVAX 4000 106/108, ...) - EZA	<input type="radio"/> SGEC		
XMI systems (VAX 66x0) - EXA	<input type="radio"/> DEMNA		
<b>CHARON-AXP (AlphaServer DS10, DS20, ES40, GS80, GS160, ...) - EWA</b>	<input checked="" type="radio"/> DE500AA	<input type="radio"/> DE500BA	<input type="radio"/> DE435
CHARON-AXP (AlphaServer DS10, DS20, ES40, GS80, GS160, ...) - EIA	<input type="radio"/> DE602		
DEBNI systems (VAX 81 or VAX 6000) - DEBNI	<input type="radio"/> DEBNI		
PMAD-AA (TurboChannel add-on card for VAX 4000-90A)	<input type="radio"/> PMADAA		

NIC connection name.  
load DE500AA/dec21x4x EWA interface=EWA0  
load packet\_port/chnetwk EWA0 interface="connection:Charon"

Copy suggestions to clipboard

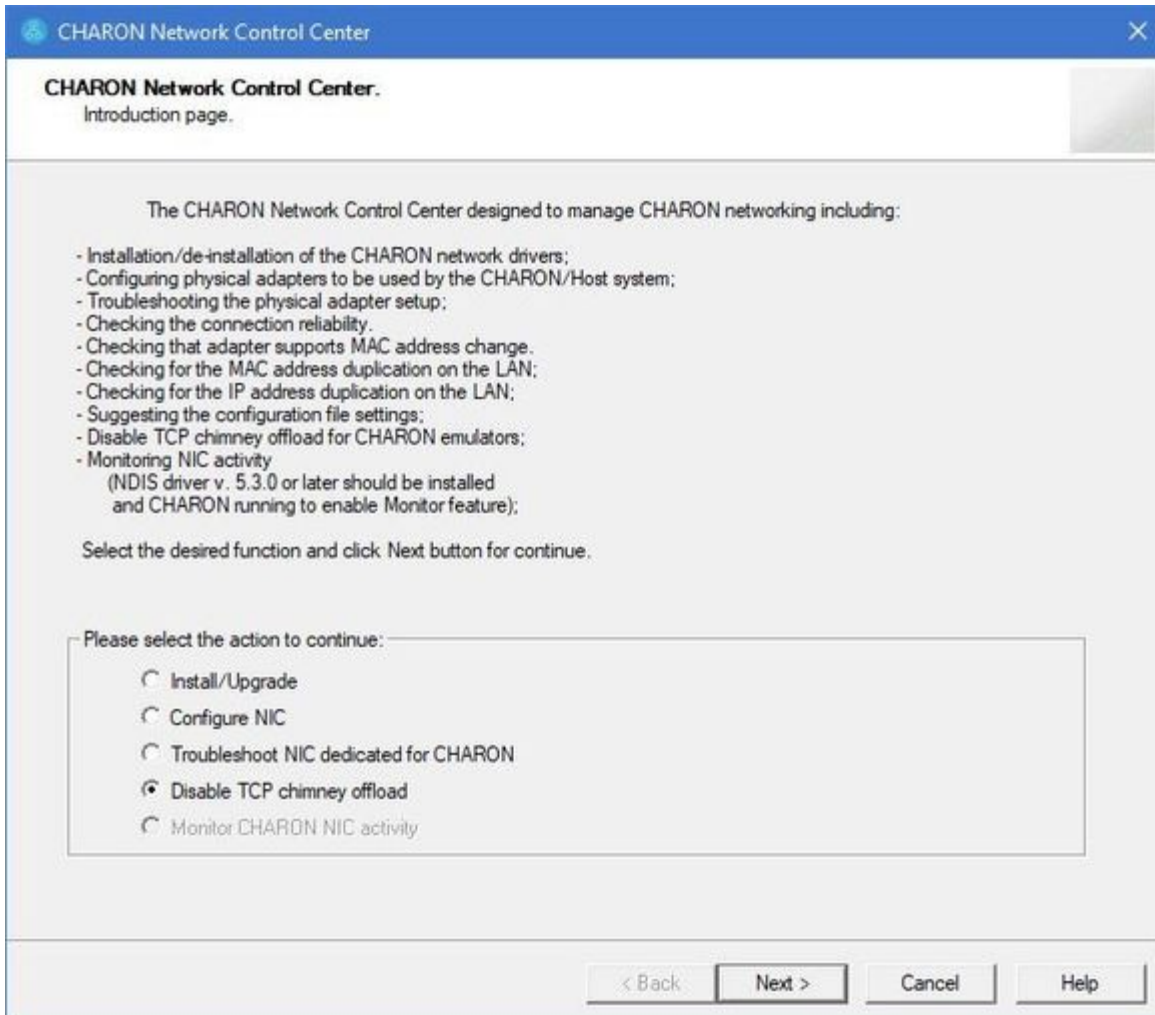
< Back   Next >   Cancel   Help

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-VAX / CHARON-PDP configuration file during the next editing session by pressing "Ctrl-V".

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

## Disabling TCP Chimney Offloading

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.

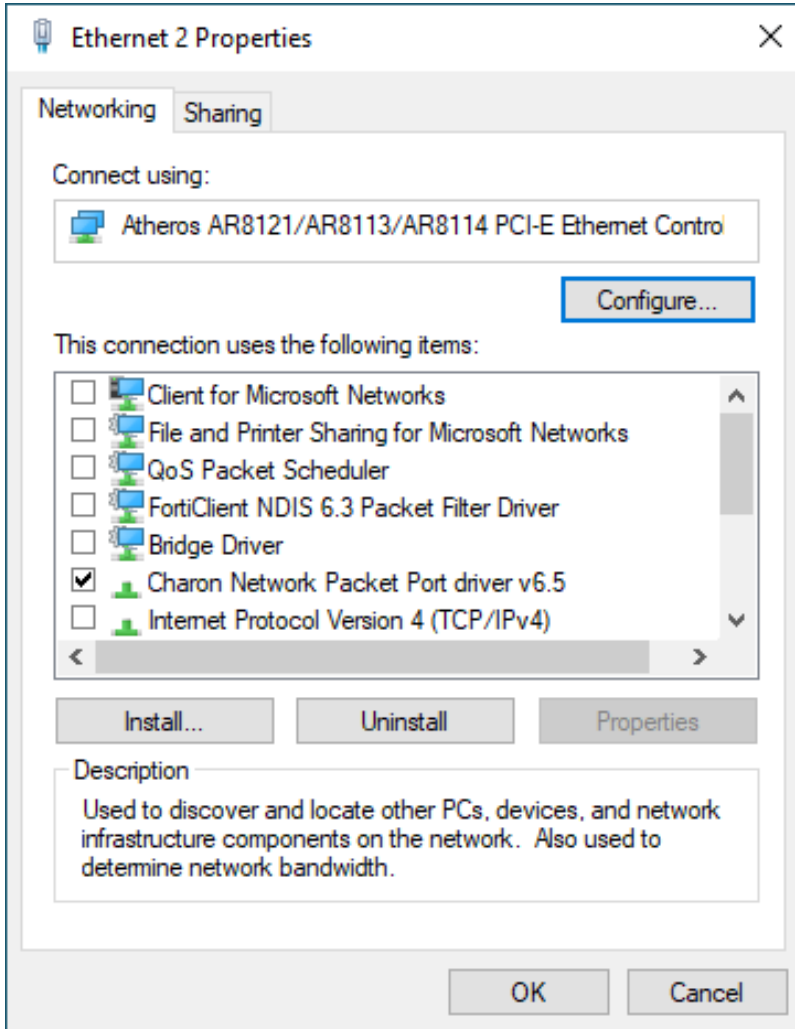
Press the "Cancel" button to exit from the "Network Control Center" utility.

**⚠** 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.

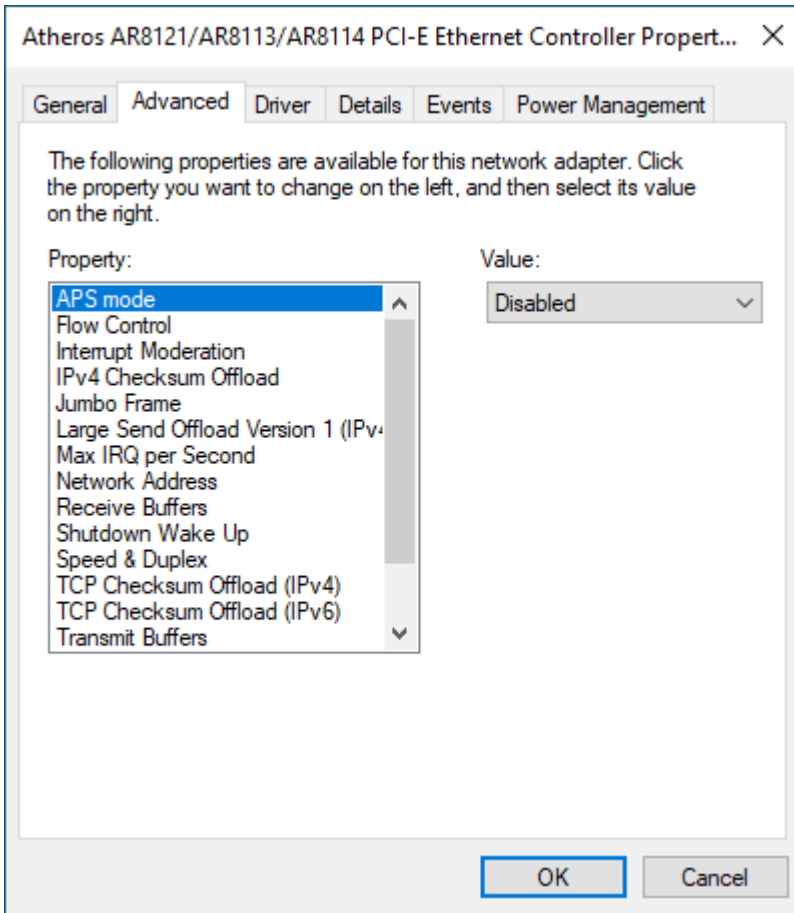
## Disable Network Interface Attributes Manually

For every Network Interface (NIC) that you will use with Charon-VAX, you should manually disable various offloading functions in the Windows driver. The following description applies to Windows 10 Pro. It may differ somewhat for other Windows versions.

Open the "Settings" dialog, select "Network & Internet", then "Change Adapter Options". A new dialog window will appear showing all the NICs known to the system. For all NICs that will be used by Charon-VAX, right-click on the NIC and choose "Properties". The following dialog should appear:



Click the "Configure" button to open. The following dialog window should appear. Select the "Advanced" tab.



Go through every "Property" and look at the options available in the "Value" drop-down. The properties that you see here will differ depending on the networking hardware, the network driver, and the Windows version, so please keep in mind that the specific examples mentioned below are exactly that – examples. If the value "disable" is offered, you should select that. For properties with numeric values, you should not change the value.

The following are examples of properties that must be disabled:

- Adaptive Inter-Frame Spacing
- Interrupt Moderation
- IPv4 Checksum Offload
- Jumbo Packet
- Large Send Offload V1 or V2 (IPv4 & IPv6)
- Packet Priority & VLAN
- Receive Side Scaling
- TCP Checksum Offload (IPv4 & IPv6)
- UDP Checksum Offload (IPv4 & IPv6)

Examples of properties that should not be changed:

- Flow Control
- Locally Administered Address
- Maximum Number of RSS Queues
- Receive Buffers
- Transmit Buffers

## After installation

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

**Please note:** if the CHARON-VAX / CHARON-PDP user belongs to an AD domain, you have to add the user to the CHARON-GRP group as it is described in [this article](#) and then reboot the CHARON host

Otherwise it is possible to use a standard account (both local and domain) for running CHARON-VAX / CHARON-PDP:

1. Login as the local system administrator ("Administrator") on the host system.
2. Create a special user for running CHARON-VAX / CHARON-PDP. This user must have administrative or standard privileges (in this case login as Administrator for manging NDIS drivers and running NCC utility). Please consult with Windows [online documentation](#) for 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.

# Running and managing CHARON-VAX and CHARON-PDP for Windows

## Table of Contents

- Starting CHARON Virtual Machine Manager
- Creating a CHARON Virtual Machine
  - Creating a CHARON VM using provided templates
  - Creating a CHARON VM using an existing configuration file
- Running a CHARON Virtual Machine
- Stopping a CHARON Virtual Machine
- Removing a CHARON Virtual Machine
- Modifying a CHARON Virtual Machine
  - Exploring CHARON Virtual Machine Home Directory
  - Making a new virtual disk
  - CHARON Virtual Machine configuration file
  - CHARON Virtual Machine startup mode
  - CHARON version to be used for CHARON VM
- Service functions
- Running CHARON utilities

## Starting CHARON Virtual Machine Manager

CHARON execution is controlled by the "CHARON Virtual Machine Manager" that manages each CHARON instance, called a "CHARON Virtual Machine", running on the host.

The CHARON installation procedure creates a shortcut on the desktop and under the Windows Start menu for the CHARON Virtual Machine Manager.

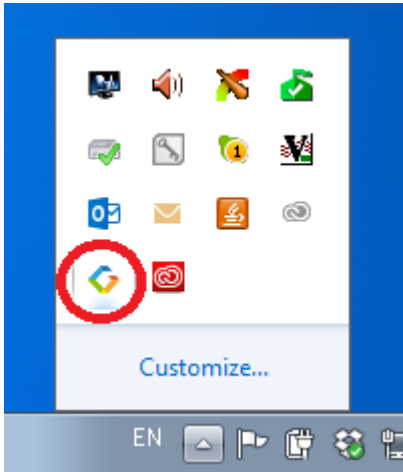
Desktop shortcut:



"Start" menu shortcut:



The CHARON Virtual Machine Manager runs automatically at Windows startup. If it is minimized to tray, it can be re-opened in the following way:



After clicking the shortcuts, the main dialog of CHARON Virtual Machine Manager appears.

## Creating a CHARON Virtual Machine

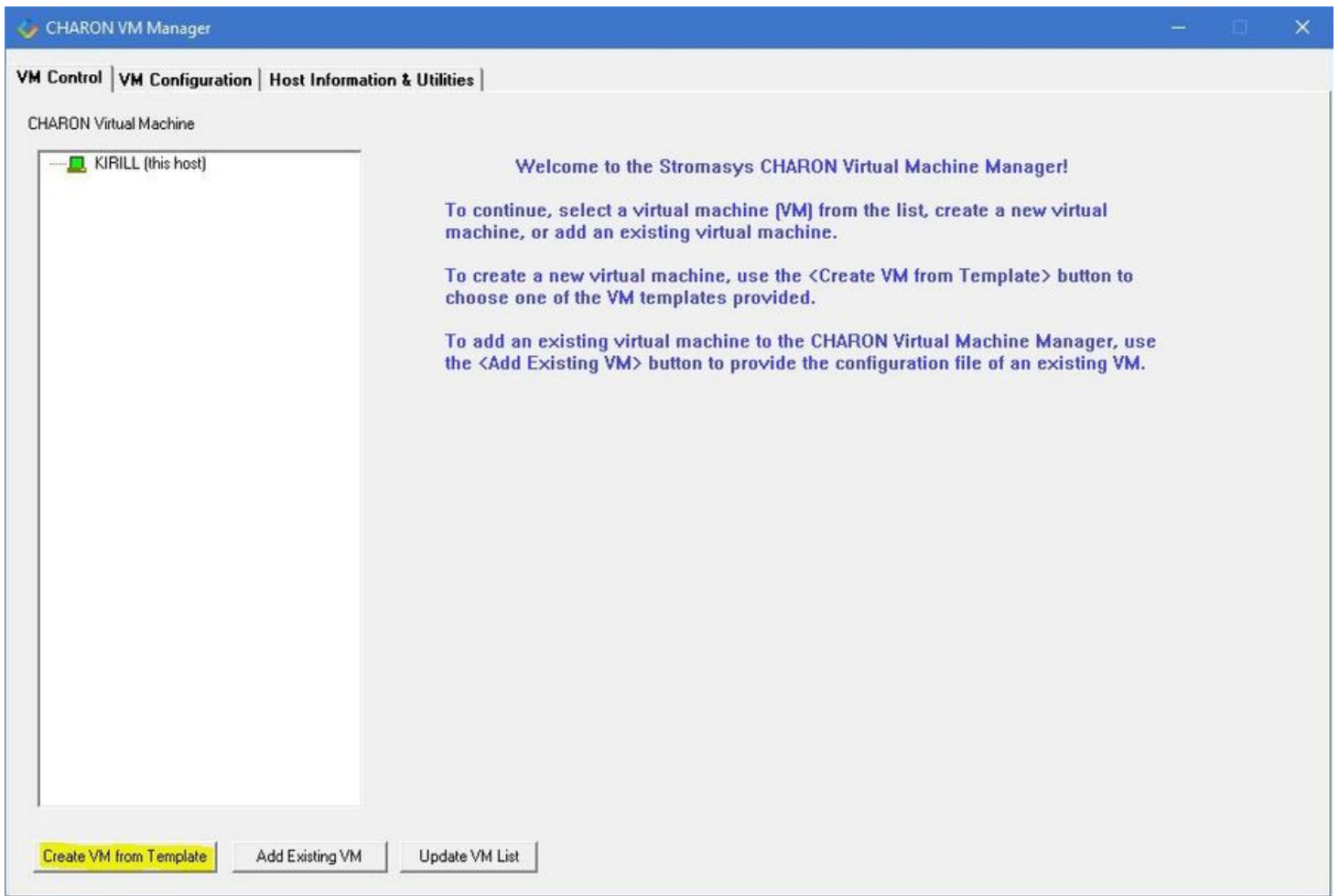
### Creating a CHARON VM using provided templates

CHARON installation contains a set of template files for creating a base configuration for each VAX/PDP11 model. Once the CHARON Virtual Machine (VM) is created using these templates it can be configured further to meet your requirements and all the specifics of the emulated system.

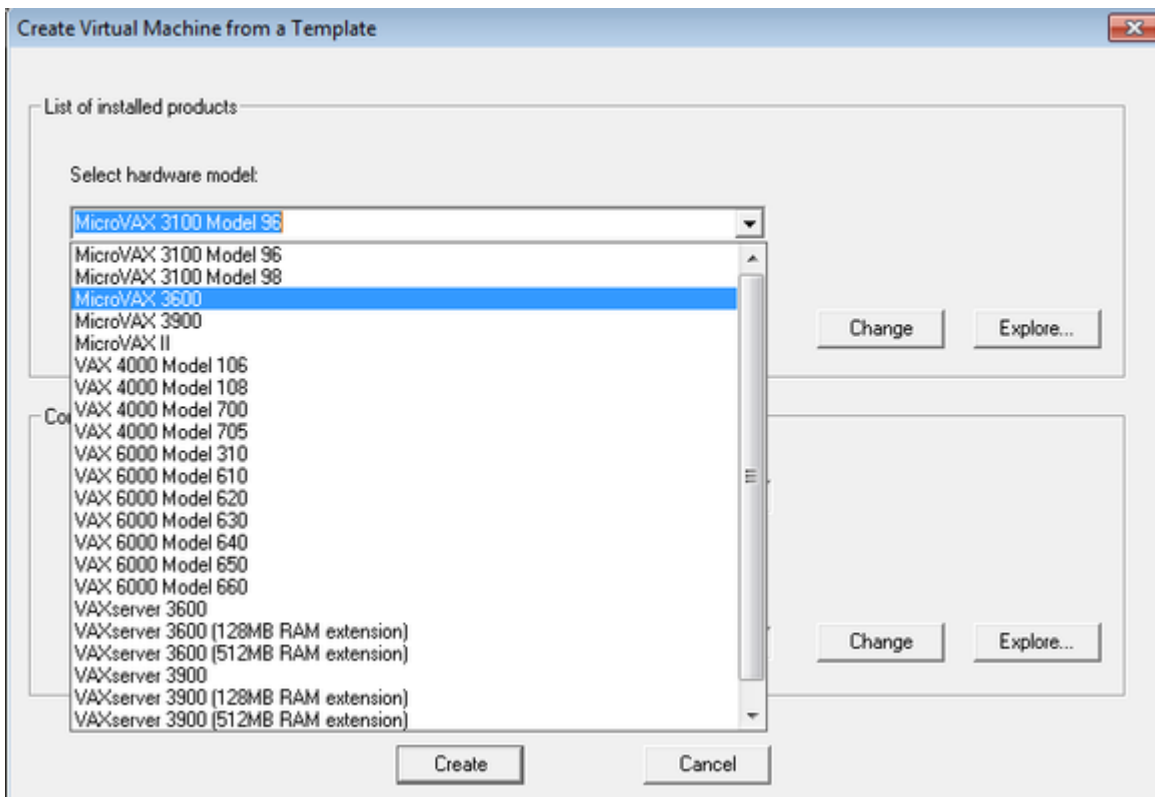
The CHARON templates do not contain any specifications for disks and network; you must update the configuration file created from the template to specify your disk and network configuration.

Start the CHARON Virtual Machine Manager (see above) and press the "Create VM from Template" button:



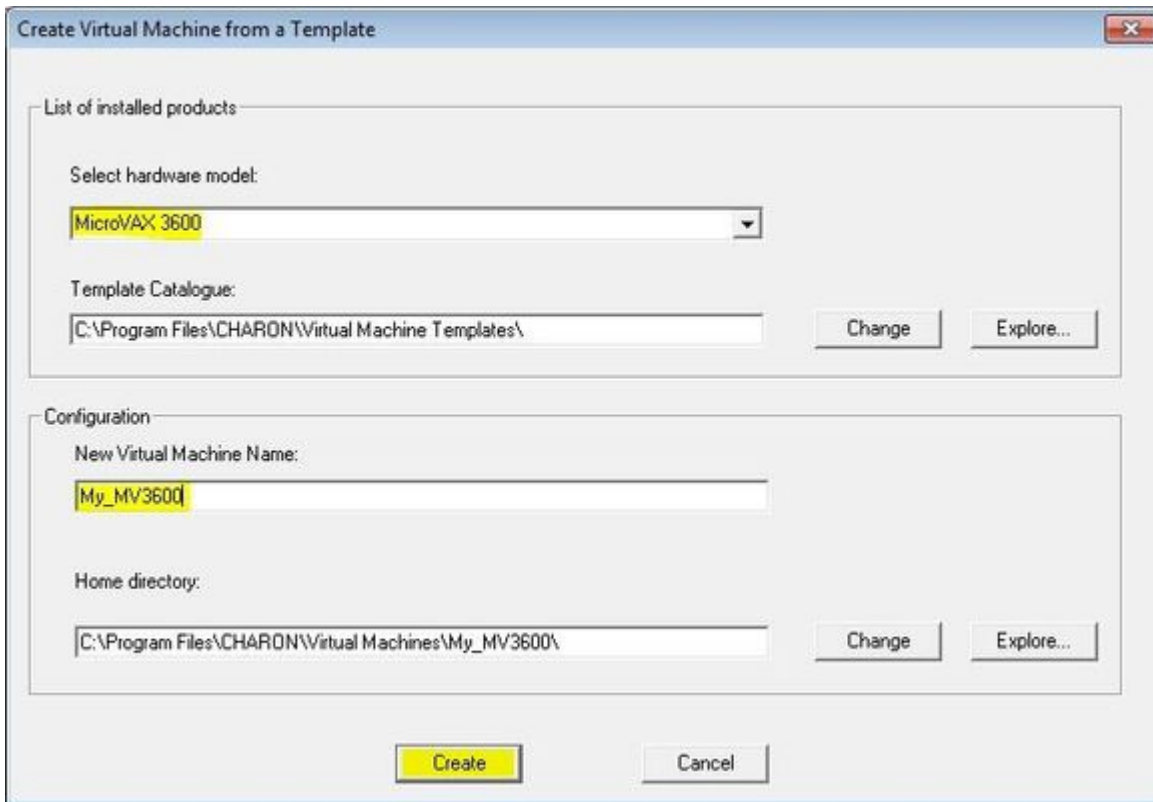


In the dialog that appears select the VAX/PDP11 hardware model:

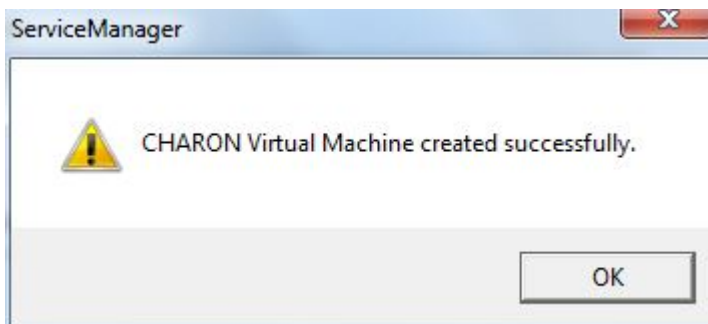


Note the Home Directory for the VM (see screenshot below), it will be used by default for storing logs, virtual disk images, toy and container files if no path is specified in the configuration file.

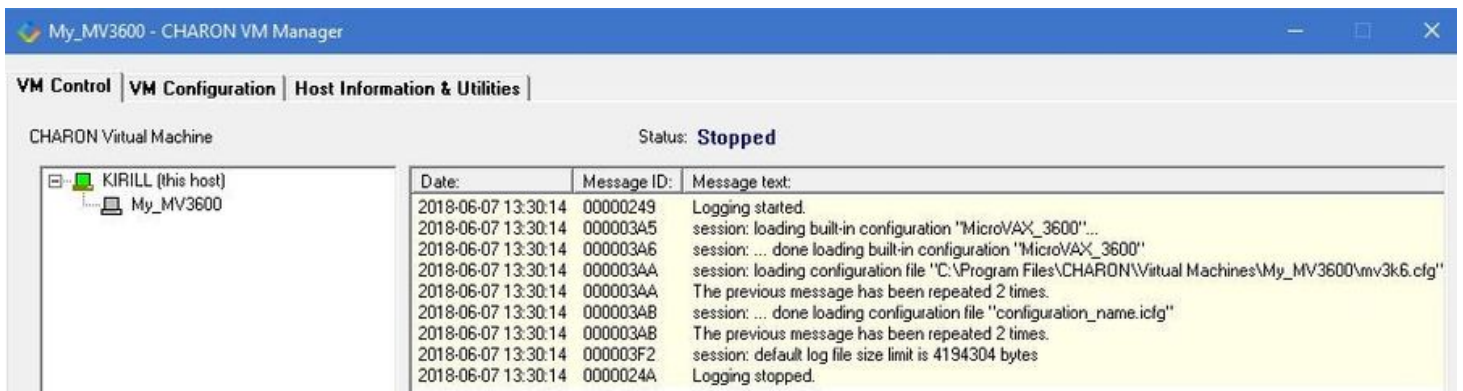
Enter a name for the VM and press the "Create" button:



The CHARON Virtual Machine Manager will report the VM has been created successfully:



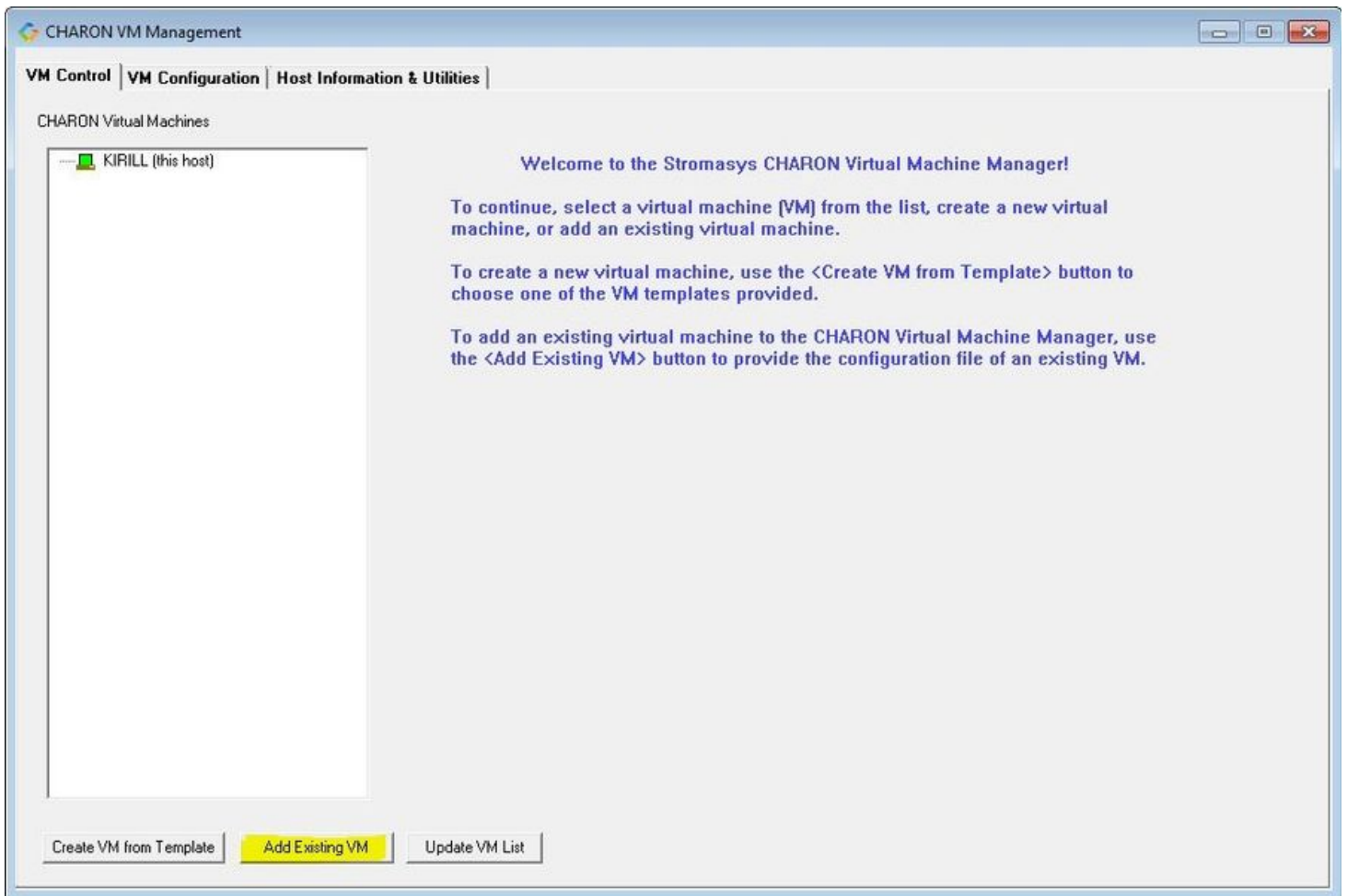
It will then appear in the left pane:



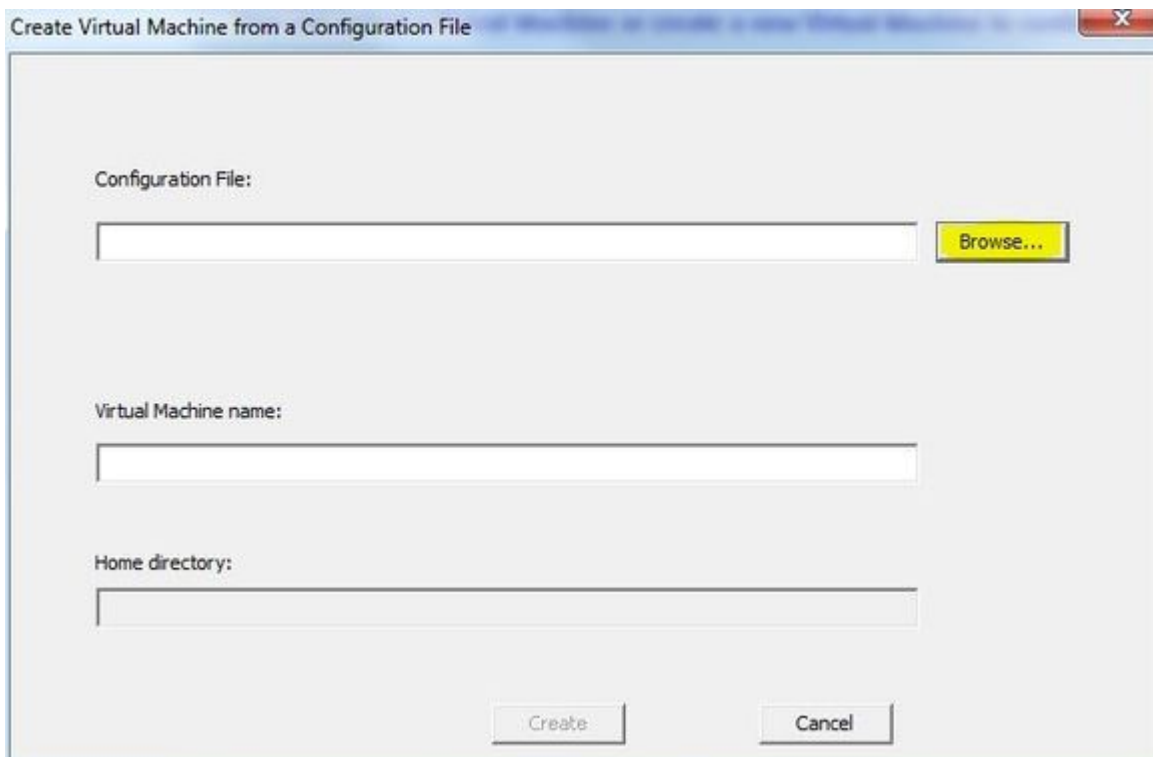
## Creating a CHARON VM using an existing configuration file

It is also possible to use an existing configuration file for creating a CHARON Virtual Machine.

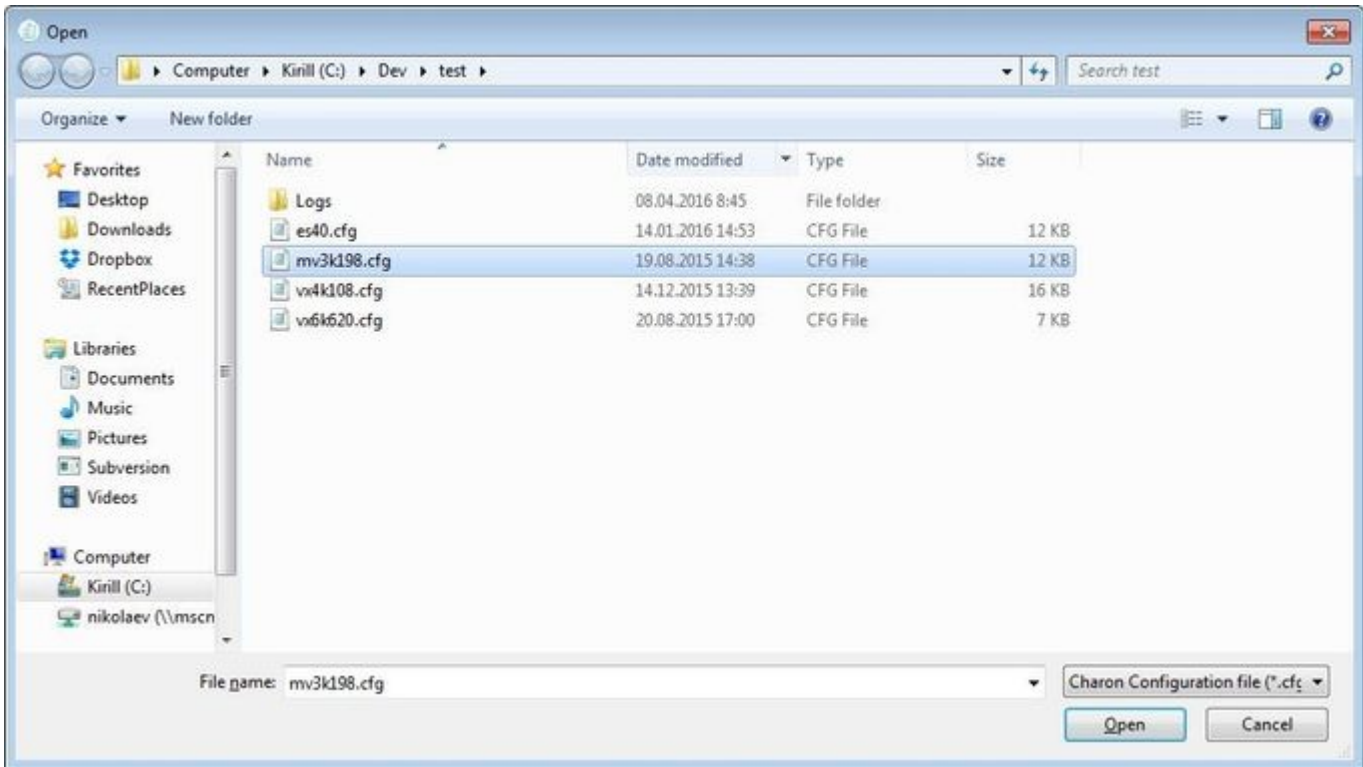
Open the CHARON Virtual Machine Manager (see above) and press the "Add Existing VM" button:



In the dialog that appears select the configuration file to import by pressing the "Browse" button:



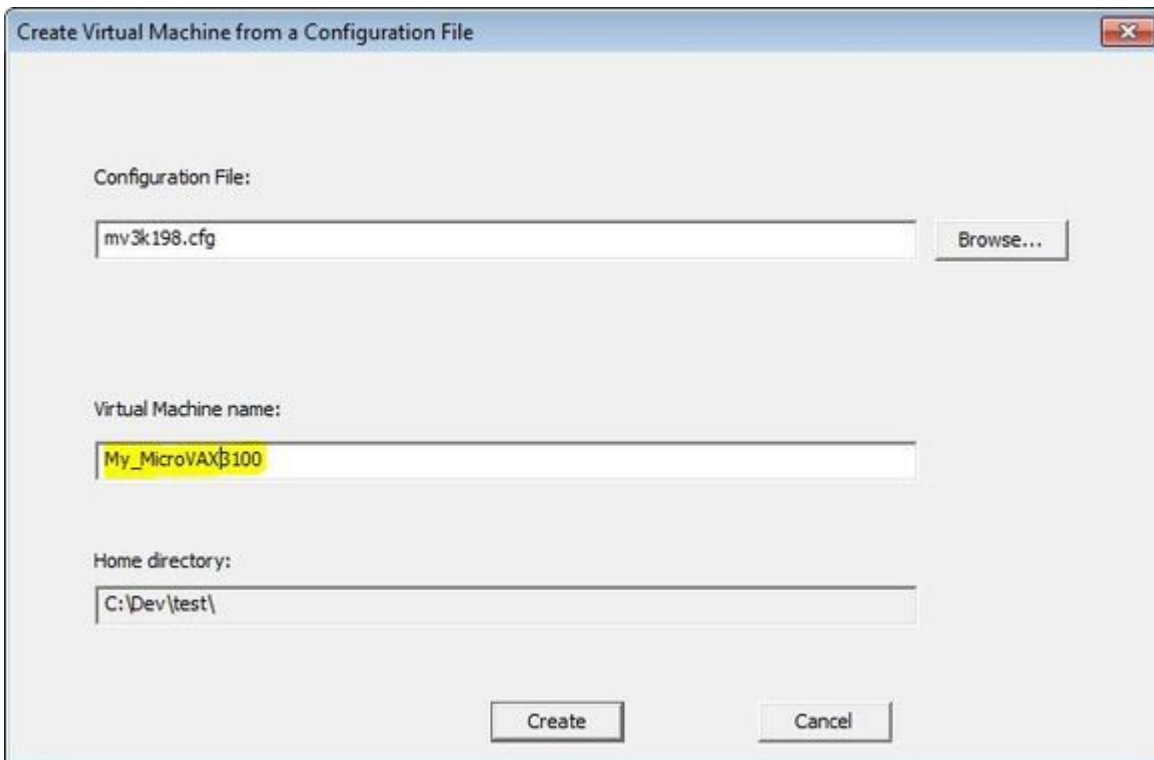
Select the configuration file and press the "Open" button:



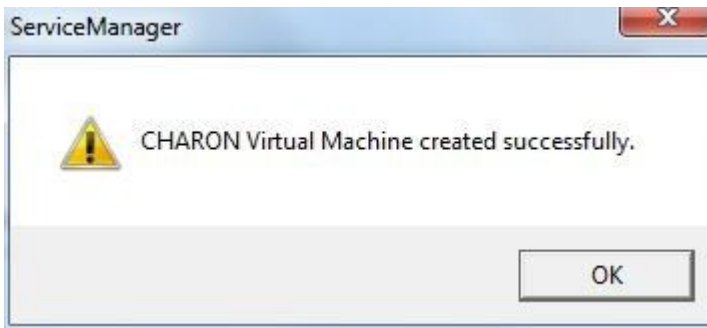
Never select a configuration file template provided in the Charon installation folder for editing as these files could be overwritten with patching

Enter the CHARON Virtual Machine name and note the Home Directory for this VM, it will be used by default for storing logs, virtual disk images, toy and container files if no path is specified in the configuration file.

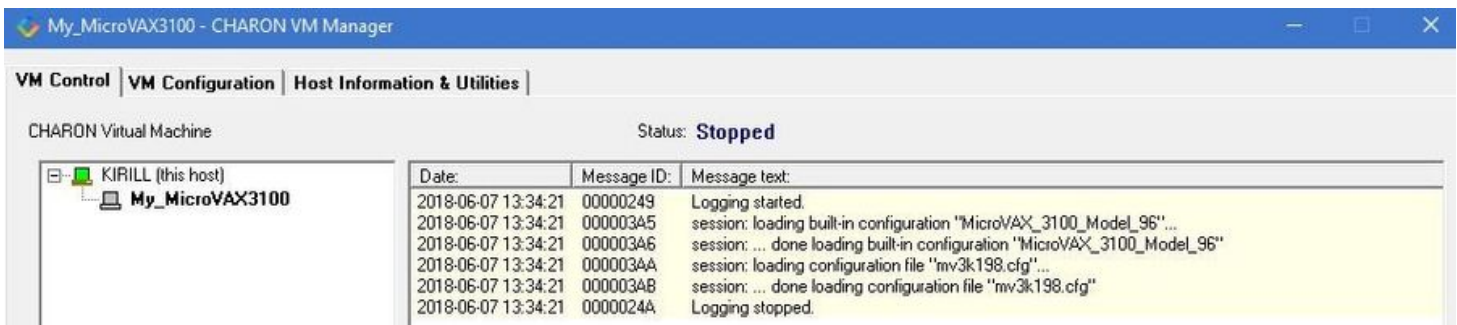
Press the "Create" button:



The CHARON Virtual Machine Manager will report that the VM has been created successfully:

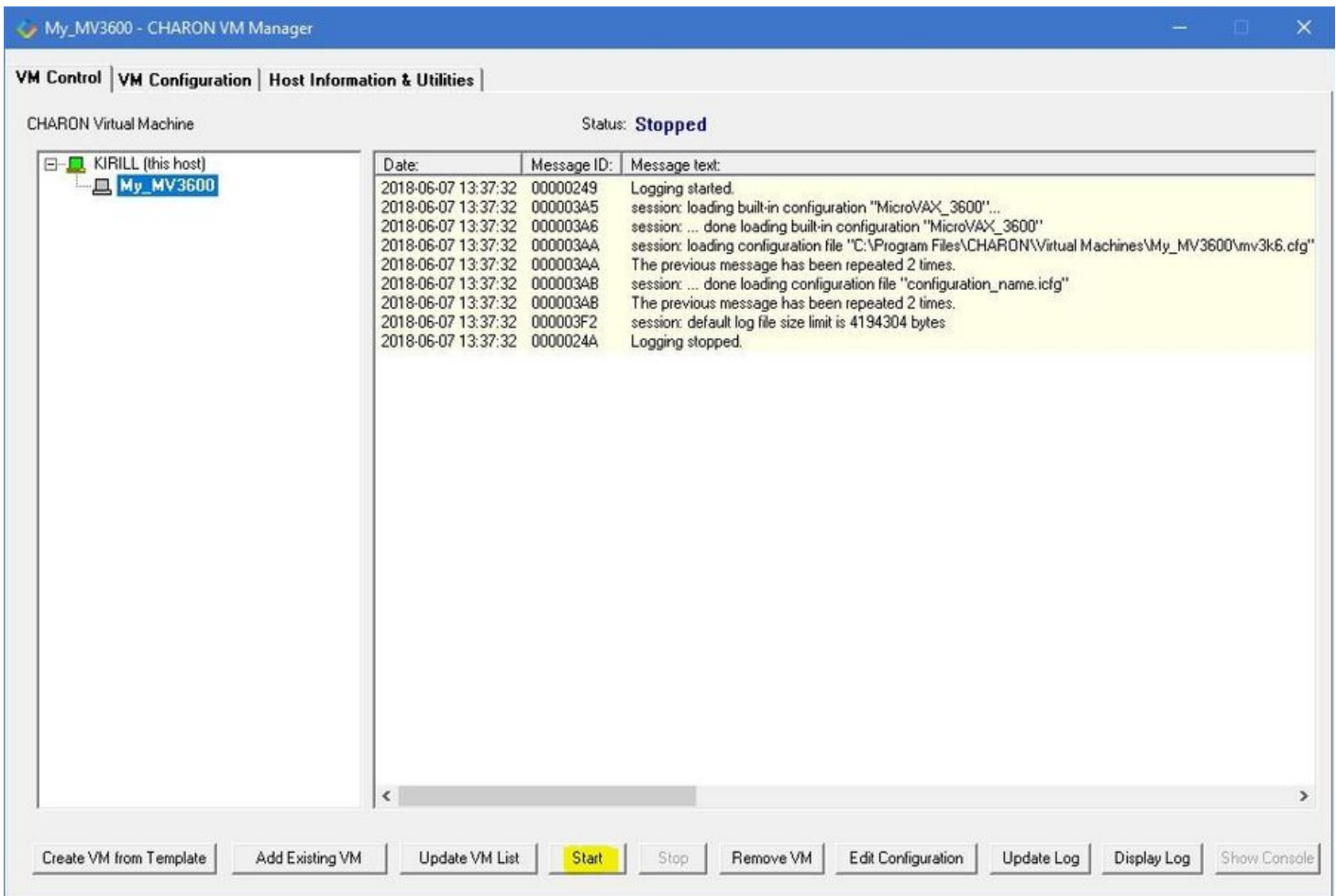


It will then appear in the left pane:



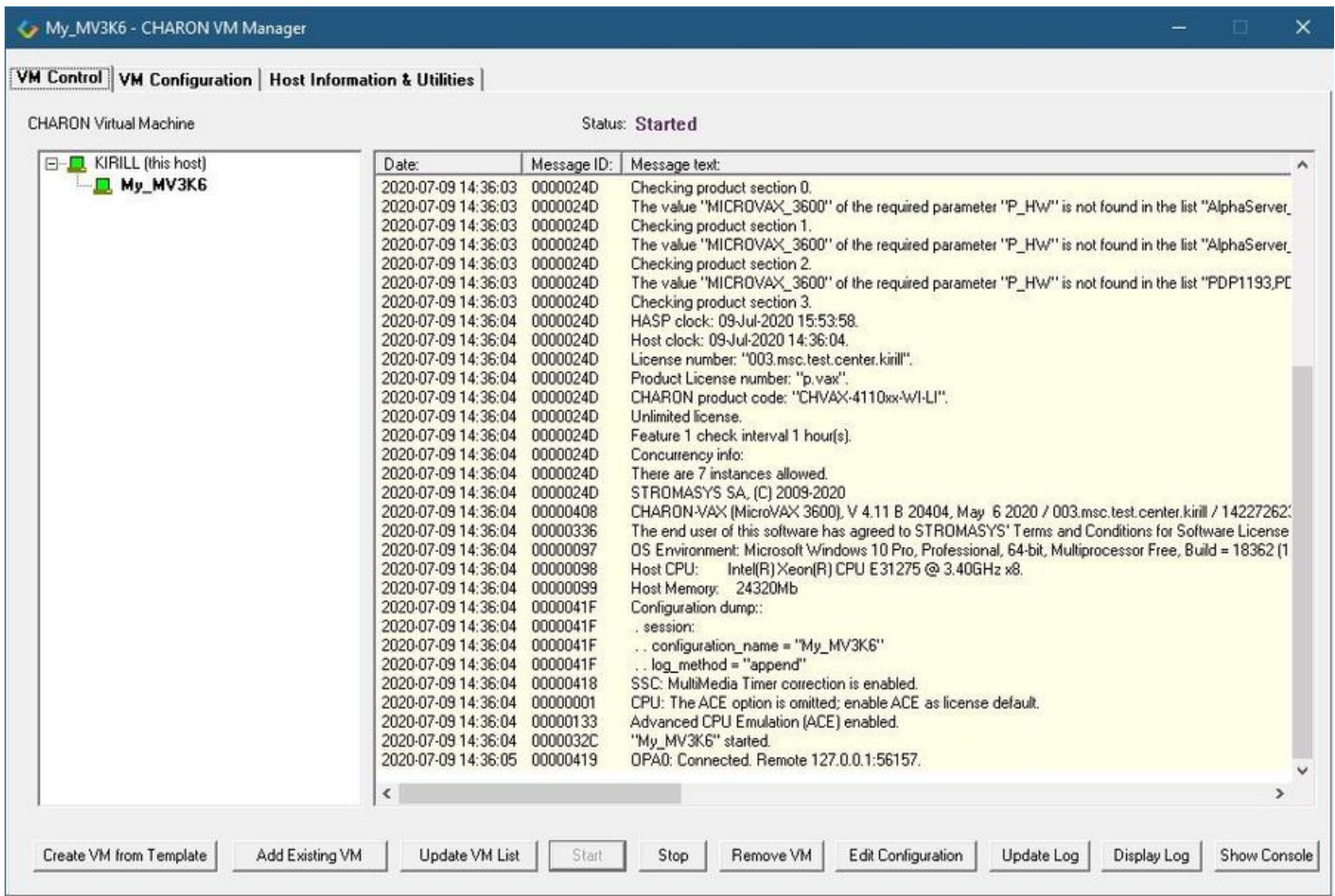
## Running a CHARON Virtual Machine

Select the target CHARON VM in the left pane and press the "Start" button:





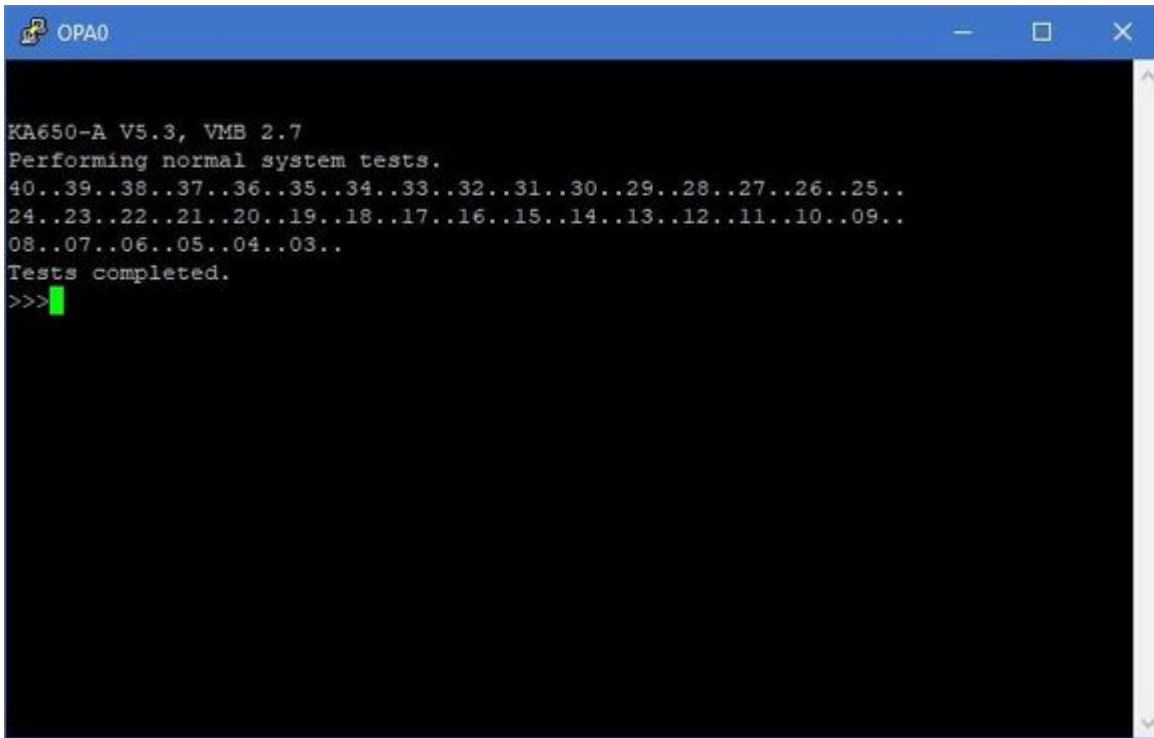
CHARON Virtual Machine Manager will display the execution log:



**i** Note the color meaning:

Colors	Meaning	Example
Red	Error	This is an error message
Light blue	Warning	This is a warning message
Light yellow	Information	This is an informational message
Purple	Begin/End of the Windows system log	Begin / End

The console of this particular CHARON VM will then appear (if defined in the configuration file):

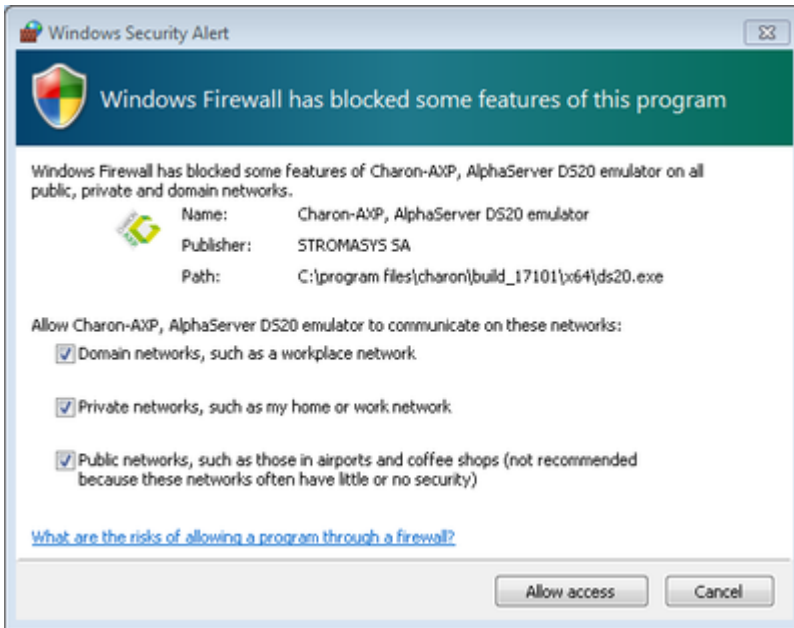


```

OPA0
KA650-A V5.3, VMB 2.7
Performing normal system tests.
40..39..38..37..36..35..34..33..32..31..30..29..28..27..26..25..
24..23..22..21..20..19..18..17..16..15..14..13..12..11..10..09..
08..07..06..05..04..03..
Tests completed.
>>>

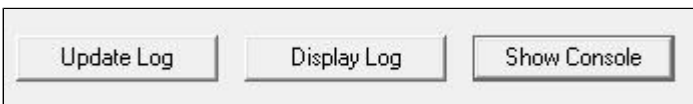
```

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



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

If you closed the CHARON VM console, it is always possible to re-open it by pressing the "Show Console" button:



**Closing the console will not stop the CHARON Virtual Machine.**

## Stopping a CHARON Virtual Machine

**!** Note: before stopping the virtual machine from the CHARON Virtual Machine Manager, a clean shutdown of the operating system running on the VM has to be performed.

Select the target CHARON VM in the left pane and press the "Stop" button. A dialog will appear to confirm, select "Yes" to continue to stop the VM:

The screenshot shows the CHARON VM Manager window titled "My\_MV3K6 - CHARON VM Manager". The interface includes tabs for "VM Control", "VM Configuration", and "Host Information & Utilities". The "VM Control" tab is active, showing the "CHARON Virtual Machine" section with a tree view on the left containing "KIRILL (this host)" and "My\_MV3K6". The "Status" is "Started". A log window on the right displays the following messages:

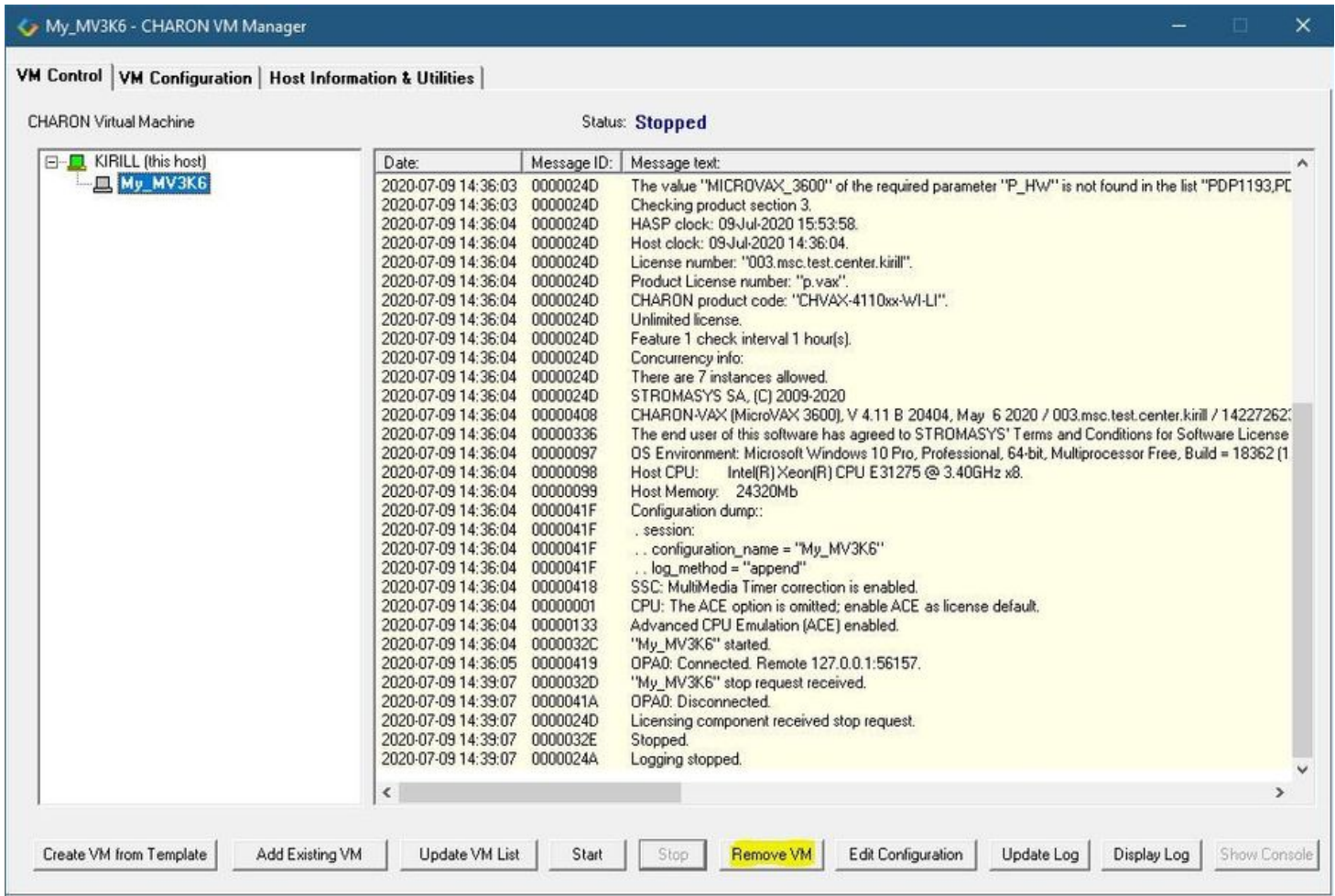
Date:	Message ID:	Message text:
2020-07-09 14:36:03	0000024D	Checking product section 0.
2020-07-09 14:36:03	0000024D	The value "MICROVAX_3600" of the required parameter "P_HW" is not found in the list "AlphaServer_
2020-07-09 14:36:03	0000024D	Checking product section 1.
2020-07-09 14:36:03	0000024D	The value "MICROVAX_3600" of the required parameter "P_HW" is not found in the list "AlphaServer_
2020-07-09 14:36:03	0000024D	Checking product section 2.
2020-07-09 14:36:03	0000024D	The value "MICROVAX_3600" of the required parameter "P_HW" is not found in the list "PDP1193,PC
2020-07-09 14:36:03	0000024D	Checking product section 3.
2020-07-09 14:36:04	0000024D	HASP clock: 09-Jul-2020 15:53:58.
2020-07-09 14:36:04	0000024D	Host clock: 09-Jul-2020 14:36:04.
2020-07-09 14:36:04	0000024D	License number: "003.msc.test.center.kirill".
2020-07-09 14:36:04	0000024D	Product License number: "p.vax".
2020-07-09 14:36:04	0000024D	CHARON product code: "CHVAX-4110xx-WI-LI".
2020-07-09 14:36:04	0000024D	Unlimited license.
2020-07-09 14:36:04	0000024D	Feature 1 check interval 1 hour(s).
2020-07-09 14:36:04	0000024D	Concurrency info:
2020-07-09 14:36:04	0000024D	There are 7 instances allowed.
2020-07-09 14:36:04	0000024D	STROMASYS SA, (C) 2009-2020
2020-07-09 14:36:04	00000408	CHARON-VAX (MicroVAX 3600), V 4.11 B 20404, May 6 2020 / 003.msc.test.center.kirill / 14227262:
2020-07-09 14:36:04	00000336	The end user of this software has agreed to STROMASYS' Terms and Conditions for Software License
2020-07-09 14:36:04	00000097	OS Environment: Microsoft Windows 10 Pro, Professional, 64-bit, Multiprocessor Free, Build = 18362 [1
2020-07-09 14:36:04	00000098	Host CPU: Intel(R) Xeon(R) CPU E31275 @ 3.40GHz x8.
2020-07-09 14:36:04	00000099	Host Memory: 24320Mb
2020-07-09 14:36:04	0000041F	Configuration dump::
2020-07-09 14:36:04	0000041F	. session:
2020-07-09 14:36:04	0000041F	.. configuration_name = "My_MV3K6"
2020-07-09 14:36:04	0000041F	.. log_method = "append"
2020-07-09 14:36:04	00000418	SSC: MultiMedia Timer correction is enabled.
2020-07-09 14:36:04	00000001	CPU: The ACE option is omitted; enable ACE as license default.
2020-07-09 14:36:04	00000133	Advanced CPU Emulation (ACE) enabled.
2020-07-09 14:36:04	0000032C	"My_MV3K6" started.
2020-07-09 14:36:05	00000419	OPAO: Connected. Remote 127.0.0.1:56157.

At the bottom of the window, there is a toolbar with buttons: "Create VM from Template", "Add Existing VM", "Update VM List", "Start", "Stop" (highlighted in yellow), "Remove VM", "Edit Configuration", "Update Log", "Display Log", and "Show Console".

## Removing a CHARON Virtual Machine

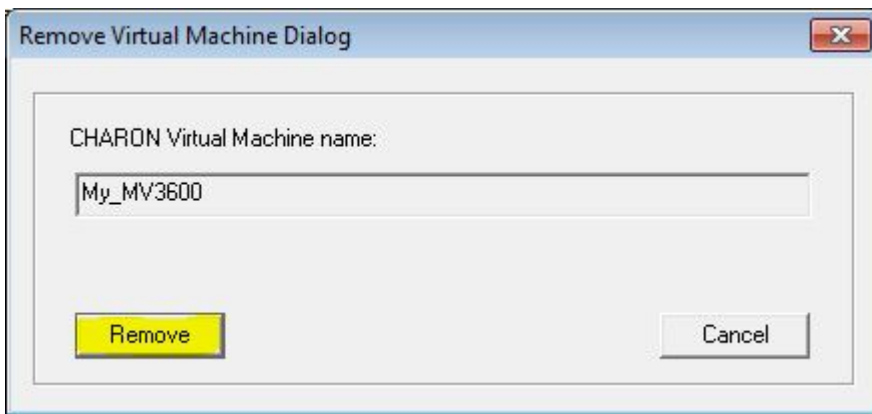
To remove a CHARON VM, select it and press the "Remove VM" button:





⚠ The Virtual machine must be stopped before removing it (see [Running and managing CHARON-VAX and CHARON-PDP for Windows#Stopping a CHARON Virtual Machine](#) chapter above)

A confirmation dialog will be displayed:



Press the "Remove" button and confirm the CHARON VM removal by pressing the "Yes" button:



## Modifying a CHARON Virtual Machine

Select the target CHARON VM and switch to the "VM Configuration" tab to see its features:

The screenshot shows the 'VM Configuration' tab in the CHARON VM Manager. The window title is 'My\_MV3K6 - CHARON VM Manager'. The 'VM Configuration' tab is selected, with other tabs being 'VM Control' and 'Host Information & Utilities'. The main content area contains the following text and controls:

**The Home directory contains the virtual machine configuration file and is the default location for the following components of this virtual machine:**

- Virtual machine log file(s);
- Virtual machine ROM, VDisk(s) and other data.

Home directory:

**Press the <Edit Configuration File> button to edit the configuration file using the Notepad editor. Modify the configuration to accurately reflect the features of the system to be emulated. Press the <Apply> button to confirm the virtual machine configuration file changes.**

Configuration file:

**Services are set to manual start by default. Automatic start at boot can be enabled after successfully testing the configuration. You can change this setting below.**

Startup type:

**By default, virtual machine is created with the latest CHARON executable image version installed on the system. To change the version of the executable image for the VM: choose the desired executable image from the list box below, press the <Apply> button, restart the VM.**

Executable:

Using this dialog it is possible to:

- Explore the Home Directory of the CHARON VM.
- Create virtual disks by pressing the "Create Virtual Disks" button.
- Edit the CHARON VM configuration file.
- Change the startup mode.
- Select the CHARON version used for this Virtual Machine.

## Exploring CHARON Virtual Machine Home Directory

Press the "Explore" button to open up a Windows Explorer window showing the content of the selected CHARON VM home folder:

This screenshot is identical to the previous one, but the 'Explore...' button is highlighted in yellow, indicating it is the focus of the instruction.

**i** This content includes the CHARON VM configuration file and may also include virtual disks and tapes images, logs, ROM files, etc.

## Making a new virtual disk

It is very convenient to use the "Create Virtual Disk" button to create empty disk images in the CHARON VM Home Directory:

**The Home directory contains the virtual machine configuration file and is the default location for the following components of this virtual machine:**

- Virtual machine log file(s);
- Virtual machine ROM, VDisk(s) and other data.

Home directory:

The Mkdisk utility will be called for this operation:



Refer to the "Utilities" section of this User's Guide for more information about the [MkDisk](#) utility.

## CHARON Virtual Machine configuration file

Press the "Edit Configuration file" button:

**Press the <Edit Configuration File> button to edit the configuration file using the Notepad editor.  
Modify the configuration to accurately reflect the features of the system to be emulated.  
Press the <Apply> button to confirm the virtual machine configuration file changes.**

Configuration file:

Alternatively it is possible to select the target CHARON VM and press the "Edit Configuration" button:



The screenshot shows the CHARON VM Manager window with the 'Host Information & Utilities' tab selected. The VM 'My\_MV3K6' is shown as 'Stopped'. The log window displays the following messages:

Date:	Message ID:	Message text:
2020-07-09 14:36:03	0000024D	The value "MICROVAX_3600" of the required parameter "P_HW" is not found in the list "PDP1193.PC"
2020-07-09 14:36:03	0000024D	Checking product section 3.
2020-07-09 14:36:04	0000024D	HASP clock: 09-Jul-2020 15:53:58.
2020-07-09 14:36:04	0000024D	Host clock: 09-Jul-2020 14:36:04.
2020-07-09 14:36:04	0000024D	License number: "003.msc.test.center.kirill".
2020-07-09 14:36:04	0000024D	Product License number: "p.vax".
2020-07-09 14:36:04	0000024D	CHARON product code: "CHVAX-4110xx-WI-LI".
2020-07-09 14:36:04	0000024D	Unlimited license.
2020-07-09 14:36:04	0000024D	Feature 1 check interval 1 hour(s).
2020-07-09 14:36:04	0000024D	Concurrency info:
2020-07-09 14:36:04	0000024D	There are 7 instances allowed.
2020-07-09 14:36:04	0000024D	STROMASYS SA, (C) 2009-2020
2020-07-09 14:36:04	00000408	CHARON-VAX (MicroVAX 3600), V 4.11 B 20404, May 6 2020 / 003.msc.test.center.kirill / 14227262:
2020-07-09 14:36:04	00000336	The end user of this software has agreed to STROMASYS' Terms and Conditions for Software License
2020-07-09 14:36:04	00000097	OS Environment: Microsoft Windows 10 Pro, Professional, 64-bit, Multiprocessor Free, Build = 18362 (1
2020-07-09 14:36:04	00000098	Host CPU: Intel(R) Xeon(R) CPU E31275 @ 3.40GHz x8.
2020-07-09 14:36:04	00000099	Host Memory: 24320Mb
2020-07-09 14:36:04	0000041F	Configuration dump:
2020-07-09 14:36:04	0000041F	. session:
2020-07-09 14:36:04	0000041F	... configuration_name = "My_MV3K6"
2020-07-09 14:36:04	0000041F	... log_method = "append"
2020-07-09 14:36:04	00000418	SSC: MultiMedia Timer correction is enabled.
2020-07-09 14:36:04	00000001	CPU: The ACE option is omitted; enable ACE as license default.
2020-07-09 14:36:04	00000133	Advanced CPU Emulation (ACE) enabled.
2020-07-09 14:36:04	0000032C	"My_MV3K6" started.
2020-07-09 14:36:05	00000419	OPAO: Connected. Remote 127.0.0.1:56157.
2020-07-09 14:39:07	0000032D	"My_MV3K6" stop request received.
2020-07-09 14:39:07	0000041A	OPAO: Disconnected.
2020-07-09 14:39:07	0000024D	Licensing component received stop request.
2020-07-09 14:39:07	0000032E	Stopped.
2020-07-09 14:39:07	0000024A	Logging stopped.

At the bottom of the window, there are several buttons: 'Create VM from Template', 'Add Existing VM', 'Update VM List', 'Start', 'Stop', 'Remove VM', 'Edit Configuration' (highlighted in yellow), 'Update Log', 'Display Log', and 'Show Console'.

The configuration file specifies all the settings of the CHARON VM. This will be discussed in the next chapter.

## CHARON Virtual Machine startup mode

By default, the CHARON VM startup mode is set to "Manual" meaning a user has to start the CHARON VM manually using the CHARON Virtual Machine Manager after a host reboot. If the CHARON VM has to start automatically, select the "Automatic" option.

⚠ Setting the mode to "Automatic" will only start the CHARON VM, it will not boot the operating system. This part will be discussed in the next chapter.


Select the CHARON VM startup behavior in the "Startup type" drop down list:

**Services are set to manual start by default. Automatic start at boot can be enabled after successfully testing the configuration. You can change this setting below.**

Startup type:

## CHARON version to be used for CHARON VM

The CHARON installation procedure allows several versions of CHARON to be installed on the host at the same time (V4.8 and later only). This option can be used for testing.

 If a patch has been applied, only the patched version is offered. If you need to rollback to the previous version, you must uninstall the patch according to the documentation provided with the patch.

Select the desired CHARON version from the "Executable" drop-down list and press the "Apply" button:

**By default, virtual machine is created with the latest CHARON executable image version installed on the system. To change the version of the executable image for the VM: choose the desired executable image from the list box below, press the <Apply> button, restart the VM.**

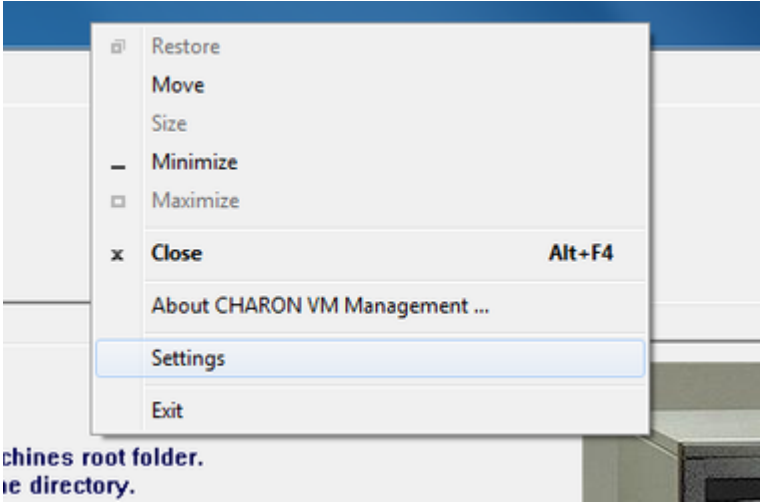
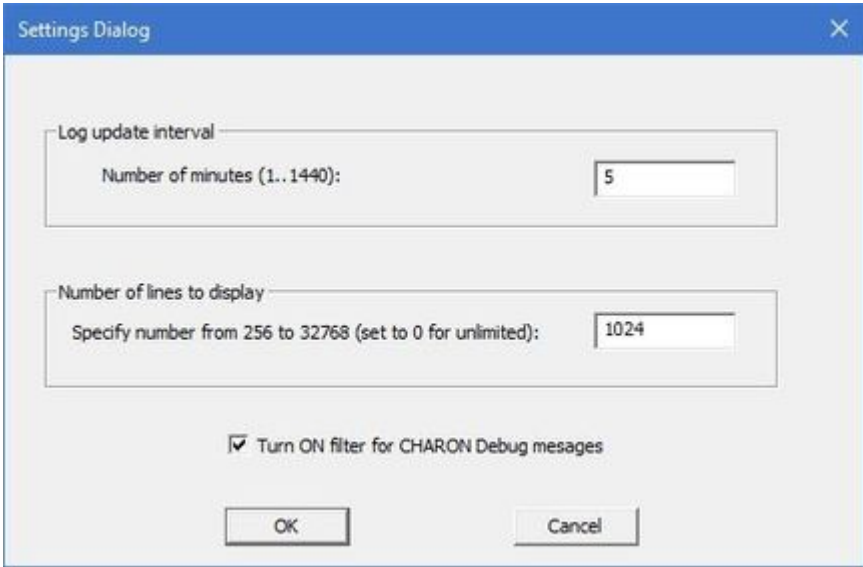
Executable:

CHARON-VAX, 4.11.20404

Apply

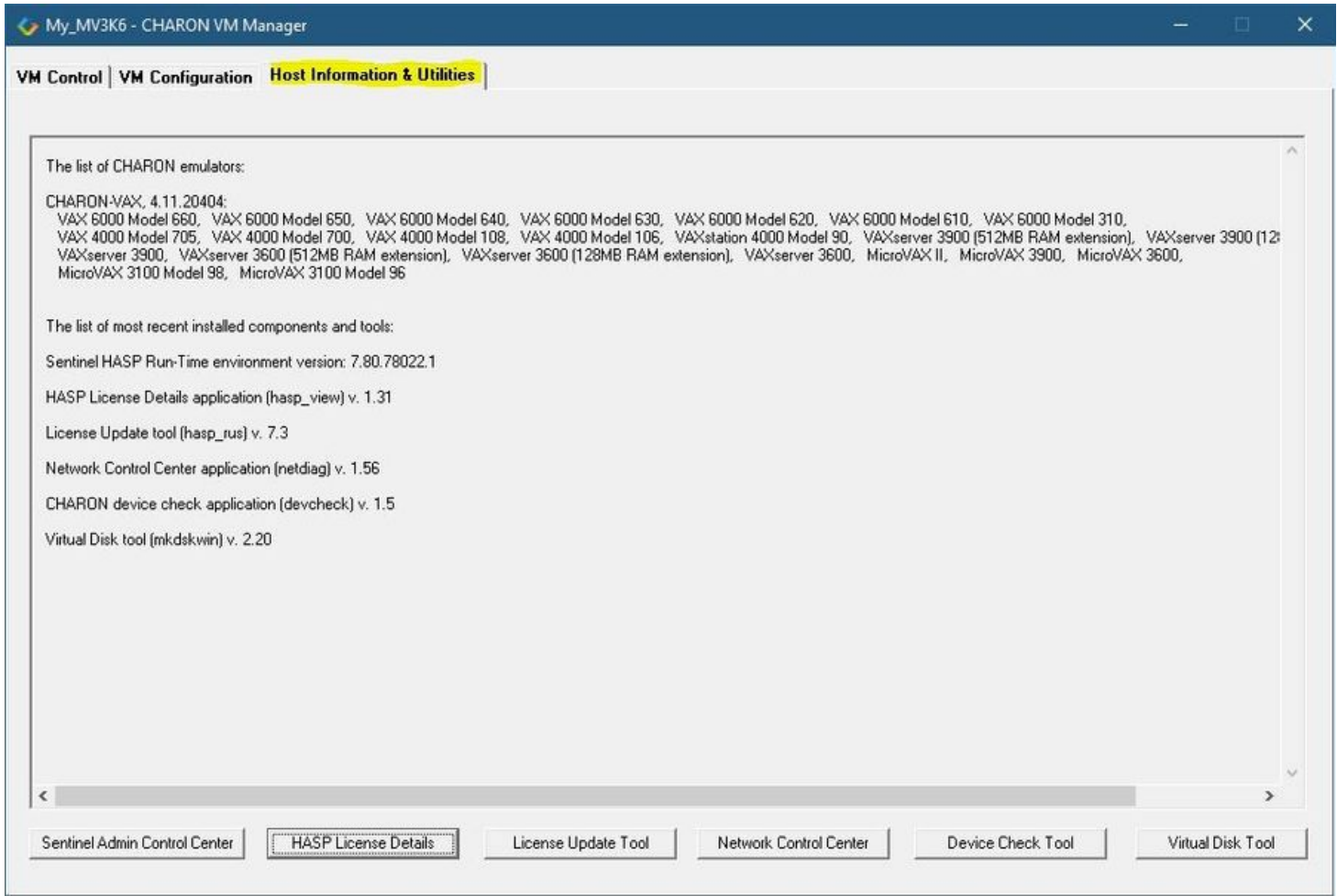
## Service functions

Additional functionalities are available and described below:

Function	Description
<b>Update Log</b>	<p>The CHARON Virtual Machine Manager updates the displayed log every 5 minutes and displays up to 1024 lines by default. It can be changed in the "Settings" dialog of the program system menu.</p> <p>Right click on top of the CHARON Virtual Machine Manager window and select "Settings":</p>  <p>In the appearing dialog you can change the interval:</p>  <p>If it is required to see most up-to-date version of the log, press the "Update Log" button.</p> <p><b>i</b> The "Turn ON filter for CHARON Debug messages" option is used only with specific executable files provided by Stromasys for debugging. It is of no use with standard releases.</p>
<b>Display Log</b>	<p>Available from the VM control tab, bottom right.</p> <p>Creates a separate window containing the selected CHARON VM log</p>
<b>Show Console</b>	<p>Available from the VM control tab, bottom right, and if the VM is running.</p> <p>Opens the selected CHARON VM console if it has been closed</p>

# Running CHARON utilities

Select the "Host Information & Utilities" tab to get quick access to the CHARON utilities:



These utilities are described below:

Button	Description
Sentinel Admin Control Center	Used to view CHARON licenses and manage them.
HASP License Details	Used to display the CHARON license content.
License Update Tool	Used to manage the CHARON licenses, collect the host system fingerprint.
Network Control Center	Used to configure the CHARON network.
Device check tool	Used to review the system resources that can be mapped to CHARON.
Virtual Disk Tool	GUI-based utility used to create custom or standard CHARON virtual disk containers.

# CHARON-VAX and CHARON-PDP for Windows configuration

## Table of Contents

- Creation of CHARON VM configuration
- VAX/PDP11 model specification
- Configuration name
- Log file parameters
  - Rotating log (default)
  - Single log
- TOY, ROM and EEPROM containers
- KW11 timer
- ACE mode
- Emulated memory (RAM) size
- Console
  - Mapping to system resources
  - Exit on F6 keypress
- Disk subsystem
  - MSCP disk controllers (RQDX3, KDB50, KDM70)
  - SCSI controller NCR53C94
- Tape subsystem
  - TQK50 controller
  - TUK50 controller
- Serial Lines
- Networking
- Auto boot
- Host load balance for SMP systems

## Creation of CHARON VM configuration

When a CHARON Virtual Machine (VM) is created from a template using the CHARON Virtual Machine Manager, it has to be updated to meet the desired configuration.

This configuration is represented as a text file ("configuration file") containing some specific keywords to define the main settings such as amount of memory, number of CPUs, peripheral devices as well as specifics of CHARON VM executions such as name of VM log file, number of host CPUs used for emulation, etc.

To change the configuration file, open the CHARON Virtual Machine Manager from its shortcut on Desktop / Start menu or from the tray menu item, select the target CHARON VM and press the "Edit Configuration" button:



My\_MV3K6 - CHARON VM Manager

VM Control | VM Configuration | Host Information & Utilities

CHARON Virtual Machine Status: **Stopped**

Date:	Message ID:	Message text:
2020-07-09 14:36:03	0000024D	The value "MICROVAX_3600" of the required parameter "P_HW" is not found in the list "PDP1193.PC"
2020-07-09 14:36:03	0000024D	Checking product section 3.
2020-07-09 14:36:04	0000024D	HASP clock: 09-Jul-2020 15:53:58.
2020-07-09 14:36:04	0000024D	Host clock: 09-Jul-2020 14:36:04.
2020-07-09 14:36:04	0000024D	License number: "003.msc.test.center.kirill".
2020-07-09 14:36:04	0000024D	Product License number: "p.vax".
2020-07-09 14:36:04	0000024D	CHARON product code: "CHVAX-4110xx-WI-LI".
2020-07-09 14:36:04	0000024D	Unlimited license.
2020-07-09 14:36:04	0000024D	Feature 1 check interval 1 hour(s).
2020-07-09 14:36:04	0000024D	Concurrency info:
2020-07-09 14:36:04	0000024D	There are 7 instances allowed.
2020-07-09 14:36:04	0000024D	STROMASYS SA, (C) 2009-2020
2020-07-09 14:36:04	00000408	CHARON-VAX (MicroVAX 3600), V 4.11 B 20404, May 6 2020 / 003.msc.test.center.kirill / 14227262:
2020-07-09 14:36:04	00000336	The end user of this software has agreed to STROMASYS' Terms and Conditions for Software License
2020-07-09 14:36:04	00000097	OS Environment: Microsoft Windows 10 Pro, Professional, 64-bit, Multiprocessor Free, Build = 18362 (1
2020-07-09 14:36:04	00000098	Host CPU: Intel(R) Xeon(R) CPU E 31275 @ 3.40GHz x8.
2020-07-09 14:36:04	00000099	Host Memory: 24320Mb
2020-07-09 14:36:04	0000041F	Configuration dump:
2020-07-09 14:36:04	0000041F	. session:
2020-07-09 14:36:04	0000041F	... configuration_name = "My_MV3K6"
2020-07-09 14:36:04	0000041F	... log_method = "append"
2020-07-09 14:36:04	00000418	SSC: MultiMedia Timer correction is enabled.
2020-07-09 14:36:04	00000001	CPU: The ACE option is omitted; enable ACE as license default.
2020-07-09 14:36:04	00000133	Advanced CPU Emulation (ACE) enabled.
2020-07-09 14:36:04	0000032C	"My_MV3K6" started.
2020-07-09 14:36:05	00000419	OPAD: Connected. Remote 127.0.0.1:56157.
2020-07-09 14:39:07	0000032D	"My_MV3K6" stop request received.
2020-07-09 14:39:07	0000041A	OPAD: Disconnected.
2020-07-09 14:39:07	0000024D	Licensing component received stop request.
2020-07-09 14:39:07	0000032E	Stopped.
2020-07-09 14:39:07	0000024A	Logging stopped.

Notepad will be used to edit the configuration file:

```

#
# Copyright (C) 1999-2018 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 VAXserver 3600.
#
#-----

set session hw_model = VAXserver_3600

#-----
#
# Choose a name for the instance, if needed, to differentiate it among other
# instances running on the same host.
#
#-----

include configuration_name.icfg

#-----
#
# Use the following commands to disable the rotating LOG files and enable
# a single LOG file. Select either append or overwrite (for each time the
# instance starts) and specify desired log path and file name.
#
#-----

set session log_method = append
#set session log_method = overwrite
#set session log = VAXserver_3600.log

#-----
#
# The following line tells the emulator where to preserve NVRAM content.
# The file maintains the current time of the emulated VAX (when it is not
# running) and other console parameters (such as default boot device).
#
#-----

```

## VAX/PDP11 model specification

The first configuration statement is the specification of the exact VAX/PDP11 hardware model to emulate, for example:

```
set session hw_model = MicroVAX_3600
```

You must leave this line untouched.

⚠ If you create the CHARON VM configuration file from scratch, it must be the very first uncommented line in the configuration file.

## Configuration name

The next configuration statement is the "Configuration name" option. If the virtual machine has been created using an existing template, the configuration name is defined in the `configuration_name.icfg` file otherwise it is defined directly in the configuration file using the "set session configuration\_name = <...>" syntax:

```
include configuration_name.icfg
```

### Notes:

- The configuration name is reported in the log file and is used to set the log file name for rotating log (see explanations [further](#)).
- Changing the configuration name in the "configuration\_name.icfg" file does not change the name of the virtual machine at Virtual Machine Manager level.
- The "configuration\_name.icfg" file can be found in the home directory of the VM. Select the target VM, open "VM Configuration" tab and press "Explore" button.

The configuration name can be any label that is meaningful.

### Example:

```
set session configuration_name = My_VAX
```

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

```
set session configuration_name = "My VAX"
```

## Log file parameters

The execution of a CHARON VM 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 problems with either the running CHARON VM or the emulated system configuration (such as the absence or malfunction of certain devices), the log file is the primary source to be analyzed for troubleshooting.

**i** If it becomes necessary to contact Stromasys for support, the configuration and log files will be requested to begin the problem resolution.

### CHARON-VAX log file example

```
20200510:133937:INFO :0:00000249:ethane.cxx(9010): Logging started.
20200510:133937:INFO :0:000003A5:ethane.cxx( 131): session: loading built-in configuration "MicroVAX_3600"...
20200510:133937:INFO :0:000003A6:ethane.cxx( 163): session: ... done loading built-in configuration
"MicroVAX_3600"
20200510:133937:INFO :0:000003A7:ethane.cxx( 321): session: loading service configuration "My_MV3600"...
20200510:133937:INFO :0:000003A8:ethane.cxx( 356): session: ... done loading service configuration
"My_MV3600"
20200510:133937:INFO :0:000003AA:ethane.cxx( 404): session: loading configuration file "C:\Program
Files\CHARON\Virtual Machines\My_MV3600\mv3k6.cfg"...
The previous message has been repeated 2 times.
20200510:133937:INFO :0:000003AB:ethane.cxx( 579): session: ... done loading configuration file
"configuration_name.icfg"
The previous message has been repeated 2 times.
20200510:133937:INFO :0:000003F2:sesmgr.cxx(1410): session: default log file size limit is 4194304 bytes
20200510:133937:INFO :0:0000032B:ethane.cxx(2694): Start request received.
20200510:133937:INFO :0:000003AC:ethane.cxx( 805): session: process affinity is 00000000000000FF, system
affinity is 00000000000000FF
20200510:133937:INFO :0:000003D1:ethane.cxx(1463): session: I/O domain affinity is 0000000000000003, CPU
domain affinity is 00000000000000FC
20200510:133937:INFO :0:0000024D:licenseman(1823): Checking the available license key "1422726238".
The previous message has been repeated 26 times.
20200510:133938:INFO :0:00000408:ethane.cxx(2867): CHARON-VAX (MicroVAX 3600), V 4.11 B 20404, May 10 2020 /
000.msc.test.center.nikolaev / 1918154109
20200510:133938:INFO :0:00000336:ethane.cxx(2908): 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
20200510:133938:INFO :0:00000097:ethane.cxx(2987): OS Environment: Microsoft Windows 10 Pro, 64-bit (Build
17134).
20200510:133938:INFO :0:00000098:ethane.cxx(2992): Host CPU: Intel(R) Xeon(R) CPU E31275 @ 3.40GHz x8.
20200510:133938:INFO :0:00000099:ethane.cxx(2997): Host Memory: 24320Mb
20200510:133938:INFO :0:0000041F:ethane.cxx(3223): Configuration dump::
. session:
. . configuration_name = "My_MV3600"
. . log_method = "append"
. . log_locale = "English"
. . log_show_messages = "all"
. . log_repeat_filter = "on"
. cdal:
. . mm_timer_correction_disable = "false"
. IPR:
. . mm_timer_correction_disable = "false"
. RAM:
. . size = "16"
. SSC:
. . clock_period = "20000"
. . mm_timer_correction_disable = "false"
. q_bus:
. . mm_timer_correction_disable = "false"
. . interface = "enabled"
. UART:
. . line = "OPA0"
. . rx_done_delay = "10000"
. . tx_done_delay = "50"
. . tx_buffer_interval = "2000"
. . baud_rate = "9600"
. . stop_len = "1"
. . char_len = "8"
. . parity = "none"
. . dtr = "off"
```

```

. . rts = "off"
. . force_character_mask = "255"
. BDR:
. . boot = "manual"
. . mode = "normal"
. OPA0:
. . trace = "disabled"
. . port = "10003"
. . application = "putty.exe -load OPA0"
20200510:133938:INFO :0:00000418:busemul.cx( 188): SSC: MultiMedia Timer correction is enabled.
20200510:133939:INFO :0:00000001: tpool.cxx(1374): CPU: The ACE option is omitted; enable ACE as license
default.
20200510:133939:INFO :0:00000133: tpool.cxx(1606): Advanced CPU Emulation (ACE) enabled.
20200510:133939:INFO :0:0000032C:ethane.cxx(2730): "My_MV3600" started.
20200510:133939:INFO :0:00000419:winsckio.c(2703): OPA0: Connected. Remote 127.0.0.1:61468.
20200510:134413:INFO :0:0000032D:ethane.cxx(2776): "MicroVAX 3600" stop request received.
20200510:134413:INFO :0:0000024D:licenseman(1823): Licensing component received stop request.
20200510:134413:INFO :0:0000032E:ethane.cxx(2794): Stopped.
20200510:134413:INFO :0:0000024A:ethane.cxx(9706): Logging stopped.

```

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

```

set session log_method = append
#set session log_method = overwrite
#set session log = "MicroVAX_3600.log"

```

## Rotating log (default)

By default CHARON utilizes a so-called "rotating log" mechanism. This means that a new default log file is created each time CHARON starts and can switch to another log file if the size of the log file exceeds 64Kb (this behavior can be changed with the "set session log\_file\_size" and "set session log\_rotation\_period" parameters; see more details in the "General Settings" chapter of this guide).

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


The names of the rotating log files are composed as follows:


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

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

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


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

 The "log" parameter, if specified, must correspond to an existing folder.

 If the path is not specified, the log file is created in the "Home directory" mentioned in the VM Configuration tab of the CHARON Virtual Machine Manager.

## Single log

Alternatively it is possible to use a single log file. Uncomment the "set session log" line and specify the desired 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 "Home directory" mentioned in the VM Configuration tab of the CHARON Virtual Machine Manager.

The log file can be extended specifying "log\_method = append" (\*recommended for reporting issues\*) or overwritten, specifying "log\_method = overwrite".

Below is a specification of a CHARON VM log file located in the "C:\My\_logs" directory which will be appended each time the CHARON VM starts:

```

set session log_method = append
set session log = "C:\My_logs\my_vax.log"

```

## TOY, ROM and EEPROM containers

The TOY, ROM and EEPROM containers must be configured. **Their presence depends on the VAX/PDP11 model.** Stromasys recommends enabling them. If a container file of the given name does not exist, starting the CHARON VM will create it.

TOY means "Time of Year". Its container records time, date and some console parameters while the CHARON VM is not running. It is highly recommended to define and activate this container:

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

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

```
set rom container = "vx4k106.rom"
```

EEPROM stores the NVRAM content. It is highly recommended to define and activate this container:

```
set eeprom container = "charon.rom"
```

## KW11 timer

PDP11/93 and PDP11/94 emulators require the KW11 system timer to be configured:

```
#set KW11 frequency = 50
#set KW11 frequency = 60
```

Uncomment the line corresponding to the frequency used in the RSX11/RT11 systems to be migrated if needed.

## ACE mode

The next setting indicates whether the CHARON VM will use the Advanced CPU Emulation (ACE) mode:

```
#set cpu ace_mode = false
#set cpu ace_mode = true
```

**i** Note that ACE significantly speeds up CHARON execution.

Uncomment the desired mode. ACE uses a significant amount of host memory for its normal operation. Check the host resources before enabling.

The "ace\_mode" mode is license dependent, the "Charon-xx/Plus" products have ACE mode enabled by default.

## Emulated memory (RAM) size

The next parameter defines the amount of host memory the CHARON VM reserves for the emulation:

```
#set ram size = 32
set ram size = 64
```

The amount of RAM is specified in MB. It cannot exceed or be lower than certain values specific for each VAX/PDP11 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
PDP1193	2	4	2	2
PDP1194	2	4	2	2
MicroVAX_II	1	16	16	1,8,16
MicroVAX_3600	16	64	16	16
MicroVAX_3900	16	64	16	16
VAXserver_3600	16	64	16	16
VAXserver_3900	16	64	16	16
VAXserver_3600_128	32	128	32	32
VAXserver_3900_128	32	128	32	32
MicroVAX_3100_Model_96	16	128	16	16
VAXstation_4000_Model_90	16	128	16	16
VAX_4000_Model_106	16	128	16	16
VAX_6000_Model_310	32	512	32	32
VAXserver_3600_512	32	512	32	32
VAXserver_3900_512	32	512	32	32
MicroVAX_3100_Model_98	16	512	16	16
VAX_4000_Model_108	16	512	16	16
VAX_4000_Model_700	64	512	64	64
VAX_4000_Model_705	64	512	64	64
VAX_6610	128	3584	128	128
VAX_6620	128	3584	128	128
VAX_6630	128	3584	128	128
VAX_6640	128	3584	128	128
VAX_6650	128	3584	128	128
VAX_6660	128	3584	128	128

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 VAX/PDP11 model.



# Console

## Mapping to system resources

The next step is the specification of the VAX/PDP11 console (OPA0) serial line:

```
#load physical_serial_line/chserial OPA0 line="\\.COM1"
#load virtual_serial_line/chserial OPA0 port=10003
load virtual_serial_line/chserial OPA0 port=10003 application="putty.exe -load OPA0"
```

The goal of this configuration step is to tell the CHARON VM 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>" physical serial line.
virtual_serial_line	<p>Mapping to an IP port of CHARON VM host. Using this mapping it is possible to connect to the CHARON VM console and disconnect from it at any time without usage of CHARON Virtual Machine Manager "Show Console" option. It is also possible to start up some chosen application to connect to the CHARON VM console IP port.</p> <p><b>Parameters:</b></p> <ul style="list-style-type: none"> <li>port: specifies an IP port on the host to which the CHARON VM console will be mapped</li> <li>application: a string specifying a name of an application to be started together with CHARON VM to provide emulation of VAX/PDP11 console (terminal emulator) It is assumed that this application has already been configured to use the IP port specified by the corresponding "port" parameter</li> </ul> <p><b>Example:</b></p> <pre>load virtual_serial_line/chserial OPA0 port=10003 application="putty.exe -load OPA0"</pre> <p>In the example above, the CHARON VM uses the IP port "10003" for the VAX/PDP11 console and starts the "Putty" terminal emulator (included in the CHARON installation kit) 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" feature from the CHARON Virtual Machine Manager.</p>

The default setting is:

```
load virtual_serial_line/chserial OPA0 port=10003 application="putty.exe -load OPA0"
```



Note that the VAX 4000 and MicroVAX 3100 models have a 4-line QUART adapter onboard, so their configuration for the console line looks a bit different:

```
#load physical_serial_line/chserial TTA0 line="\\.\COM1"
#load virtual_serial_line/chserial TTA0 port=10000
#load virtual_serial_line/chserial TTA0 port=10000 application="putty.exe -load TTA0"

#set quart line[0]=TTA0

#load physical_serial_line/chserial TTA1 line="\\.\COM2"
#load virtual_serial_line/chserial TTA1 port=10001
#load virtual_serial_line/chserial TTA1 port=10001 application="putty.exe -load TTA1"

#set quart line[1]=TTA1
...

#load physical_serial_line/chserial OPA0 line="\\.\COM4"
#load virtual_serial_line/chserial OPA0 port=10003
load virtual_serial_line/chserial OPA0 port=10003 application="putty.exe -load OPA0"

set quart line[3]=OPA0
```

When using the VAX 4000 and MicroVAX 3100 models, it is possible to configure up to 4 independent console lines: OPA0, TTA0, TTA1 and TTA2. The main one is OPA0.

Note that additional parameters for the CHARON VM serial lines configuration can be added. Follow [this link](#) for details.

## Exit on F6 keypress

The following directive configures the emulator to exit immediately when the F6 key sequence is sent to the emulator console.

```
set OPA0 stop_on = F6
```

This means that if the F6 key is pressed in the terminal emulator being used to connect to the console (PuTTY, for example), the emulator will exit immediately.

Stromasys does not recommend enabling this feature. It means that any person (or software) which connects to the console port on the emulator host can stop the emulator and thus "pull the virtual power cable" from the OpenVMS Tru64 UNIX guest. In particular, many organizations have deployed automated port scanners in their datacenters to discover open ports on the systems in the network. Many such port-scanning solutions, if they are able to connect to an open port, will next try to send data to the port. In many cases this data can be interpreted as the stop sequence and cause the emulator to exit.

If you must allow incoming connections to the console port, you should configure the Windows firewall with a "whitelist" of IP addresses from which such connections are allowed, and disallow all other addresses.

## Disk subsystem

The disk subsystem and the mapping to the system resources can be done using the samples given in the template configuration files.

CHARON supports MSCP, DSSI, CI and SCSI disk controllers.

The examples below are for MSCP and SCSI controllers only. DSSI controllers are discussed in details in the [following section](#), CI controllers - in [this section](#).

## MSCP disk controllers (RQDX3, KDB50, KDM70)

Below is a typical configuration sample for the MSCP disk controller RQDX3:

```
#load RQDX3 DUA

#set DUA container[0]="<file-name>.vdisk"
#set DUA container[1]="\\.\PhysicalDrive<N>"
#set DUA container[2]="\\.\PhysicalDrive(DevID = <XXX-XXXX-XXXX-XXXX-XXXX-XXXX-XXXX-XXXX>)"
#set DUA container[3]="\\.\PhysicalDrive(iScsiTarget = <iSCSI target>, LUN = <LUN number>)"
#set DUA container[4]="\\.\CdRom<N>"
#set DUA container[5]="<file-name>.iso"
#set DUA container[6]="\\.\A:"
#set DUA container[7]="\\.\H:"

#load RQDX3 DUB address=...
#load RQDX3 DUC address=...
```

The first line ("load RQDX3 DUA") loads disk controller RQDX3 with name DUA, followed by 6 lines showing different ways of mapping to the host resources:

### ■ Mapping to the file representing a physical disk of the VAX/PDP11 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"


#### Example:

```
set DUA container[0]="my_bootable_disk.vdisk"
```

**Important note:** using compressed folders to store virtual disks and tapes is not supported.

### ■ Mapping to physical disk.

#### ■ "\\.\PhysicalDrive<N>"


 Be careful not to destroy all the information from the disk dedicated to CHARON VM by mistake. These disks must not be formatted by the host OS.

#### Example:

```
set DUA container[1]="\\.\PhysicalDrive1"
```

### ■ Mapping to 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 VM 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 DUA container[2]="\\.\PhysicalDrive(DevID= 6008-05F3-0005-2950-BF8E-0B86-A0C7-0001)"
```

### ■ Mapping to 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 DUA container[3]="\\.\PhysicalDrive(iScsiTarget= ign.2008-04:iscsi.charon-target-test1, LUN= 1)"
```

### ■ Mapping to CD-ROM device.

#### ■ "\\.\CdRom<N>"

#### Example:

```
set DUA container[4]="\\.\CdRom0"
```

- **Mapping to an ISO file for reading distribution CD-ROM image.**

- "<file-name>.iso"

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

**Example:**

```
set DUA container[5]="vms_distributive.iso"
```

- **Mapping to Floppy drive.**

- "\\A:"

**Example:**

```
set DUA container[6]="\\.\A:"
```

- **Mapping to some other type of drive, for example magneto-optical drive.**

- "\\<N>:"

**Example:**

```
set DUA container[6]="\\.\H:"
```

The numbers in the square brackets represent the unit numbers associated with each container of the MSCP controller. For example, the 3rd "set DUA" line of the configuration sample above creates disk "DUA2". The maximum unit number allowed is 9999, significantly more than the original hardware provided.

It is possible to load several RQDX3 controllers DUB, DUC, etc. (see lines 6-7, above) by configuring specific addresses for them on the Qbus. Use the "CONFIGURE" utility available on the VAX console to determine the addresses. Please refer to specific HP documentation for further information.

Please also refer to the HP documentation for information on placement of additional KDM70 controllers on an XMI bus (VAX 6000 models) and additional KDB50 controllers on a BI bus (VAX 6310).

Note that the KDM70 controller is capable of mapping to files representing tapes (tape images) and physical tape devices:

```
set PUA container[600] = "<file-name>.vtape"
set PUA container[601] = "\\.\Tape0"
```

Follow [this link](#) for details of (T)MSCP controllers configuration.

## SCSI controller NCR53C94

The VAX 4000 and MicroVAX 3100 models have an NCR53C94 SCSI controller on board for the support of different types of SCSI devices including disks and tapes. Optionally a second controller can be added.

Below is a typical configuration template for a preloaded "PKA" NCR53C94 SCSI controller:

```
# Mapping to disk image
load virtual_scsi_disk pka_0 scsi_bus=pka scsi_id=0
set pka_0 container="<file-name>.vdisk"

# Mapping to physical disk
load virtual_scsi_disk pka_1 scsi_bus=pka scsi_id=1
set pka_1 container[0]="\\.\PhysicalDrive<N>"
set pka_1 container[1]="\\.\PhysicalDrive(DevID =<XXX-XXXX-XXXX-XXXX-XXXX-XXXX-XXXX-XXXX>)"

# Mapping to host CD-ROM or DVD-ROM
load virtual_scsi_cdrom pka_3 scsi_bus=pka scsi_id=3
set pka_3 container="\\.\CdRom<N>"

# Mapping to *.ISO image
load virtual_scsi_cdrom pka_4 scsi_bus=pka scsi_id=4
set pka_4 container="<file-name>.iso"

# Mapping to tape drive recognized by Windows
load physical_scsi_device pka_5 scsi_bus=pka scsi_id=5
set pka_5 container="\\.\Tape<N>"

# Mapping to tape image
load virtual_scsi_tape pka_6 scsi_bus=pka scsi_id=6
set pka_6 container="<file-name>.vtape"

# Include this line to get access to "PKB" adapter
include kzdda.cfg

# Mapping to host floppy drive
load floppy_scsi_device pkb_6 scsi_by=pkb scsi_id=6
set pkb_6 container="\\.\A:"
```

Note that NCR53C94 SCSI controller mapping to system resources is done via specific auxiliary objects:

Mapping Object	Description
virtual_scsi_disk	<p>Mapping to a file representing VAX/PDP11 disk (disk image) on the host physical disk:</p> <ul style="list-style-type: none"> <li>• <b>"&lt;file-name&gt;.vdisk"</b> These files can be created from scratch with "MkDisk" utility. Data and OS disk backups are transferred from the original system via tapes or network and restored into these container files. Mapping may also include the full path (recommended), for example: "C:\My disks\my_boot_disk.vdisk"</li> </ul> <p><b>Important note:</b> using compressed folders to store virtual disks and tapes is not supported</p> <ul style="list-style-type: none"> <li>• <b>"\\.\PhysicalDrive&lt;N&gt;"</b> - mapping to physical disk. Be careful not to destroy all the information from the disk dedicated to CHARON VM by mistake! These disks must not be formatted by the host OS.</li> <li>• <b>"\\.\PhysicalDrive(DevID =XXXX-XXXX-XXXX-XXXX-XXXX-XXXX-XXXX-XXXX)"</b> - mapping to physical disk by its WWID. Be careful not to destroy all the information from the disk dedicated to CHARON VM 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).</li> <li>• <b>"\\.\PhysicalDrive(iScsiTarget = &lt;iSCSI target&gt;, LUN = &lt;LUN number&gt;)"</b> - mapping to iSCSI disks. iScsiTarget addresses the disk by its iSCSI target name. LUN specifies LUN on connected iSCSI disk.</li> </ul>
physical_scsi_device	<p>Mapping to a host SCSI device:</p> <ul style="list-style-type: none"> <li>• <b>"\\.\Tape&lt;N&gt;"</b> - mapping to a SCSI tape drive known to Windows</li> </ul>
virtual_scsi_cdrom	<p>Mapping to a host CD-ROM (not only SCSI) or to ISO image:</p> <ul style="list-style-type: none"> <li>• <b>"\\.\CdRom&lt;N&gt;"</b> - name of host CD-ROM drive</li> <li>• <b>"&lt;file-name&gt;.iso"</b> - name of ISO image. It may contain the full path, for example: "C:\My disks\vms_distributive.iso"</li> </ul>
virtual_scsi_tape	<p>Mapping to a file representing tape (tape image). It may contain a path, for example: "C:\My tapes\backup.vtape"</p>
floppy_scsi_device	<p>Mapping to a host floppy drive:</p> <ul style="list-style-type: none"> <li>• <b>"\\.\A:"</b> or <b>"\\.\B:"</b> - name of host CD-ROM drive. This parameter can be omitted.</li> </ul>

Let's look at the syntax of the mapping objects. All of them have several important parameters:

Mapping objects parameters	Description
scsi_bus	The name of the NCR53C94 SCSI controller. A typical value for the first and only preloaded NCR53C94 SCSI controller is "PKA"
scsi_id	SCSI address of this particular mapped device. Note that the preloaded NCR53C94 SCSI controller claims address "7"; addresses 0-6 are vacant and usable. The resulting names of virtual SCSI devices as they are seen in VAX/VMS are made up of the VMS name of the SCSI controller and the device address. For PKA, the device names in VMS will be DKA0, DKA1 etc
container	A keyword for specification of which host device is mapped to a particular virtual SCSI device. It can be disk or tape image, physical disk etc

It is possible to configure another NCR53C94 SCSI controller "PKB" by uncommenting the "include kzdda.cfg" line:

```
#include kzdda.cfg

#load virtual_scsi_disk pkb_0 scsi_bus=pkb scsi_id=0
#set pkb_0 container="<file-name>.vdisk"
...
```

In the example above the "pkb\_0" virtual SCSI device uses "PKB" controller by specifying a parameter "scsi\_bus=pkb"

**Note that versions of VAX/VMS older than 5.5-2H4 do not support the optional SCSI controller and might fail to boot if it is loaded.**

Follow [this link](#) for details of NCR53C94 SCSI controller controllers configuration.

## Tape subsystem

Some MSCP and SCSI controllers support tape devices. CHARON also emulates specific MSCP tape devices such as TQK50 and TUK50.

Follow [this link](#) for more details of (T)MSCP controllers configuration.

**Important note:** using compressed folders to store virtual disks and tapes is not supported

## TQK50 controller

Example statements to configure a TQK50 controller are shown below:

```
#load TQK50 MUA
#set MUA container[0]="<file-name>.vtape"
#set MUA container[1]="\\.\TapeN"
#load TQK50 MUB address=...
#load TQK50 MUC address=...
```

The first line ("load TQK50 MUA") loads a tape controller TQK50 and names it MUA. The following 2 lines demonstrate different ways of mapping to host resources:

Type of mapping	Description
"<file-name>.vtape"	Mapping to the file representing the tape (tape image). These files are created automatically. Mapping may also include a full path (recommended), for example: "C:\My tapes\backup.vtape"
"\\.\TapeN"	Mapping to a host tape device.

Numbers in the square brackets represent unit numbers associated with each container of the TQK50 controller. For example, the 2nd "set MUA" line of the configuration sample above creates the "MUA1" tape drive. The maximum unit number allowed is 9999, significantly more than the original hardware provided

It is possible to load several TQK50 controllers (see the lines 4-5) by configuring specific addresses for them on the Qbus. Use the "CONFIGURE" utility available on the VAX console to determine the addresses. Please refer to specific HP documentation for further information.

## TUK50 controller

TUK50 is a UNIBUS controller used by the VAX 6310:

```
load DWBUA UBA vax_bi_node_id = 14
load TUK50 MUA
#set MUA container[0] = "<file-name>.vtape"
#set MUA container[0] = "\\.\TapeN"
```

The first line loads a UNIBUS BI adapter "DWBUA". Configure then the "TUK50" tape controller the same way as the TQK50.

## Serial Lines

CHARON supports the following serial lines controllers: CXA16, CXB16, CXY08, DHQ11, DHV11, DZV11, DZQ11, DL11, DLV11, DZ11, DHW42-AA, DHW42-BA and DHW42-CA.

All of them are configured according to the following template:


```
#load DHV11/DHV11 TXA
load DHQ11/DHV11 TXA
#load CXY08/DHV11 TXA
#load CXA16/DHV11 TXA
#load CXB16/DHV11 TXA

#load physical_serial_line/chserial TXA0 line="\\.\COMn"
#load virtual_serial_line/chserial TXA0 port=10010
#set TXA line[0]=TXA0

#load physical_serial_line/chserial TXA1 line="\\.\COMn"
#load virtual_serial_line/chserial TXA1 port=10011
#set TXA line[1]=TXA1
...
#load DHV11/DHV11 TXB address=...
#load DHQ11/DHV11 TXB address=...
#load CXY08/DHV11 TXB address=...
#load CXA16/DHV11 TXB address=...
#load CXB16/DHV11 TXB address=...
```

The first 5 lines above are example given for loading serial line controllers of different types. The name of the controller (in this example) will be "TXA".

Once the controller is loaded it can be mapped to system resources (lines 6-11). The following options are available:

Option	Description
physical_serial_line	Mapping to host "\\.\COM<N>" serial line
virtual_serial_line	<p>Mapping to an IP port. This mapping makes it possible to connect to and disconnect from the CHARON VM serial line at any time. It is also possible to start up some chosen application to connect to the CHARON serial line IP port.</p> <p><b>Parameters:</b></p> <ul style="list-style-type: none"> <li>■ <code>port</code>: specifies an IP port on the host to which the CHARON VM serial line will be mapped</li> <li>■ <code>application</code>: a string specifying a name of an application to be started together with CHARON to provide emulation of VAX/PDP11 serial line (terminal emulator). It is assumed that this application has already been configured to use the IP port specified by the corresponding "port" parameter</li> </ul> <p><b>Example:</b></p> <pre>load virtual_serial_line/chserial TXA0 port=10011 application="putty.exe -load TXA0"</pre> <p>In the example above, the CHARON VM uses the IP port "10011" for the VAX/PDP11 serial line and starts the "Putty" terminal emulator (included the CHARON installation kit) with a "TXA0" saved session directing it to connect to the IP port "10011".</p> <p> Note that the "Putty" saved session used in this example is not installed during the CHARON installation procedure. It has to be created manually to connect to the chosen port ("10011" in this example). You can use the OPA0 console configuration specifics as an example to create your own "Putty" configurations. It is also possible to use other terminal emulators. Consult with their User's Guides to learn how to specify loading specific configuration on their startup.</p>

Look at the line "`set TXA line[0]=TXA0`" in the example. This one and the following lines of similar syntax map the loaded virtual controller ("TXA") to instances of the host serial lines ("TXA<N>").

The number of possible serial lines for each controller depends on its type and corresponds to the HP specification for a given controller.

It is possible to load several CXA16, CXB16, CXY08, DHQ11, DHV11, DZV11, DZQ11, DL11, DLV11 and DZ11 controllers (see the lines 12-16) by configuring specific addresses for them on the Qbus. Use the "CONFIGURE" utility available on the VAX/PDP11 console to determine the addresses. Please refer to specific HP documentation for further information.

The VAX 4000 and MicroVAX3100 models support DHW42-AA, DHW42-BA and DHW42-CA serial lines adapters:

```
#load DHW42AA/DHV11 TXA
#load DHW42BA/DHV11 TXA
#load DHW42CA/DHV11 TXA

#load physical_serial_line/chserial TXA0 line="\\.\COMn"
#load virtual_serial_line/chserial TXA0 port=10010
#set TXA line[0]=TXA0
```

Configuring these adapters is the same as above, except it is possible to load one and only one instance of DHW42-AA, DHW42-BA or DHW42-CA.

Note that additional parameters for CHARON VM serial lines configuration can be added. Follow [this link](#) for details.



# Networking

CHARON supports DEQNA, DESQA, DELQA, DEUNA, DELUA, DEMNA, DEBNI and PMADAA virtual network adapters.

All of them are configured in a similar way:

```
load DELQA/DEQNA XQA

load packet_port/chnetwrk XQA0 interface="connection:Charon"
set XQA interface=XQA0

#load DELQA XQB address=...
#load DELQA XQC address=...
```

In the example above the first line loads DELQA virtual adapter with a name "XQA"; the following 2 lines map it to host network interface having a name "Charon" ("connection" is a key word).

The mapping is performed in 2 steps:

1. A mapping object "packet\_port" with a name "XQA0" is loaded and connected to host interface having a name "Charon" so the CHARON VM will use this interface for its networking
2. The loaded DELQA virtual adapter "XQA" is connected to the "packet\_port" object "XQA0"

It is possible to load several DEQNA, DESQA, DELQA, DEUNA and DELUA controllers (see the lines 4-5) by configuring specific addresses for them on the Qbus. Use the "CONFIGURE" utility available on the VAX/PDP11 console to determine the addresses. Please refer to specific HP documentation for further information.

Some network adapters available in the CHARON VM are preloaded (for example, the SGEC controller for the MicroVAX 3100 with the predefined name "EZA"), so their configuration is even more simple:

```
load packet_port/chnetwrk EZA0 interface="connection:Charon"
set EZA interface=EZA0
```

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.

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

Follow [this link](#) for more details of CHARON VM network controllers configuration.

## Auto boot

CHARON can be configured to automatically boot an operating system at start up.

The MicroVAX 3100, VAX/PDP11 6310 and VAX/PDP11 4000 models boot automatically if the correct boot flags are set at the VAX/PDP11 console level:

```
>>>set halt reboot
```

Please check that the TOY, EEPROM and ROM containers (see above) are enabled so the updated parameters are saved upon reboot.

The ROM of certain VAX models (MicroVAX II, MicroVAX 3600, MicroVAX 3900, VAXserver 3600 and VAXserver 3900) does not allow the SRM console to accept the commands to enable auto booting. In this case, a specific parameter can be defined in the configuration file as a workaround:

```
set bdr boot=auto
```

The CHARON-VAX 6000 models have a similar configuration setting:

```
set xmi boot=auto
```

The CHARON-PDP models have the following setting:

```
set cpu_0 auto_boot = "DU0"
```

These configuration lines are specified in the template configuration files in commented out form. In case of absence please add the corresponding line according to the examples above.

## Host load balance for SMP systems

The VAX 6620 through VAX6660 models emulate 2-6 CPUs respectively. In this situation, loading of the host system can be tuned with the following configuration file settings:

Setting	Description
affinity	<p>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.</p> <p>By default the emulator instance allocates as many host CPUs as possible.</p> <p>The "affinity" parameter overrides the default and allows explicit specification of which host CPUs will be used by the instance. The affinity does not reserve the CPU for exclusive use.</p> <p><b>Example:</b></p> <pre>set session affinity = "0, 2, 4, 6"</pre>
n_of_io_cpus	<p>Reserves host CPUs (of those specified by "affinity" parameter, if any) for use by the emulator for I/O handling.</p> <p>By default the emulator instance reserves one third of available host CPUs for I/O processing (round down, at least one).</p> <p>The "n_of_io_cpus" parameter overrides the default by specifying the number of I/O host CPUs explicitly.</p> <p><b>Example:</b></p> <pre>set session n_of_io_cpus = 2</pre>

# Migration to CHARON-VAX and CHARON-PDP for Windows


## Table of Contents

- Introduction
- Creating CHARON Virtual Machine
- Collecting information about the source VAX system
- Creation of CHARON-VAX configuration file
- Making disk images
- Installation of VAX operating system
- Making remote backups
- Restore backups to CHARON-VAX disks
- Alternative ways of data transfer

## Introduction

This section describes how to migrate your VAX system to CHARON-VAX. We will use a sample MicroVAX 3600 system to demonstrate the migration procedure. The process is similar for all CHARON-VAX models.

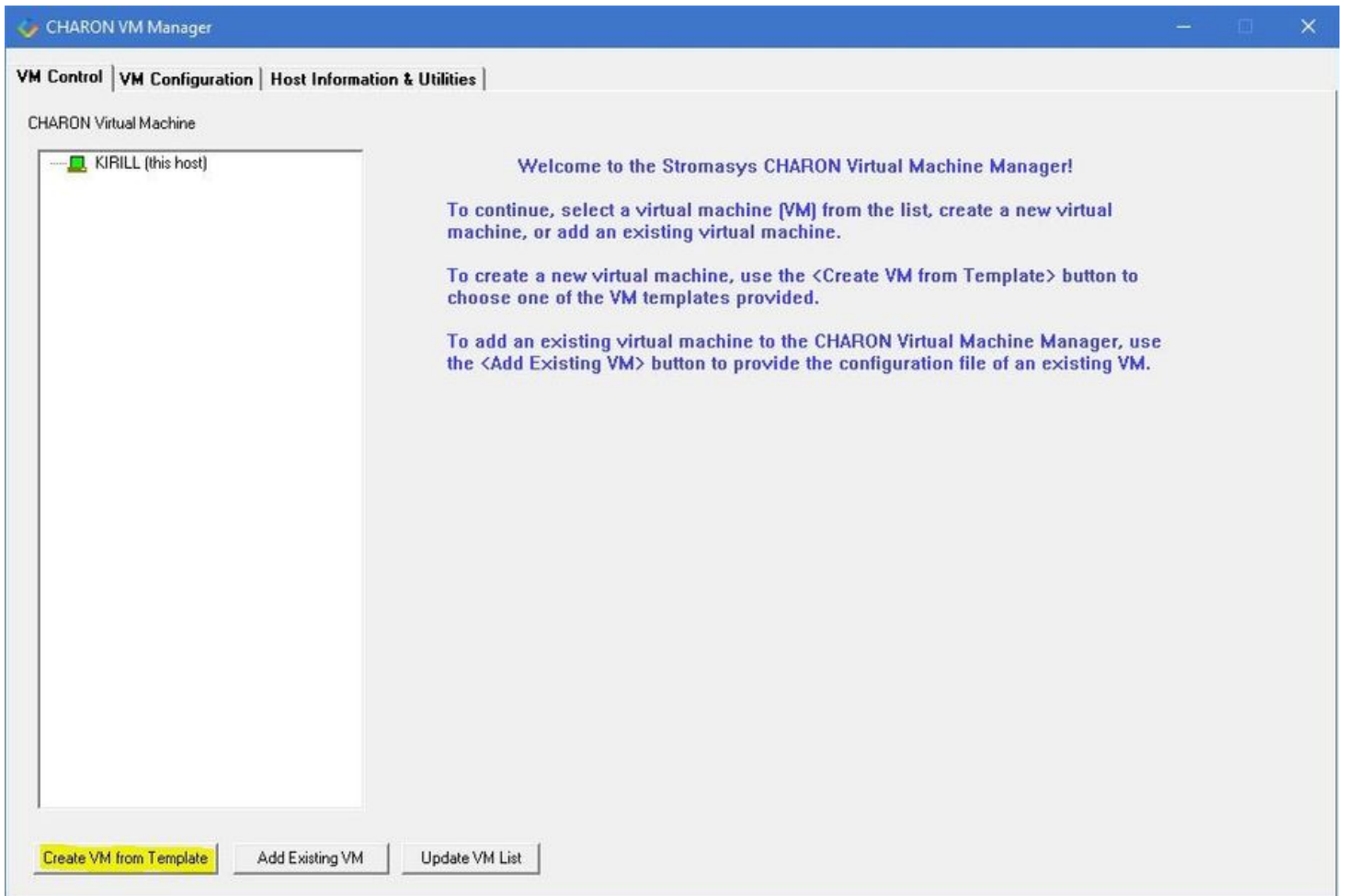
CHARON-PDP/11 migration is done in similar way except for the step of acquiring the QBUS addresses and vectors directly from console. Instead this information should be collected directly from the RSX11/RT11 operating system installed on the host to be migrated. Please refer to the RSX11/RT11 User's Guide for guidelines.

-  If CHARON-VAX based virtual system needs to be created from scratch, refer to [this Application Note](#) describing how to find proper Qbus addresses and Vectors for each virtual device.

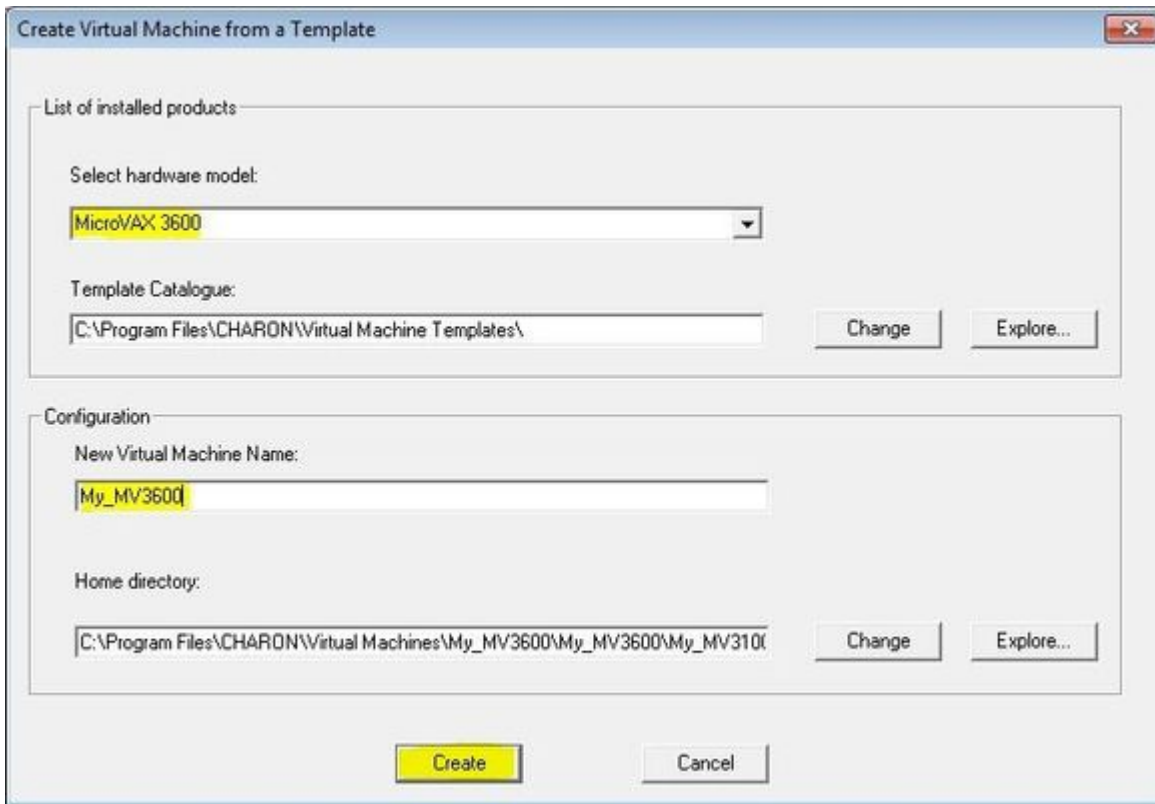
# Creating CHARON Virtual Machine

As a first step it is required to create a CHARON Virtual Machine (VM) using the CHARON Virtual Machine Manager in the following way:

Press the "Create VM from Template" button:



Specify the HP VAX model to be emulated and the name of the VM then press the "Create" button:



The VM will be created:



Select the VM from the left pane and press the "Edit Configuration" button:

The screenshot shows the CHARON VM Manager window for a VM named "My\_MV3K6". The status is "Stopped". The left pane shows a tree view with "KIRILL (this host)" and "My\_MV3K6". The main area displays a log of events. The bottom toolbar contains buttons for "Create VM from Template", "Add Existing VM", "Update VM List", "Start", "Stop", "Remove VM", "Edit Configuration", "Update Log", "Display Log", and "Show Console".

Date:	Message ID:	Message text:
2020-07-09 14:36:03	0000024D	The value "MICROVAX_3600" of the required parameter "P_HW" is not found in the list "PDP1193,PC
2020-07-09 14:36:03	0000024D	Checking product section 3.
2020-07-09 14:36:04	0000024D	HASP clock: 09-Jul-2020 15:53:58.
2020-07-09 14:36:04	0000024D	Host clock: 09-Jul-2020 14:36:04.
2020-07-09 14:36:04	0000024D	License number: "003.msc.test.center.kirill".
2020-07-09 14:36:04	0000024D	Product License number: "p.vax".
2020-07-09 14:36:04	0000024D	CHARON product code: "CHVAX-4110xx-WI-LI".
2020-07-09 14:36:04	0000024D	Unlimited license.
2020-07-09 14:36:04	0000024D	Feature 1 check interval 1 hour(s).
2020-07-09 14:36:04	0000024D	Concurrency info:
2020-07-09 14:36:04	0000024D	There are 7 instances allowed.
2020-07-09 14:36:04	0000024D	STROMASYS SA, (C) 2009-2020
2020-07-09 14:36:04	00000408	CHARON-VAX (MicroVAX 3600), V 4.11 B 20404, May 6 2020 / 003.msc.test.center.kirill / 14227262:
2020-07-09 14:36:04	00000336	The end user of this software has agreed to STROMASYS' Terms and Conditions for Software License
2020-07-09 14:36:04	00000097	OS Environment: Microsoft Windows 10 Pro, Professional, 64-bit, Multiprocessor Free, Build = 18362 (1
2020-07-09 14:36:04	00000098	Host CPU: Intel(R) Xeon(R) CPU E31275 @ 3.40GHz x8.
2020-07-09 14:36:04	00000099	Host Memory: 24320Mb
2020-07-09 14:36:04	0000041F	Configuration dump:
2020-07-09 14:36:04	0000041F	. session:
2020-07-09 14:36:04	0000041F	... configuration_name = "My_MV3K6"
2020-07-09 14:36:04	0000041F	... log_method = "append"
2020-07-09 14:36:04	00000418	SSC: MultiMedia Timer correction is enabled.
2020-07-09 14:36:04	00000001	CPU: The ACE option is omitted; enable ACE as license default.
2020-07-09 14:36:04	00000133	Advanced CPU Emulation (ACE) enabled.
2020-07-09 14:36:04	0000032C	"My_MV3K6" started.
2020-07-09 14:36:05	00000419	OPAD: Connected. Remote 127.0.0.1:56157.
2020-07-09 14:39:07	0000032D	"My_MV3K6" stop request received.
2020-07-09 14:39:07	0000041A	OPAD: Disconnected.
2020-07-09 14:39:07	0000024D	Licensing component received stop request.
2020-07-09 14:39:07	0000032E	Stopped.
2020-07-09 14:39:07	0000024A	Logging stopped.

The VM configuration file will be opened in Notepad. Now the file can be updated to reproduce the target HP VAX configuration.

## Collecting information about the source VAX system

The first step is to determine the exact configuration of your VAX hardware in order to create the CHARON VM configuration file.

Turn on your source VAX system. At the ">>>" prompt, issue the "show qbus" and "show device" commands:

```
>>>show qbus
Scan of Qbus I/O Space
-200000DC (760334) = FFFF (300) RQDX3/KDA50/RRD50/RQC25/KFQSA-DISK
-200000DE (760336) = 0B40
-20000124 (760444) = FFFF (304) TQK50/TQK70/TU81E/RV20/KFQSA-TAPE
-20000126 (760446) = 0BC0
-20000140 (760500) = 0080 (310) DHQ11/DHV11/CXA16/CXB16/CXY08
-20000142 (760502) = F081
-20000144 (760504) = DD18
-20000146 (760506) = 0140
-20000148 (760510) = 0000
-2000014A (760512) = 0000
-2000014C (760514) = 8000
-2000014E (760516) = 0000
-20000150 (760520) = 0080 (320) DHQ11/DHV11/CXA16/CXB16/CXY08
-20000152 (760522) = F081
-20000154 (760524) = DD18
-20000156 (760526) = 0140
-20000158 (760530) = 0000
-2000015A (760532) = 0000
-2000015C (760534) = 8000
-2000015E (760536) = 0000
-20001468 (772150) = FFFF (154) RQDX3/KDA50/RRD50/RQC25/KFQSA-DISK
-2000146A (772152) = 0B40
-20001920 (774440) = FF08 (120) DELQA/DEQNA/DESQA
-20001922 (774442) = FF00
-20001924 (774444) = FF01
-20001926 (774446) = FF02
-20001928 (774450) = FFD2
-2000192A (774452) = FF14
-2000192C (774454) = C000
-2000192E (774456) = 1030
-20001940 (774500) = FFFF (260) TQK50/TQK70/TU81E/RV20/KFQSA-TAPE
-20001942 (774502) = 0BC0
-20001F40 (777500) = 0020 (004) IPCR

Scan of Qbus Memory Space
>>>
```

```
>>>show device
UQSSP Disk Controller 0 (772150)
-DUA0 (RZ23)
-DUA1 (RZ24)

UQSSP Disk Controller 1 (760334)
-DUB2 (RZ25)
-DUB3 (RZ26)

UQSSP Tape Controller 0 (774500)
-MUA0 (TK50)

UQSSP Tape Controller 1 (760444)
-MUB3 (TK50)

Ethernet Adapter 0 (774440)
-XQA0 (08-00-01-02-D3-CC)
```



The source VAX configuration in this example is:

Controller	Address	Devices on controller
RQDX3	772150	-DUA0 (RZ23) -DUA1 (RZ24)
RQDX3	760334	-DUB2 (RZ25) -DUB3 (RZ26)
TQK50	774500	-MUA0 (TK50)
TQK50	760444	-MUB3 (TK50)
DHQ11	760520	
DHQ11	760500	
DESQLA	774440	-XQA0

To find out the exact types of controllers please refer to the source VAX system documentation.

## Creation of CHARON-VAX configuration file

Using the above information, the following parameters and values can be set in the configuration file:

```
...
#
# First RQDX3 controller on address 772150
#
load RQDX3/RQDX3 DUA address=017772150
set DUA container[0]="C:\My disks\rz23.vdisk"
set DUA container[1]="C:\My disks\rz24.vdisk"

#
# Second RQDX3 controller on address 760334
#
load RQDX3/RQDX3 DUB address=017760334
set DUB container[2]="C:\My disks\rz25.vdisk"
set DUB container[3]="C:\My disks\rz26.vdisk"

#
# First TQK50 controller on address 774500
#
load TQK50/TQK50 MUA address=017774500
set MUA container[0]="C:\My tapes\tape1.vtape"

#
# Second TQK50 controller on address 760444
#
load TQK50/TQK50 MUB address=017760444
set MUB container[3]="C:\My tapes\tape2.vtape"

#
# First DHQ11 controller on address 760500
#
load DHQ11/DHV11 TXA address=017760500
load virtual_serial_line/chserial TXA0 port=10010
set TXA line[0]=TXA0

#
# Second DHQ11 controller on address 760520
#
load DHQ11/DHV11 TXB address=017760520
load virtual_serial_line/chserial TXB0 port=10011
set TXB line[0]=TXB0

#
# DESQA controller on address 774440
#
load DESQA/DEQNA XQA address=017774440 interface=XQA0
load packet_port/chnetwrk XQA0 interface="connection:Charon"
...
```

Note the Qbus addresses specification:

- The number is prefixed with "0", meaning it is an octal value.
- The number of digits reflects the 22 bit Qbus architecture.

Additional DHQ11 serial lines can be mapped later. For now, only 2 lines are configured. They are mapped to IP ports 10010 and 10011.

The DESQA controller is mapped to a network interface having the name "Charon". This interface will be used for the CHARON VM (see the [Installation section](#) for more details) on this particular host.

## Making disk images

In our example, the mapping of the RQDX3 controller and the TQK50 tapes include physical devices and disk and tape images. The tape images have not to be manually created whereas the disk images have to be created as described below.

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

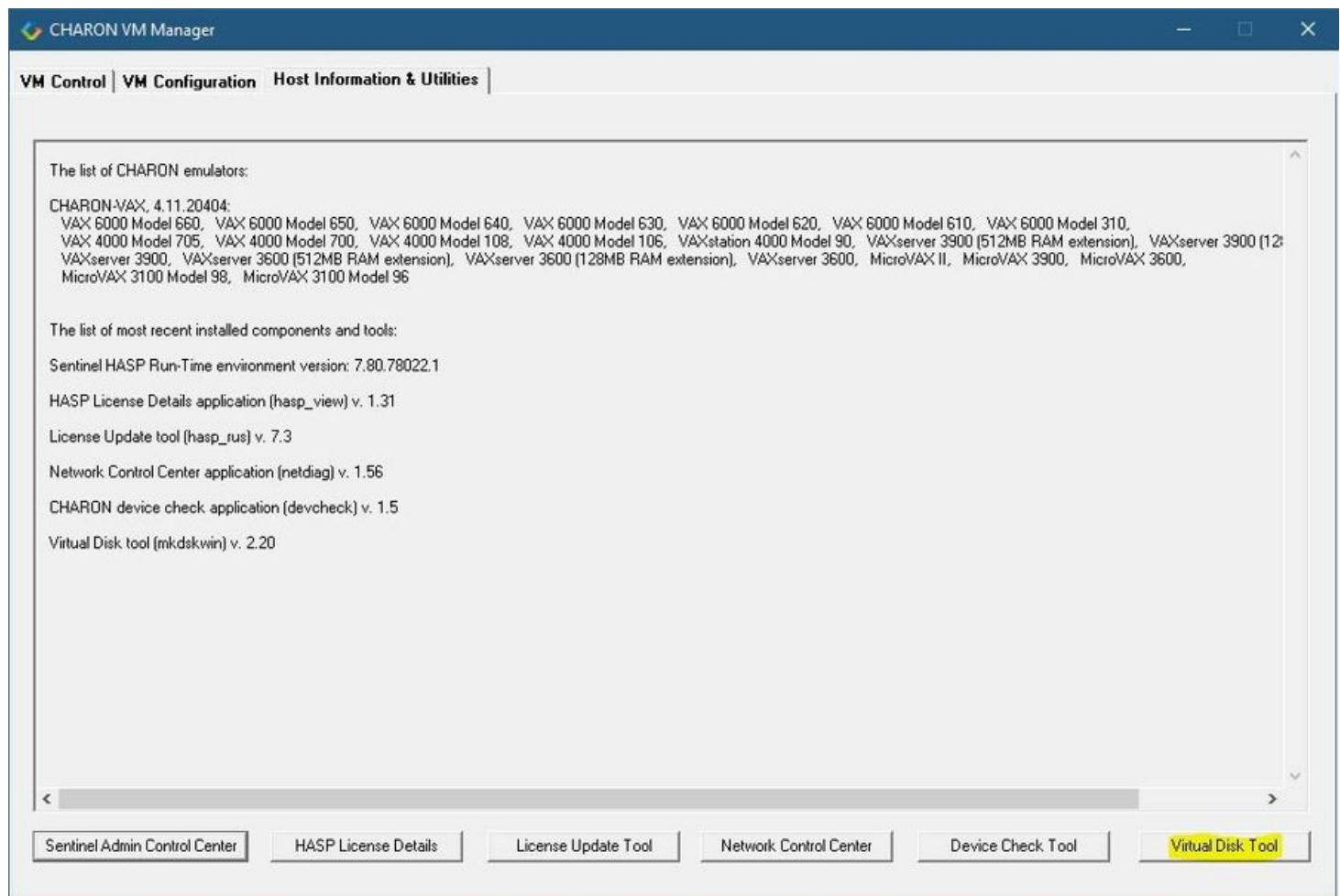
Create special directories for storing disk and tape images. The created directories are referenced in the sample configuration file above.

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

This operation can also be done using Windows Explorer.

**i** Alternatively it is possible to put the tape and disk images in the CHARON VM Home Directory.

Start the "MkDisk" utility by pressing the "Virtual Disk Tool" button in the "Host Information & Utilities" tab of the CHARON Virtual Machine Manager:



The following dialog will appear:

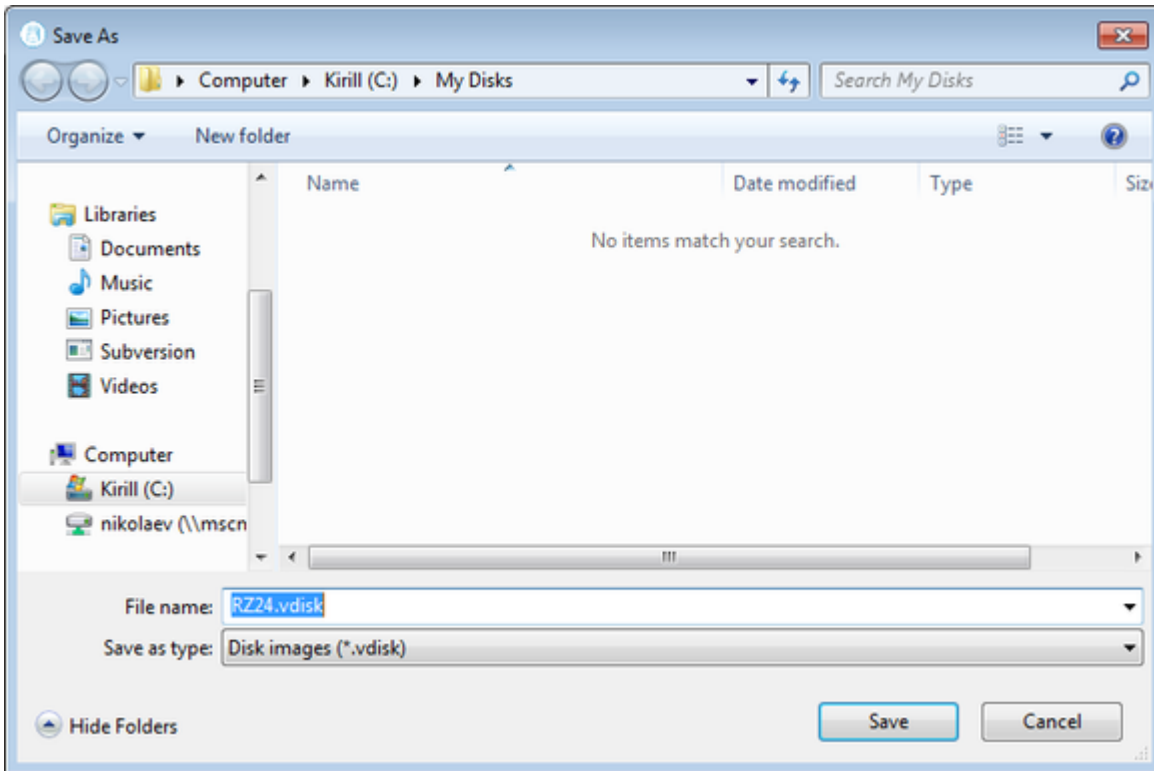


Select "VAX" in the "Select System" drop-down menu, "MSCP" in the "All Controllers" drop-down menu and "RZ24" (for example) in the "Select Disk Type" drop-down menu.

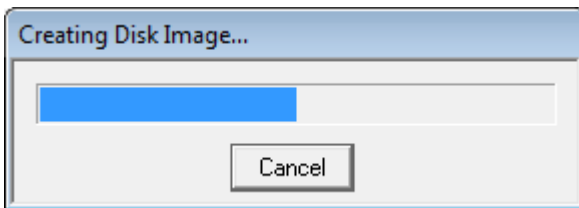
Press then the "Create Disk Image" button:



The utility will ask you to specify the output file name. Select the "C:\My disks" directory that has been created earlier, or create it directly in the dialog, and press the "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.

## Installation of VAX operating system

The next step is to transfer the data from the source VAX system to CHARON VM. The easiest way to do this is via backup over the network but for this operation a bootable network-enabled operating system on a CHARON-VAX disk image or physical disk is needed.

The example configures the CHARON-VAX MicroVAX 3600 system for installation of VAX/VMS from a distribution CD-ROM (usually it is "\\.\CdRom0" if the host has only one CD-ROM drive):

```
#
# First RQDX3 controller on address 772150 with addition of 3 units: a disk for VAX/VMS, storage disk and CD-
ROM drive
#
load RQDX3/RQDX3 DUA address=017772150
set DUA container[0]="C:\My disks\rz23.vdisk"
set DUA container[1]="C:\My disks\rz24.vdisk"
set DUA container[2]="C:\My disks\new_vms_system.vdisk"
set DUA container[3]="C:\My disks\backup_storage.vdisk"
set DUA container[4]="\\.\CdRom0"
```

**!** DUA3 will be the disk where all the source disks will be copied so its size needs to be large enough to store all the disk backup images.

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

Run the CHARON VM and boot from the CDROM named "dua4" ("migration.cfg" is the configuration file we use in this example):

```
...> mv3k6 migration.cfg
KA650-A V5.3, VMB 2.7
Performing normal system tests.
40..39..38..37..36..35..34..33..32..31..30..29..28..27..26..25..
24..23..22..21..20..19..18..17..16..15..14..13..12..11..10..09..
08..07..06..05..04..03..
Tests completed.
>>>boot dua4
```

Install VAX/VMS including DECnet on "dua2". The DECnet address must belong to the same area as the source VAX system.

Login to the newly installed VMS system and initialize the disk that will be used to store the backups. Let's assume its prompt is "newvms\$ "

```
newvms$ INIT DUA3: SCRATCH
newvms$ MOUNT/SYSTEM/NOASSIST DUA3: SCRATCH
```

## Making remote backups

Now we are ready to create disk backups of the source VAX system on the CHARON VM.

Boot the CHARON VM and make sure that the source VAX system is available via DECnet.

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

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

In this example, the CHARON VM system is node 1.400.

Issue the following commands from the source VAX. Let's assume its prompt is "source\$ ":

```
source$ BACKUP/IMAGE/IGNORE=INTERLOCK DUA0: 1.400"username password": :DUA3: [000000]DUA0.BCK/SAVE
source$ BACKUP/IMAGE/IGNORE=INTERLOCK DUA1: 1.400"username password": :DUA3: [000000]DUA1.BCK/SAVE
```

When the backup operation will be completed, the disk "DUA3" of the CHARON VM will contain 2 savesets: "DUA0.BCK" and "DUA1.BCK".

## Restore backups to CHARON-VAX disks



The savesets have now to be restored on their corresponding virtual disks. Login to the CHARON VM guest OS and issue this sequence of commands:

```
newvms$ MOUNT/FOREIGN DUA0:
newvms$ BACKUP/IMAGE DUA3:[000000]DUA0.BCK/SAVE DUA0:
newvms$ DISMOUNT DUA0:

newvms$ MOUNT/FOREIGN DUA1:
newvms$ BACKUP/IMAGE DUA3:[000000]DUA1.BCK/SAVE DUA1:
newvms$ DISMOUNT DUA1:
```

If you are going to have the CHARON VM and the original physical VAX on the network at the same time, you must change the network identity of one system, usually the CHARON VM.

The easiest way is to boot the CHARON VM on the restored system disk with the network disabled and to configure new addresses then enable the network and reboot.

 The NIC can be disabled with a `interface="(disabled)"` statement in the CHARON VM configuration file.

## Alternative ways of data transfer

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

- Connect a SCSI tape drive to the CHARON-VAX / CHARON-PDP host via a PCI card
  - Map the tape drive in the CHARON VM configuration file
    - a. Restore the source VAX system backups from tape to disk images via VMS/R SX11/RT11 running on the CHARON VM
    - b. Boot from standalone backups and restore the content to the CHARON VM virtual disks
  - Dump the source VAX/PDP11 system backups to tape images using the "mtd" utility and:
    - a. Boot from the freshly installed VMS/R SX11/RT11 system and restore the tape images to the CHARON VM virtual disks
    - b. Boot from the standalone backup and restore the content to the CHARON VM virtual disks
- Create a network cluster between the source VAX/PDP11 system and the CHARON VM (it is possible to use the source system as a boot server); then perform simple backups from one disk to another:

```
$ BACKUP/IMAGE/IGNORE=INTERLOCK REAL$DUA0: DUA0:
```

# CHARON-VAX for Windows DSSI cluster

## Table of Contents

- Introduction
- General description
- Configuration steps
- Example 1: Dual node DSSI cluster with 4 shared disks
- Example 2: Triple node DSSI cluster with multiple iSCSI disks

## Introduction

This section describes how to configure a DSSI cluster in CHARON-VAX for Windows.

## General description

The DSSI storage subsystem for the CHARON VAX 4000 106, 108, 700 and 705 models is based on the emulation of "SHAC" host adapters. Routing of SCS cluster information among the emulated "SHAC" host adapters of multiple nodes is done via separate TCP/IP links.

The DSSI storage subsystem is functionally emulated and operates at a much higher rate of throughput than the original hardware. Connections to the physical DSSI hardware is neither possible nor planned for future releases.

The current version of DSSI emulation for CHARON-VAX supports up to 3 VAX nodes in a virtual DSSI cluster and handles a maximum cluster size of 8 nodes. A single virtual DSSI network supports up to 256 storage elements.

For more details on a DSSI configuration follow [this link](#).

## Configuration steps

To create a CHARON-VAX DSSI cluster, the following elements must be configured:

1. "SHAC" host adapter
2. "HSD50" storage controller

DSSI hardware topology is emulated by establishing TCP/IP channels between the emulated SHAC host adapters of each CHARON Virtual Machine (VM). The emulated HSD50 storage controllers are then connected to every SHAC host adapter in the virtual DSSI network.

Cluster operation requires (virtual) disks that are simultaneously accessible by all CHARON VMs involved. This can be implemented for instance by using a properly configured iSCSI initiator / target structure or a fiber channel storage back-end. Disks on a multiport SCSI switch are not acceptable, as a SCSI switch does not provide true simultaneous access to multiple nodes.

Steps to configure DSSI cluster:

1. Set unique ID for SHAC controller of each VM

<b>Configuration</b>	<p>IDs of SHAC is set using console (and stored in ROM file, so make sure the ROM file is specified in configuration file) in the following way:</p> <pre>&gt;&gt;&gt;sho dssi_id DSSI_ID Bus 0/A = 1 DSSI_ID Bus 1/B = 1  &gt;&gt;&gt;set dssi_id 0 2 &gt;&gt;&gt;set dssi_id 1 2</pre>
<b>Description</b>	<p><b>⚠</b> Note that value of "dssi_id" must be unique for each SHAC instances and HSD50 controllers must not use them; for example if 0 and 1 is used as SHAC ID of different VMs the HSD50 controller instance cannot use ID 0 and 1.</p>



2. Configure preloaded SHAC adapters PAA

<b>Configuration</b>	<p>set PAA port[&lt;ID of the node to connect to&gt;]=&lt;port to receive connection from the node specified by ID&gt; host[&lt;ID of the node to connect to&gt;]="&lt;connected node host&gt;:&lt;connected node port&gt;"</p> <p><b>Example:</b></p> <pre>set PAA port[2]=11012 host[2]="pollux:11021"</pre> <p>In this example a VAX node SHAC adapter connects to a host "pollux" (having ID=2) to its port 11021 and expects connection from it on the port 11012.</p>
<b>Description</b>	<p>The second step is to interconnect SHAC adapters of different VMs.</p> <p>Use ID of the nodes as a reference for interconnections as it is shown in the example at left.</p> <p>Detailed meaning of the parameters is explained in "Example 1" section below</p>

3. Load HSD50 adapter


<b>Configuration</b>	<p>load HSD50 &lt;instance name&gt; dssi_host=&lt;SHAC instance&gt; dssi_node_id=&lt;HSD50 node ID&gt;</p> <p><b>Example:</b></p> <pre>load HSD50 DISKS dssi_host=PAA dssi_node_id=3</pre> <p>In this example HSD50 controller instance "DISKS" is loaded, connected to "PAA" SHAC controller and assigned with ID 3.</p>
<b>Description</b>	<p>Load HSD50 instance, assign ID for it and connect it to "PAA" SHAC adapter.</p> <p>For each member of cluster an identical definition of HSD50 controllers must be loaded. It is convenient to introduce a specific include configuration file containing the same configuration commands as it is shown in "Example 2" section below.</p> <p><b>⚠</b> Note that &lt;HSD50 node ID&gt; parameter must be unique for each SHAC and HSD50 instances; for example if 0 and 1 are used as IDs for nodes (see the Step 1) the HSD50 instance cannot use 0 or 1, but it can be set to 3, for example.</p>

## 4. Set SCS system ID and allocation class

<b>Configuration</b>	<pre>set &lt;HSD50 instance name&gt; scs_system_id=&lt;SCS ID&gt; mscp_allocation_class=&lt;allocation class&gt;</pre> <p><b>Example:</b></p> <pre>set DISKS scs_system_id=3238746238 mscp_allocation_class=1</pre> <p>In this example HSD50 instance "DISKS" (configured at the previous step) is assigned with SCS ID "3238746238" and MSCP allocation class 1.</p>
<b>Description</b>	<p>Set SCS system ID and the MSCP allocation class.</p> <p> These setting must be the same for all HSD50 controllers loaded!</p> <p> Note that the MSCP allocation class must be different from the allocation class of the VAX node (excluding some special cases when they can be the same - please refer to OpenVMS Guides on Clusters configuration for more details).</p> <p>For nodes it is set using MODPARAMS.DAT and AUTOGEN</p> <p>Refer to OpenVMS User's Guides for details</p>

## 5. Configure mapping to the system resources

<b>Configuration</b>	<pre>set &lt;HSD50 instance name&gt; container[&lt;unit number&gt;] = &lt;mapping&gt;</pre> <p><b>Example:</b></p> <pre>set DISKS container[0]="\\DiskServer\Share\dua0-rz24-vms-v6.2.vdisk"</pre> <p>In this example the Unit 0 of the HSD50 controller instance "DISKS" is mapped to some disk image, located in some shared directory on disk server.</p>
<b>Description</b>	<p>The final step is specifying HSD50 mapping to the system resources.</p> <p>Since the disks should be distributed between different VMs they should be located as disk images on some shared space or - alternatively - CHARON host should use such mechanism as iSCSI to address the common disks on some disk storage (they appear locally as "\\.\PhysicalDrive&lt;N&gt;").</p> <p>See the sections below for different examples of the mapping.</p>

 It is advisable to start any field test with implementing the cluster examples provided below

## Example 1: Dual node DSSI cluster with 4 shared disks

To setup two emulated VAX 4000 Model 108 VMs, we need two host machines, preferably running the same version of Windows.

Assume that these host systems have network host names *CASTOR* and *POLLUX* in the host TCP/IP network.

The following are CHARON VM configuration files for the emulated VAX 4000 Model 108 nodes running on *CASTOR* and *POLLUX*:

```
CASTOR node
...
set PAA port[2]=11012 host[2]="pollux:11021"
load HSD50 DISKS dssi_host=PAA dssi_node_id=3
set DISKS scs_system_id=3238746238 mscp_allocation_class=1
set DISKS container[0]="\\DiskServer\Share\dua0-rz24-vms-v6.2.vdisk"
set DISKS container[1]="\\DiskServer\Share\dua1-rz24-vms-v6.2.vdisk"
set DISKS container[2]="\\DiskServer\Share\dua2-rz24-vms-v6.2.vdisk"
set DISKS container[3]="\\DiskServer\Share\dua3-rz24-vms-v6.2.vdisk"
...
```

```
POLLUX node
...
set PAA port[1]=11021 host[1]="castor:11012"
load HSD50 DISKS dssi_host=PAA dssi_node_id=3
set DISKS scs_system_id=3238746238 mscp_allocation_class=1
set DISKS container[0]="\\DiskServer\Share\dua0-rz24-vms-v6.2.vdisk"
set DISKS container[1]="\\DiskServer\Share\dua1-rz24-vms-v6.2.vdisk"
set DISKS container[2]="\\DiskServer\Share\dua2-rz24-vms-v6.2.vdisk"
set DISKS container[3]="\\DiskServer\Share\dua3-rz24-vms-v6.2.vdisk"
...
```



Let's review both configurations step-by-step.


1. The first line of both configuration files establishes parameters for the preloaded "PAA" SHAC host adapter. Only 2 parameters of SHAC are important for us in this situation:

Parameter	Description
port	An integer value that specifies the TCP/IP port number on which an emulated SHAC host adapter listens for connections from another emulated SHAC host adapter.  Possible port values range from 1024 through 32767.
host	A string value that specifies the TCP/IP host name (and optional TCP/IP port number) to connect to another emulated SHAC host adapter.  The syntax for the string is "host-name[:port-no]" with possible values for "port-no" in the range from 1024 through 32767.

Thus, *CASTOR* connects to *POLLUX*'s port 11021 and listens for *POLLUX*'s connection on port 11012, *POLLUX* connects to *CASTOR*'s port 11012 and listens for *CASTOR*'s connection on port 11021

2. The second and third lines of both configuration files are for loading "DISKS", HSD50 storage controllers and its parametrization:

Parameter	Description
dssi_host	A string value that specifies the instance name of the emulated SHAC host adapter serving the virtual DSSI network. If this value is not set, CHARON VM tries to locate the host adapter automatically. This automatic lookup works only if the CHARON VM configuration has exactly one instance of the emulated SHAC host adapter.
dssi_node_id	An integer value that specifies the address of an emulated HSD50 storage controller on a virtual DSSI network. Possible values range from 0 through 7 (initially set to 0).  Note that this parameter must be unique for each SHAC and HSD50 instance; for example if 0 and 1 are used as the IDs for different ports of "PAA" the HSD50 instance cannot use 0 or 1. But it can use 3 for example.
scs_system_id	A string value that specifies the <i>SCSNODENAME</i> of the emulated HSD50 storage controller. The string can consist of up to 10 characters. Possible characters are uppercase letters: A through Z and integers.
mscp_allocation_class	An integer value that specifies the <i>ALLOCLASS</i> of an emulated HSD50 storage controller. Possible values are from 0 through 255 (initially set to 0).  Note that the MSCP allocation class must be the same as the allocation class of the VAX node. For nodes it is set using MODPARAMS.DAT and AUTOGEN Refer to OpenVMS User's Guides for details

 In both configuration files, the names of the emulated HSD50 storage controller "DISKS" must be identical. Not following this rule can cause data corruption on the (virtual) disks.

3. [The last lines demonstrate mapping](#) "DISKS" HSD50 storage controller to disk images, shared between both hosts. A "container" parameter is used for this purpose. This example assumes that all disk images are accessible from both host machines via Microsoft share or some other realization. It is also possible to create a network drive and use it in the CHARON VM configuration files above.

## Example 2: Triple node DSSI cluster with multiple iSCSI disks

In this example we assume that all three host systems have an iSCSI initiator and are connected to a common iSCSI server. The iSCSI disk server provides 8 virtual disks with R/W access on all hosts. These disks are configured as "\\.\PhysicalDrive0" to "\\.\PhysicalDrive7" on each of the host machines.

The storage configuration must be identical on all three nodes. It is recommended to describe the storage structure in a separate configuration file to be included in each CHARON VM configuration file with the use of the "include" instruction (name of the configuration file set to "disksets.cfg" in this example) and store it on a common network share ("\\DiskServer\Share"):

```
load HSD50 DISKS1 dssi_host=PAA dssi_node_id=4
set DISKS1 scs_system_id=3238746238 mscp_allocation_class=1

set DISKS1 container[1]="\\.\PhysicalDrive0"
set DISKS1 container[2]="\\.\PhysicalDrive1"
set DISKS1 container[3]="\\.\PhysicalDrive2"
set DISKS1 container[4]="\\.\PhysicalDrive3"

load HSD50 DISKS2 dssi_host=PAA dssi_node_id=5
set DISKS2 scs_system_id=1256412654 mscp_allocation_class=2

set DISKS2 container[5]="\\.\PhysicalDrive4"
set DISKS2 container[6]="\\.\PhysicalDrive5"
set DISKS2 container[7]="\\.\PhysicalDrive6"
set DISKS2 container[8]="\\.\PhysicalDrive7"
```

CHARON VM configuration file for the emulated VAX 4000 Model 108 node running on *HOST001* is as follows:

```
...
set PAA port[2]=11012 host[2]="host002:11021"
set PAA port[3]=11013 host[3]="host003:11031"

include \\DiskServer\Share\disksets.cfg
...
```

CHARON VM configuration file for the emulated VAX 4000 Model 108 node running on *HOST002* is as follows:

```
...
set PAA port[1]=11021 host[1]="host001:11012"
set PAA port[3]=11023 host[3]="host003:11032"

include \\DiskServer\Share\disksets.cfg
...
```

CHARON VM configuration file for the emulated VAX 4000 Model 108 node running on *HOST003* is as follows:

```
...
set PAA port[1]=11031 host[1]="host001:11013"
set PAA port[2]=11032 host[2]="host002:11023"

include \\DiskServer\Share\disksets.cfg
...
```



# CHARON-VAX for Windows CI cluster

## Table of Contents

- Introduction
- General description
- Configuration steps
- Example 1: Dual node CI cluster with 4 shared disks
- Example 2: Triple node CI cluster with multiple iSCSI disks

## Introduction

This section describes how to configure a CHARON-VAX for a Windows CI cluster.

## General description

A virtual CIXCD is the functional equivalent of a hardware CIXCD host adapter with the exception that there is no physical layer to connect to a hardware CI infrastructure. The current hardware is much faster than the physical CI implementation, therefore if such a connection were even possible, it would limit the virtual system throughput.

For data storage, the CIXCD connects to one or more virtual HSJ50 controllers that are loaded as separate components in the configuration file. To configure VAX CI clusters, the virtual CIXCDs of the multiple CHARON virtual machines (VMs) are interconnected via TCP/IP links.

It is advisable to start any field test based on the cluster examples provided below

Configuring virtual CI clusters requires many configurable parameters and these parameters need to be the same on all servers.

The current CI implementation for CHARON-VAX/66x0 supports up to 8 VAX VMs in a virtual CI cluster and handles a maximum cluster size of 128 nodes. A single virtual CI network supports up to 256 storage elements.

For more details on CI configuration follow [this link](#).

## Configuration steps

To create a CHARON-VAX CI cluster, both of the following elements must be configured:

1. "CIXCD" host adapter
2. "HSJ50" storage controller

CI hardware topology is emulated by establishing TCP/IP channels between the emulated CIXCD host adapters of each CHARON VM. The emulated HSJ50 storage controllers are then connected to every CIXCD host adapter in the virtual CI network.

Cluster operations require that (virtual) disks are simultaneously accessible by all CHARON VMs involved. This can be implemented, for instance, by using a properly configured iSCSI initiator / target structure or a fiber channel storage back-end. Disks on a multiport SCSI switch are not acceptable - because a SCSI switch does not provide true simultaneous access to multiple nodes.

**It is advisable to start any field test with implementing the cluster examples provided below.**

## Example 1: Dual node CI cluster with 4 shared disks

In this example, setting up two CHARON VMs, two host machines, preferably running the same version of Windows, are required.

Assume that these host systems have network host names *CASTOR* and *POLLUX* in the host TCP/IP network.

The following are CHARON VM configuration files for the emulated VAX 6610 nodes running on *CASTOR* and *POLLUX*:

```

CASTOR node
...
load CIXCD PAA ci_node_id=1
set PAA port[2]=11012 host[2]="pollux:11021"
load HSJ50 DISKS ci_host=PAA ci_node_id=101
set DISKS scs_system_id=3238746238 mscp_allocation_class=1
set DISKS container[0]="\\DiskServer\Share\dua0-rz24-vms-v6.2.vdisk"
set DISKS container[1]="\\DiskServer\Share\dua1-rz24-vms-v6.2.vdisk"
set DISKS container[2]="\\DiskServer\Share\dua2-rz24-vms-v6.2.vdisk"
set DISKS container[3]="\\DiskServer\Share\dua3-rz24-vms-v6.2.vdisk"
...

```

```

POLLUX node
...
load CIXCD PAA ci_node_id=2
set PAA port[1]=11021 host[1]="castor:11012"
load HSJ50 DISKS ci_host=PAA ci_node_id=101
set DISKS scs_system_id=3238746238 mscp_allocation_class=1
set DISKS container[0]="\\DiskServer\Share\dua0-rz24-vms-v6.2.vdisk"
set DISKS container[1]="\\DiskServer\Share\dua1-rz24-vms-v6.2.vdisk"
set DISKS container[2]="\\DiskServer\Share\dua2-rz24-vms-v6.2.vdisk"
set DISKS container[3]="\\DiskServer\Share\dua3-rz24-vms-v6.2.vdisk"
...

```

Let's review both configurations step-by-step.

1. The first two lines of both configuration files load and establish parameters for the "PAA" CIXCD host adapter. Only 3 CIXCD parameters are important for us in this situation:

Parameter	Description
ci_node_id	An integer value that specifies the address of the virtual CIXCD host adapter on the virtual CI network. Possible values range from 0 through 127 (Initially set to 127).
port	An integer value that specifies the TCP/IP port number at which the emulated CIXCD host adapter listens for connections from another emulated CIXCD host adapter with a certain CI node id. Possible values range from 1024 through 32767.
host	A string value that specifies the TCP/IP host name (and optional TCP/IP port number) to connect to another emulated CIXCD host adapter with certain CI node.  The syntax for the string is "host-name[:port-no]" with possible values for "port-no" in the range from 1024 through 32767.

Thus, *CASTOR* connects to *POLLUX*'s port 11021 and listens for *POLLUX*'s connection on port 11012, *POLLUX* connects to *CASTOR*'s port 11012 and listens for *CASTOR*'s connection on port 11021

2. The third and fourth lines of both configuration file "DISKS" HSJ50 storage controller and its parameters:

Parameter	Description
ci_host	A string value that specifies an instance name of the emulated CIXCD host adapter serving the virtual CI network.  If this value is not set, CHARON VM tries to locate the host adapter automatically. This automatic lookup works only if the CHARON VM configuration has exactly one instance of an emulated CIXCD host adapter.
ci_node_id	An integer value that specifies the address of the emulated HSJ50 storage controller on a virtual CI network. Possible values are from 0 through 7 (initially set to 0).
scs_system_id	A string value that specifies the SCSNODENAME of the emulated HSJ50 storage controller.  The string consists of up to 10 characters. Possible characters are uppercase letters A through Z and the integers 0 through 9.
mscp_allocation_class	An integer value that specifies the <i>ALLOCLASS</i> of an emulated HSJ50 storage controller.  Possible values range from 0 through 255 (initially set to 0).

In both configuration files, the data related to the emulated HSJ50 storage controller, the "DISKS" **must be identical**. Not following this rule can cause data corruption on the (virtual) disks.

3. The next lines demonstrate the mapping of the "DISKS" HSJ50 storage controller to the disk images, shared between both hosts. A "container" parameter is used for this purpose. This example assumes that all disk images are accessible from both host machines via a network share or some other realization.

## Example 2: Triple node CI cluster with multiple iSCSI disks

In this example we assume that all three host systems have an iSCSI initiator and are connected to a common iSCSI server. The iSCSI disk server provides 8 virtual disks with R/W access on all hosts. These disks are configured as "\\.\PhysicalDrive0" ... "\\.\PhysicalDrive7" on each of the host machines.

The storage configuration must be identical on all three nodes, it is recommended to describe the storage structure in a separate configuration file to be included in each CHARON VM configuration file with the "include" instruction (in this example the name of the configuration file set to "disksets.cfg") and store it on a common network share ("\\DiskServer\Share"):

```
load HSJ50 DISKS1 ci_node_id=4

set DISKS1 scs_system_id=3238746238 mscp_allocation_class=1

set DISKS1 container[1]="\\.\PhysicalDrive0"
set DISKS1 container[2]="\\.\PhysicalDrive1"
set DISKS1 container[3]="\\.\PhysicalDrive2"
set DISKS1 container[4]="\\.\PhysicalDrive3"

load HSJ50 DISKS2 ci_node_id=5

set DISKS2 scs_system_id=1256412654 mscp_allocation_class=2

set DISKS2 container[5]="\\.\PhysicalDrive4"
set DISKS2 container[6]="\\.\PhysicalDrive5"
set DISKS2 container[7]="\\.\PhysicalDrive6"
set DISKS2 container[8]="\\.\PhysicalDrive7"
```

CHARON VM configuration file for the emulated VAX 6610 node running on *HOST001* is as follows:

```
...
load CIXCD PAA ci_node_id=1

set PAA port[2]=11012 host[2]="host002:11021"
set PAA port[3]=11013 host[3]="host003:11031"

include \\DiskServer\Share\disksets.cfg
...
```

CHARON VM configuration file for the emulated VAX 6610 node running on *HOST002* is as follows:

```
...
load CIXCD PAA ci_node_id=2

set PAA port[1]=11021 host[1]="host001:11012"
set PAA port[3]=11023 host[3]="host003:11032"

include \\DiskServer\Share\disksets.cfg
...
```

CHARON VM configuration file for the emulated VAX 6610 node running on *HOST003* is as follows:

```
...
load CIXCD PAA ci_node_id=3

set PAA port[1]=11031 host[1]="host001:11013"
set PAA port[2]=11032 host[2]="host002:11023"

include \\DiskServer\Share\disksets.cfg
...
```

# CHARON-VAX and CHARON-PDP for Windows licensing

Please refer to the other sections of the *Charon Licensing Handbook* and your product documentation for any topics not covered in this overview guide and for more detailed information on the topics discussed here.

This page provides an overview of the basic steps to set up the licensing environment on a system running Charon-AXP or Charon-VAX for Windows:

- [General Description](#)
  - [Charon License Checks](#)
  - [Charon License Updates](#)
  - [License Content](#)
- [License Type Overview](#)
  - [Local Hardware License](#)
  - [Software License](#)
  - [Network License](#)
- [Overview of Initial License Installation Steps](#)
- [Overview of License Management Tools](#)
  - [Accessing the License Management Tools on Charon Version 4.7 and older](#)
  - [Accessing the License Management Tools on Charon Version 4.8 and Higher](#)
- [Define a Password for the Sentinel Admin Control Center \(ACC\)](#)
- [View Existing Licenses](#)
  - [Viewing a License with Sentinel ACC](#)
  - [Viewing a License with the License Viewing Utility](#)
    - [HASP View Utility Window](#)
- [Installing the License](#)
  - [Preparation](#)
  - [Installing a Local Hardware License](#)
  - [Installing a Software License](#)
  - [Installing a Network License](#)
    - [Firewall Considerations](#)
    - [Allowing Client Access on the License Server](#)
    - [Allowing Access to a License Server on the Client](#)
  - [Verifying License Availability](#)
- [Handling Multiple Keys](#)
  - [Backup License Characteristics](#)
  - [Primary and Backup License for Charon-AXP/VAX prior to Version 4.9](#)
  - [Prioritized List of License Keys or License IDs for Charon-AXP/VAX 4.9 and Higher](#)
    - [Prioritized List of License Keys](#)
    - [Prioritized List of Product License IDs](#)
- [Update an Existing License](#)
- [Removing a License](#)
  - [Removing a Hardware License from a System](#)
  - [Removing a Software License from a System](#)
  - [Removing Access to Network Licenses from a System](#)
    - [Full Removal of Network License](#)
    - [Block Access of Remote Emulator Hosts to Network License](#)
    - [Block Access to Network Licenses on Client System](#)
- [Manual Sentinel HASP License Driver Installation](#)
  - [New License Driver Installation](#)
  - [Replacing the Currently installed License Drivers](#)

## General Description

A Charon emulator product will not run without a valid license. For Sentinel HASP licenses (the topic of this chapter) this license can be provided on a **USB hardware dongle** (a Sentinel HASP key) or as a **software license** bound to the hardware of the host system or the network license server. A loss or defect of a license will cause the emulator to stop.

For extra protection, Stromasys recommends the use of a backup license key (purchased separately) that can replace the main license key for a limited period of time. It is possible to specify the backup license in the Charon configuration file to prevent the emulator from stopping in case its main license is no longer accessible.

To check the validity of a license, Charon performs the following steps:

- The Charon emulator checks the text block of the license to determine if it is allowed to run (if there is a product section which corresponds to the emulator).
- Then Charon checks the feature associated with the product to see if the feature is valid - that is, it is not expired, the available runtime hours are not exhausted, and if there are still some instance-slots available to run the emulator.
- If there is no matching product section or if the feature is not valid, Charon reports the problem in the log and shuts down.

## Charon License Checks

Charon products **check the availability of a valid license** at different points in time:

### 1. At startup:

If no valid license is found, an error message will be written to the emulator log file and the emulator will not start.

Charon-AXP/VAX has a configuration parameter to define the number of retries (`license_key_lookup_retry`). Please refer to the product configuration documentation for more details.

**2. At regular intervals** during the runtime of the emulator (the default license check period of 1 hour can be changed by Stromasys using the appropriate license parameters):

- If the previously used valid license has been removed, has disappeared, is defect, or has become invalid, the emulator will report the loss of the license in the log file and continue operation for a limited amount of time as described below.
- If there is another valid license, for example a backup license defined in the configuration file, it will be used.
- Starting with Charon-AXP/VAX 4.9, Charon allows for a grace period of 12 hours during which the software checks for the presence of a valid license every 10 minutes until a valid license is found. If no valid license is found after the grace period has expired, the emulator will stop.
- Earlier versions of the above products and other emulator products: should there still be no valid license at the next regular license check (this default interval may be changed by individual Charon products), the emulator will stop.
- If a time-restricted license is used and it expires, the Charon instance tries to find its replacement automatically and, if found, proceeds using the replacement license.

**Important note for Windows installations:** any running emulators will be stopped if the license service is forced to stop despite existing service dependencies (i.e., running emulators). So any running guest systems must be cleanly shutdown before restarting the license service in order to avoid data loss. This is the behavior at the time of writing. It may change in future versions.

## Charon License Updates

Under certain circumstances existing licenses must be updated.

**License updates are required, for example, in the following cases:**

- Charon product version upgrade
- Product change
- License validity (time) extension

License updates, for example a new expiration date or a new execution counter, can be performed without interrupting operation, that is, without shutting down active Charon instances, **provided that the resulting license is valid for the running Charon instance**. If any parameters of the updated license are not valid for the running Charon instance (for example, the wrong model or version), the Charon instance will stop running.

Licenses are **backwards compatible to some extent**. A **license update to a more recent version** will allow some previous Charon versions to run as well. The details depend on the Charon product.

The license check on **Charon-AXP/VAX/PDP starting with version 4.4** will be successful for a newer license if the major version (for example, the 4 in 4.7) is the same. In **version 4.3**, the emulator will start, but there are inconsistencies and incompatibilities especially in the associated tools (e.g., `hasp_view`). Product **versions 4.2 and earlier** are incompatible with the current license schema.

# License Content

The license data on a license key, independent of whether it is a hardware or software license key, consists of **several parts**: the **license key text block** contains the **license section** and **one or more product sections**. The **license features** describe additional conditions and limitations for the emulator products contained on the license.

## A) License Key Text Block

### 1) License Section (one per license key)

The **license section** contains general license information related to ownership and administration.

There is **one license section** per license key.

Please note that in rare cases, you may receive an **empty USB dongle** for which command outputs will show that **no license information was detected**. In such cases, request a license update from Stromasys as described in *Updating a License*.

#### Relevant license section parameters:

- **License type**: type of the license, i.e., hardware or software license, network-enabled or local (added in newer Charon product versions, e.g., Charon-AXP/VAX 4.9)
- **License number**: assigned by Stromasys
- **License key ID**: unique; pre-programmed; unmodifiable; on physical keys same value as the Physical Key ID
- **Master key ID**: the software vendor ID used by Stromasys for issuing licenses
- **Release date**: the day on which the license was issued
- **Release time**: the time at which the license was issued
- **Update number**: the number of license updates that have been applied
- **End user name**: the name of the person to whom the license was issued
- **Purchasing company name**: usually the company to which the license was issued

### 2) Product Section (one or more per license key)

The text block also contains **product sections** that contain product specific licensing information.

- Several products can be combined on one license key. Stromasys may place restrictions on what products can be combined on a single license key.
- The product section also may include information about the support contract and the end of support. It may contain hardware specific limitations related to the types of devices that can be connected (e.g., serial interfaces).
- A checksum is inserted after the last product section of a license. This checksum is not visible in the show command output. Charon products/versions requiring such a checksum will not start if the license does not contain a checksum.
- Each product section is associated with a **feature defining additional conditions** for the product.

#### The following list shows the most important parameters of a product section.

Please note that not all of these parameters may be present for a particular Charon product.

- **Product License Number**: identification of specific product section (optional; new starting with Charon-AXP/VAX 4.9 and Charon-PAR 1.10). Corresponds to configuration parameter **license\_id**.
- **Virtual Hardware**: range of virtual models available for running
- **Product Name**: commercial product name
- **Product Code**: commercial product code
- **Major Version, Minor Version, Maximum Build, Minimum Build**: commercial product version and range of build numbers suitable for running with this license. The parameters **Maximum** and **Minimum Build** are no longer displayed starting with Charon-AXP/VAX 4.9 and Charon-PAR 1.10. **Please note** that, depending on the product and product version, licenses have a limited backward compatibility (i.e., when the version on the license is newer than the product version).  
See *Operational Procedures Overview* in the [Sentinel HASP Licensing Documentation](#) for more information.
- **Host CPU supported or 32bit and 64bit OS requirements**: type of host CPU required (for example, x64)
- **Host Operating System required**: Linux or Windows
- **CPUs allowed**: number of virtual CPUs enabled for virtual multiprocessor systems
- **Maximum virtual memory**: 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.
- **CHAPI enabled**: whether or not CHAPI (Charon-AXP API) can be used with this product
- **Instances allowed**: maximum number of Charon emulator instances of this program/hardware model that are allowed to run concurrently on the local host. Relevant for local license keys that do not support the process count parameter. **See also** the parameter *Number of processes* in the feature section.



- **CPU Speed:** optional parameter to reduce the speed at which the program can run to a fraction of the maximum (e.g., to 25%)
- **HPSUSAN:** optional parameter. Specific to HP3000, overrides physical key ID. The HPSUSAN (HP System Unique Serially Assigned Number) is used for licensing purposes by HP. The parameter may be needed if a license on the physical system is tied to the original HPSUSAN.
- **Feature number:** number of the feature defining further limits/restrictions for the product license. Each product can use a separate feature, or the same feature can be used by several products.  
If using a **network license** and the same feature is shared by several products, the number of instances allowed by the feature is shared between the products (e.g., 2 allowed instances enable one Charon-AXP and one Charon-SSP instance to run concurrently). If using a **local license dongle** and the same feature is shared by several products, the number of allowed instances is counted separately for each product.  
The parameters defined in a feature are described below in the *Features* section.

## **B) License Feature Section** (one or more per license key)

Features define **product specific conditions**, for example expiration date and number of concurrent instances.

The following list describes these conditions:

- **Released product expiration date:** for time-based licenses. Note that at the time of writing the license of Charon-AXP/VAX products will expire at the end of the day of the expiration date, the license of Charon-SSP in the early morning of the expiration day. The exact behavior of the different versions can change without prior warning. Not supported for HL-MAX dongles.
- **Runtime counter:** number of license checks after which the license will be disabled. Default check interval is one hour. The counter will be decreased at the start of the emulator and then at every license check interval. Once the counter has been depleted, the license becomes invalid. This license type is typically used on licenses intended as backup licenses.
- **Login interval or Check interval:** frequency of license checking during emulator execution (default 1 hour). This parameter is not visible in the show command output in older Charon product version. It is visible in the show command output of newer versions, starting with Charon-AXP/VAX 4.9 and Charon-PAR 1.10.
- **Network license:** indicates that a physical license is a network license
- **Number of processes:** maximum overall number of Charon instances that are allowed to run concurrently.  
This parameter is set by Stromasys to the value matching the respective customer contract. For network dongles, the maximum possible value of this parameter is determined by the type of network dongle. For example: on a NetTime10 dongle this parameter can be set to a value between 1 and 10 (depending on the customer contract). The type of dongle can be viewed via Sentinel ACC.

### **Example with Time-Based Parameters**

The parameters below describe the following limitations to the emulator:

- License permits running up to 2 instances of Charon at the same time.
- License expires on May 31, 2020, **or** after 360 license checks (with an interval of 60 minutes between the checks).

```
Expiration date: 31/5/2020
Runtime counter: 360
Login interval: 60 minutes
Instances allowed: 2
```

If any of the conditions are no longer met (too many instances, past expiration date, all checks exhausted), the feature becomes invalid and will not allow an emulator instance (or an additional instance) to run.

## License Type Overview

Depending on customer requirements, Stromasys can provide several different license types.

### Local Hardware License

Local hardware licenses are USB dongles and work on the system they are physically connected to. For Windows systems supporting AnywhereUSB®, hardware licenses can also be connected over the network (similar products are available for Linux). They can easily be moved to a different system, if required. The content of the dongle can be updated if a change to the license or an extension of a time-based license is required. The necessary steps are described later in this document.

**Please note:**

- Hardware dongles require the Sentinel HASP run-time (driver) installation before the dongle can be connected to and used by the system.
- Hardware dongles, apart from HL-MAX dongles, are equipped with a **battery** and a clock, which makes them independent of the host clock. The battery is not rechargeable. However, the dongle can use the power provided by the host system while it is plugged in. By doing this, the depletion of the battery can be slowed down. Check the dongle at regular intervals if it is not permanently connected to a system. If the battery becomes completely depleted, the dongle will be permanently unusable and must be replaced. See also: [How long does the license USB dongle battery last upon a full charge.](#)
- Hardware dongles are not suited for running Charon emulator products in **cloud environments**. Stromasys provides other licensing solutions adapted to virtual environments (Charon VE licenses) for selected products. See [Virtual Environment \(VE\) License Server Documentation](#).

### Software License

A software license is a "virtual" key with functionality very similar to a HASP network-enabled hardware dongle.

A software license does not require any special hardware but it still **requires installation of the Sentinel runtime environment**.

**Important information:**

- To avoid unexpected problems, do not use any Sentinel runtime software that was not provided by Stromasys without being advised to do so by your Stromasys representative.
- Software licenses are best suited for stable environments, because their correct function depends on certain characteristics of the host system. **Changing these characteristics will invalidate the license.**
  - If the Charon host runs on **real hardware**, software licenses are by default tightly bound to the hardware for which they were issued. If major hardware characteristics of the system are changed, the license will be disabled.
  - If the Charon host runs in a **virtual environment (e.g., VMware)**, software licenses are normally bound to the virtual machine ID and a set of additional characteristics of the virtual machine. If any of these parameters are changed, the license will be disabled.
  - For a more detailed description of the restrictions, please refer to [Software Licensing restrictions](#) or contact your Stromasys representative.
- Software licenses are very sensitive to even small changes on the host system. Therefore, it is especially important to provide for a backup license that will ensure continued operation should there be a problem with the software license. See [Handling Multiple License Keys](#) for details.
- Standard software licenses are not suited for running Charon emulator products in **cloud environments**. Stromasys provides other licensing solutions adapted to virtual environments (Charon VE licenses) for selected products. See [Virtual Environment \(VE\) License Server Documentation](#).

### Network License

The network Sentinel HASP key (**red USB dongle**) can be shared between several hosts running a Charon emulator product (including the host on which the network license is installed).

**Please note:** all software licenses are also network licenses.

If the Charon emulator product is installed on the host where the network license is connected, no additional steps are required. The Sentinel driver is installed as part of the Charon product installation. If the host does not have a Charon emulator product installed, the host can still distribute the connected network license to emulator instances running on other hosts.

The Sentinel HASP runtime software must be installed on such a license server and on the client system. For details regarding the installation, please refer to the software installation section in this document. Once both the Sentinel runtime software and the network license are installed, the Charon emulator product can be started on any appropriate client host on the LAN, provided access to the license is enabled.

- The network license will be visible to all hosts that can access the license server over IP. Access to the license server must be possible on **port 1947** via **UDP** (discovery process) and **TCP** (actual access to license). Further information (e.g., use of **additional UDP ports**) can be found in the section *Firewall Considerations*.
- The license server and the client must both allow access to the network license using the appropriate management tools.

The maximum **number of concurrently active** Charon instances is determined by the parameters of the license.

**Please note:** Charon-AXP/VAX version 4.7 build 171-01 introduced a change with respect to network licenses. In previous versions of Charon-AXP/VAX a network license controlled the maximum number of client host systems and Charon instances per host system (station/instance mode). In the current Charon-AXP/VAX versions a network license controls the maximum overall number of active instances, which can be distributed across client host systems according to the preference of the customer.

## Overview of Initial License Installation Steps

Each Charon emulator product requires a valid license to run. To access a Sentinel HASP license, the emulator needs the Sentinel runtime software. This software is part of the Charon-AXP/Charon-VAX for Windows installation kits. Refer to the software installation section of the Charon product user's guide for software installation instructions.

After the installation of this software, the license can be installed on the system. The following steps are described in more detail below:

1. Add a password for the Sentinel Admin Control Center (ACC).
2. If you purchased a **hardware license**, you can simply plug the dongle into a free USB port on the system.
3. If you purchased a **software license**, you need to create a fingerprint file in C2V (customer-to-vendor) format containing the system characteristics. Use this file to request a license for your Charon product from Stromasys.
4. If your license is a **network license** served by a license server, make sure that the access of the client system to the license server is not blocked by the configuration or a firewall.
5. Optionally, define how **multiple licenses** will be used (selecting primary/backup license, defining license priorities).
6. If you have an **existing license** that needs to be updated, you need to create a customer-to-vendor (C2V) file and use this file to request a license update from Stromasys.

## Overview of License Management Tools

The following list shows the main tools used to manage licenses on Windows:

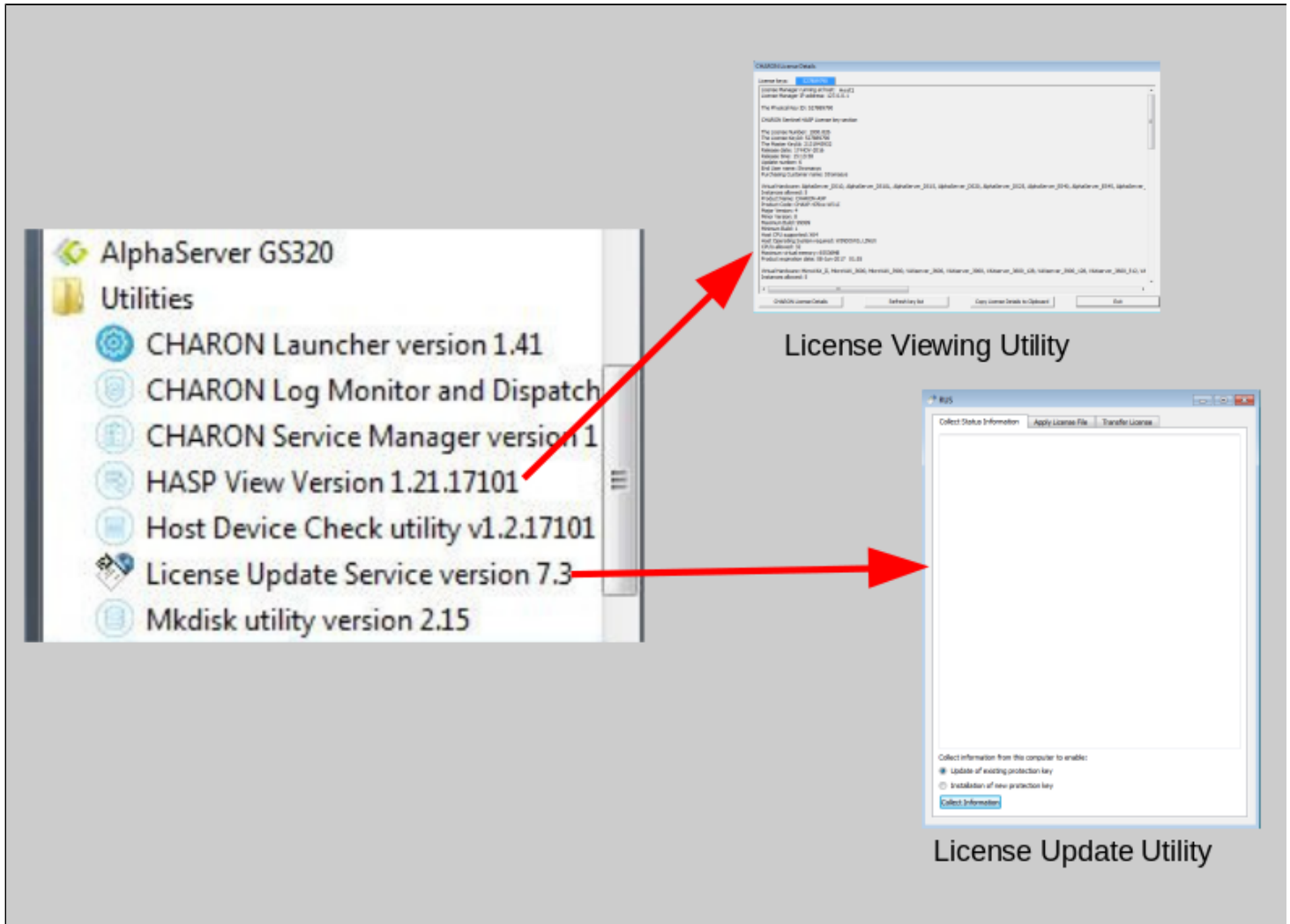
- **Sentinel Admin Control Center (ACC):** A web-based interface providing important configuration options with respect to licenses.
- **The license viewing utility:** A GUI-based tool to display the detailed license contents. Cannot be used over a remote connection when using local hardware licenses.
- **The license update utility:** A GUI-based tool to extract the information required for creating and updating licenses, and to install the resulting license data.

# Accessing the License Management Tools on Charon Version 4.7 and older

The **Sentinel ACC** on the local system is accessed by starting a web-browser and pointing it to the URL: **http://localhost:1947**.

The **license viewing utility** and the **license update utility** are accessed via

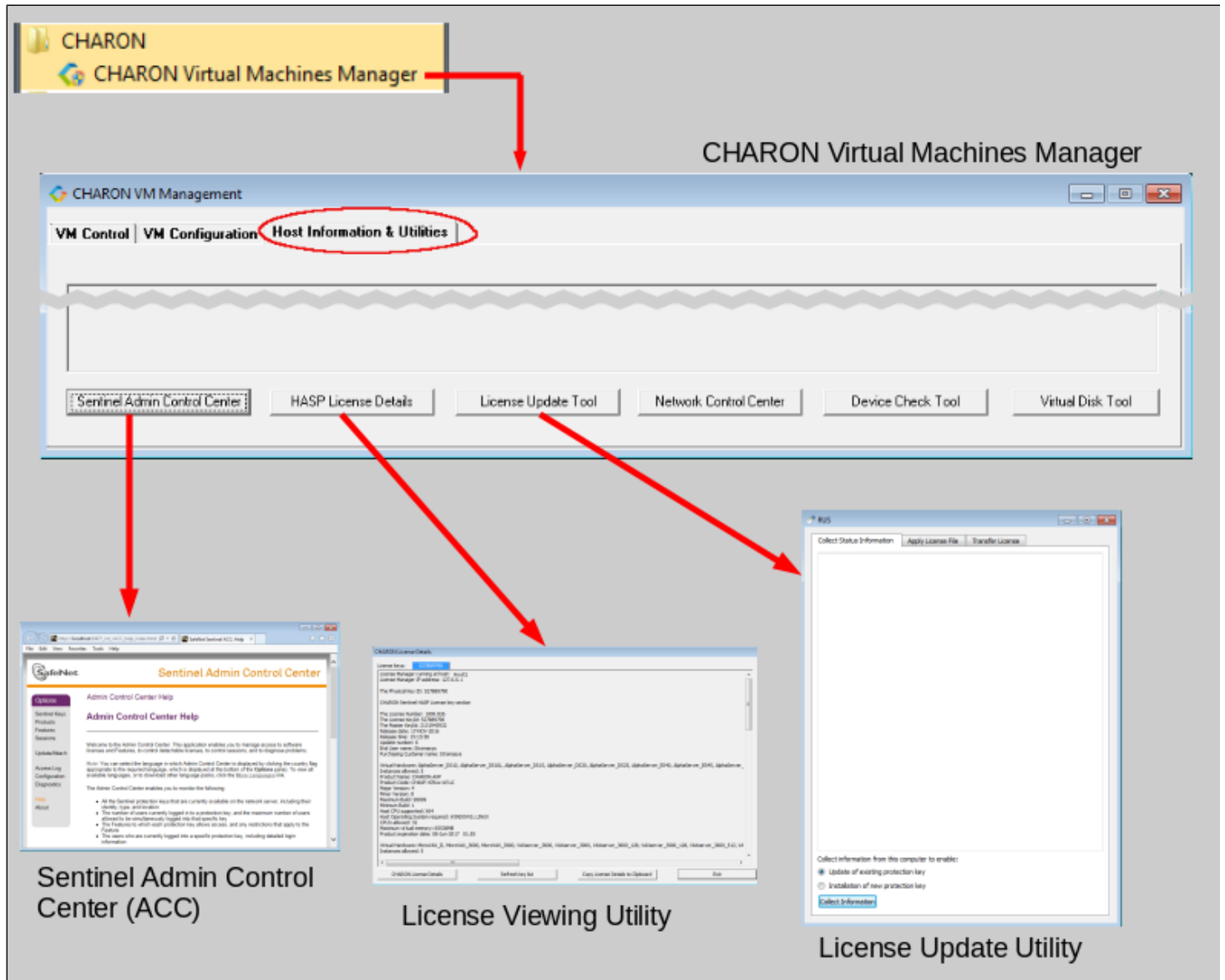
**Start > All programs > Charon > Charon product folder > Utilities** as shown in the image below.



# Accessing the License Management Tools on Charon Version 4.8 and Higher

All license management tools are accessed via the Charon Virtual Machines Manager (VMM) as shown in the image below.

The CHARON VMM is started via **Start > Charon > CHARON Virtual Machines Manager** or by clicking on the associated desktop icon.



## Define a Password for the Sentinel Admin Control Center (ACC)

**Please note:** Stromasys strongly recommends performing the steps described below to reduce the risk of unauthorized access to the Sentinel ACC.

By default, anyone on the local system with access to port 1947 can access the GUI. If remote access is enabled, users on the network with access to port 1947 can also access the GUI. To protect access to the GUI with a password, perform the following steps:

Step	Description
1	Open a web browser and navigate to <b>http://localhost:1947/</b> .
2	<b>Click</b> on the left-hand menu item labeled <b>Configuration</b> .
3	<b>Click</b> on the <b>Basic Settings</b> tab.
4	Under the entry labeled <b>Password Protection</b> , <b>click</b> the <b>Change Password</b> button.
5	At the <b>Change Password</b> window: <ul style="list-style-type: none"> <li>• Leave the <b>Current Admin Password</b> field blank (there is no password set by default).</li> <li>• Enter the desired password into the <b>New Admin Password</b> field.</li> <li>• Repeat the desired password in the <b>Re-enter new Admin Password</b> field.</li> <li>• <b>Click</b> the <b>Submit</b> button.</li> </ul>
6	Back at the <b>Basic Settings</b> tab: <ul style="list-style-type: none"> <li>• Under the section labeled <b>Password Protection</b>, select the <b>All ACC Pages</b> radio button.</li> <li>• <b>Click</b> the <b>Submit</b> button to save this change.</li> </ul>
7	Optional: to allow <b>remote access</b> to the Sentinel HASP GUI: <ul style="list-style-type: none"> <li>• <b>Click</b> the <b>Basic Settings</b> tab.</li> <li>• <b>Select</b> the <b>Allow Remote Access to ACC</b> check box.</li> <li>• <b>Click</b> the <b>Submit</b> button.</li> </ul> <p><b>Additional information for remote access to the Sentinel HASP GUI:</b></p> <ul style="list-style-type: none"> <li>• The firewall must permit access to port 1947/TCP, and</li> <li>• network visibility on the ACC network configuration tab must be set to <b>All Network Adapters</b>.</li> </ul> <p>If required, remote access can also be enabled on Linux by editing the file <code>/etc/hasplm/hasplm.ini</code> and setting the parameter <b>ACCremote</b> to <b>1</b>. Should the file not yet exist, refer to <a href="#">How-to enable remote connection to Sentinel Admin Control Center without GUI</a>. The parameter to enable network visibility is <b>bind_local_only</b>. It must be set to <b>0</b> to enable all adapters. Changed settings are recognized automatically by aksusbd.</p>

## View Existing Licenses

It is important to know which licenses are visible on a system. For example, the user can

- determine if the correct license is installed,
- identify the expiration date on time-based licenses,
- identify the remaining hours of run-time on backup licenses,
- identify the license ID of primary and backup license which are needed for the emulator configuration file.

Licenses can be viewed using two tools:

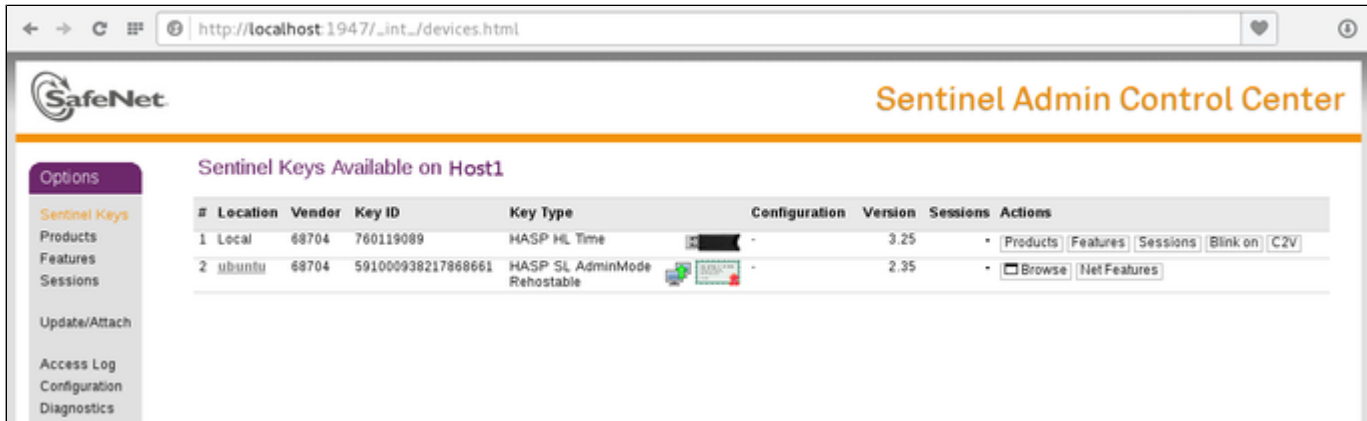
- Sentinel ACC: shows important information, but not the product specific license parameters.
- License viewing utility: shows all product details contained on the license. Can only be run from a local connection for local hardware licenses.

## Viewing a License with Sentinel ACC

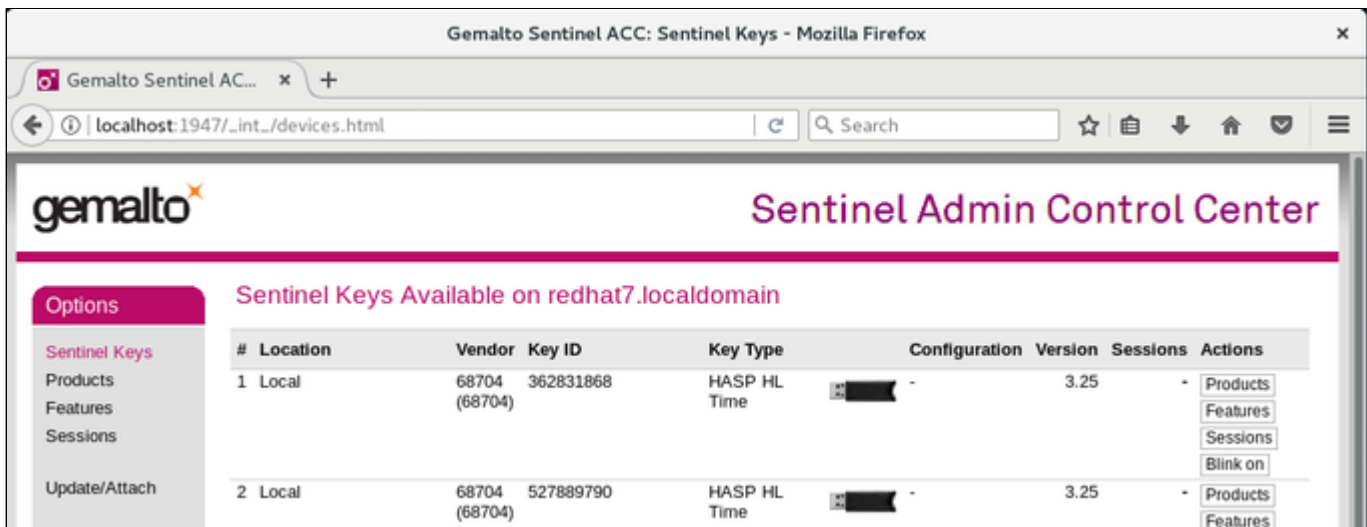
To view available licenses using Sentinel ACC, start the web interface as described above.

**Please note:** starting with Charon-AXP/VAX 4.9 a new Sentinel ACC version ( $\geq 7.60$ ) has been included in the product. The functionality is very similar but the appearance is somewhat different.

o get to the **Sentinel Keys** screen, click on the corresponding menu item or access the URL [http://localhost:1947/\\_int\\_/devices.html](http://localhost:1947/_int_/devices.html) directly. A screen similar to the following opens and displays the available license keys:



For Sentinel drivers version 7.60 and above, this screen looks similar to the following:



This page displays important information about the available licenses, including

- **Location:** Shows whether the license is local or remote. If the license is a network license, the hostname of the remote system is specified. You can access the remote license manager by clicking on the hostname, provided that access is permitted.
- **Key ID:** The unique identification of the license.
- **Key Type:** Hardware keys are marked by the abbreviation HL (hardware license) and a picture of the license dongle. Software licenses are marked by the abbreviation SL (software license).
- **Sessions:** Shows the number of active sessions opened for the specific key.

The buttons on the right-hand side can be used to retrieve more information about the license or to extract the C2V file for a license update. The **Browse** button shown for network licenses will connect to the remote license manager to show the license features.

A C2V file can only be extracted if the license in question is local to the current license manager, i.e., the license manager to which the web browser is connected. You can connect to a remote license manager by clicking on the **Location** column (if connections are allowed).

**Please note:**

- The option to create a C2V file is not available in older versions of the license software.
- Starting with ACC version 7.60, the **option to create C2V files** for USB dongles is only available if it has been **enabled in the basic configuration** section (under the *Configuration* menu item).
- Newer versions of ACC also offer the option to create a fingerprint for a new SW license.



- Some newer versions of ACC show the option on this screen to create a DEMOMA fingerprint. This is a demo key for development purposes and not relevant for Charon products.
- Some newer versions of ACC offer to create a fingerprint for a software license if such a license does not exist yet. Do not use this option. The resulting file has a format that is incompatible with the Stromasys license creation tools. Instead use the tools described in *Installing the License*.

The menu options *Products*, *Features* and *Sessions* on the left-hand side provide the same information as the buttons. However, they show the information for all licenses.

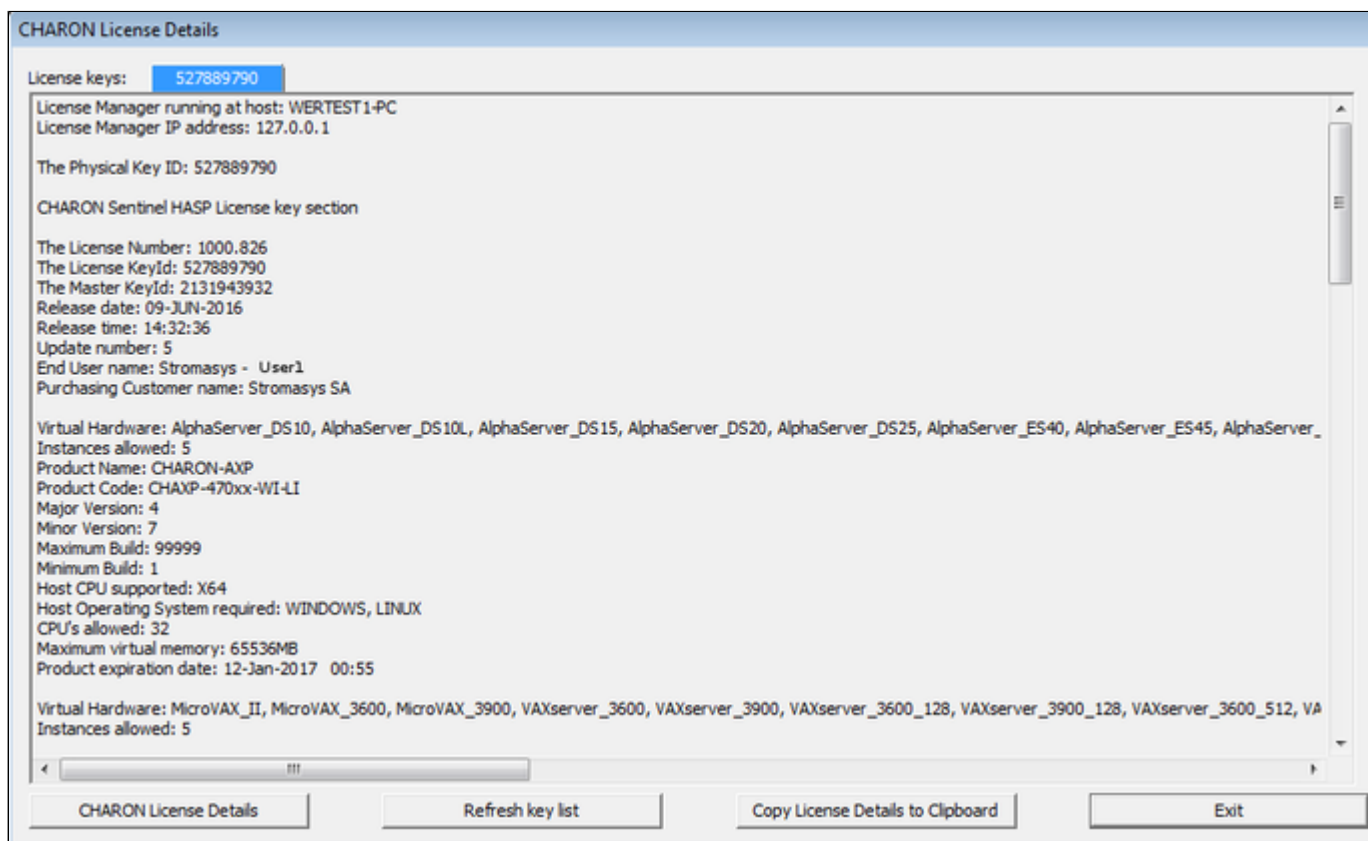
## Viewing a License with the License Viewing Utility

To view available licenses using the license viewing utility, start the program as described above.

Please note: a local hardware key cannot be read via a remote connection (e.g., via RDP). One possible workaround is described in detail in the document [How to read a HASP key over RDP on Windows](#). It shows, how to setup a scheduled task for this purpose.

## HASP View Utility Window

The following image shows sample output:



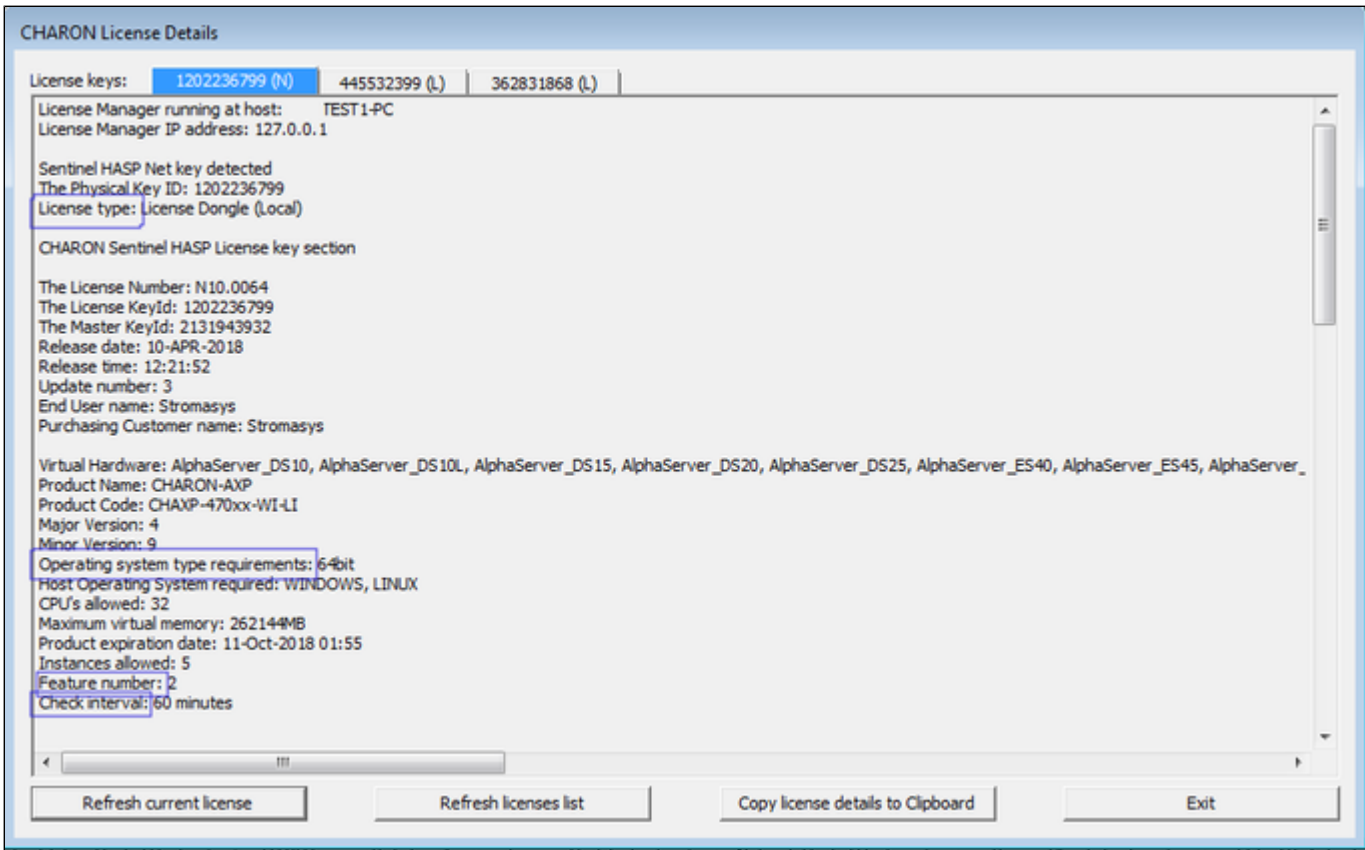
The buttons at the bottom of the window provide additional functions:

- **Charon License Details** (re-)queries the selected license key (the highlighted tab) and displays the contents.
- **Refresh** the key list. This is useful if a key has just been added or removed.
- Copy license details to **clipboard**, for example to paste into a support e-mail.
- **Exit** from the utility.

**If there is more than one license, the data for each license is shown on a separate tab.**

**Please note:** starting with Charon-AXP/VAX version 4.9, the button **Charon License Details** has been renamed to **Refresh current license**. The names of some parameters have changed, new parameters were added, and other parameters removed. See the following sample output:





## Installing the License

This section provides a short overview of the initial license installation. For more in-depth information about licensing, please refer to the licensing handbook.

### Preparation

Before a license can be installed, some preparatory steps are required.

1. Ensure that the correct Sentinel Runtime software is installed. Please refer to the **installation section of your Charon emulator user's guide** for details.
2. If a conflicting or obsolete license is available on the system it may be necessary to **(temporarily) remove or disable** it. If you need to remove a hardware license, simply unplug it. If you need to remove a software license or disable access to a network license, please refer to the relevant chapter in the licensing handbook. Before removing a license or disabling access to it ensure that it is not required by another currently active product.

### Installing a Local Hardware License

A local hardware license (USB dongle) is installed by inserting the USB license key into a free USB port of the host system after the Sentinel runtime software has been installed.

After this step, verify that the license is visible to the system by following the steps for viewing a license as described above. Please bear in mind that a local hardware license cannot be read when connected to the system via a remote connection (e.g., via RDP).

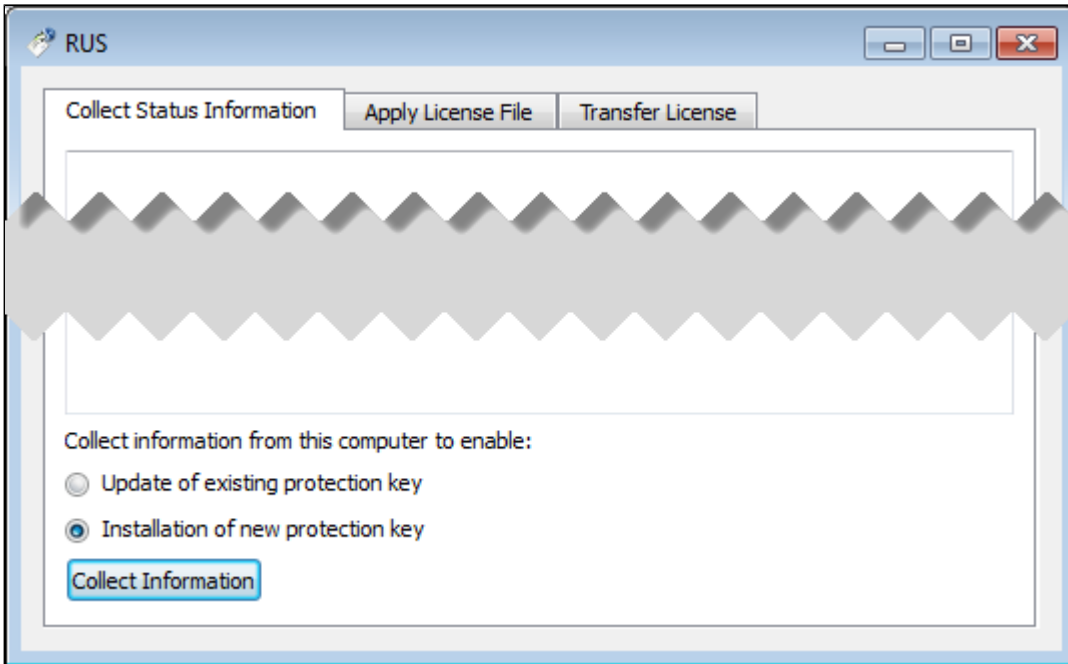
### Installing a Software License

To install a new software license, perform the following steps:

**Step 1:** Start the license update utility as described above and create a fingerprint file:

The license update utility (HASPRUS) looks the same, independent of the product variant used.

The following image shows the **necessary options** for creating a fingerprint file: select the tab **Collect Status Information** and the option **Installation of new protection key**.



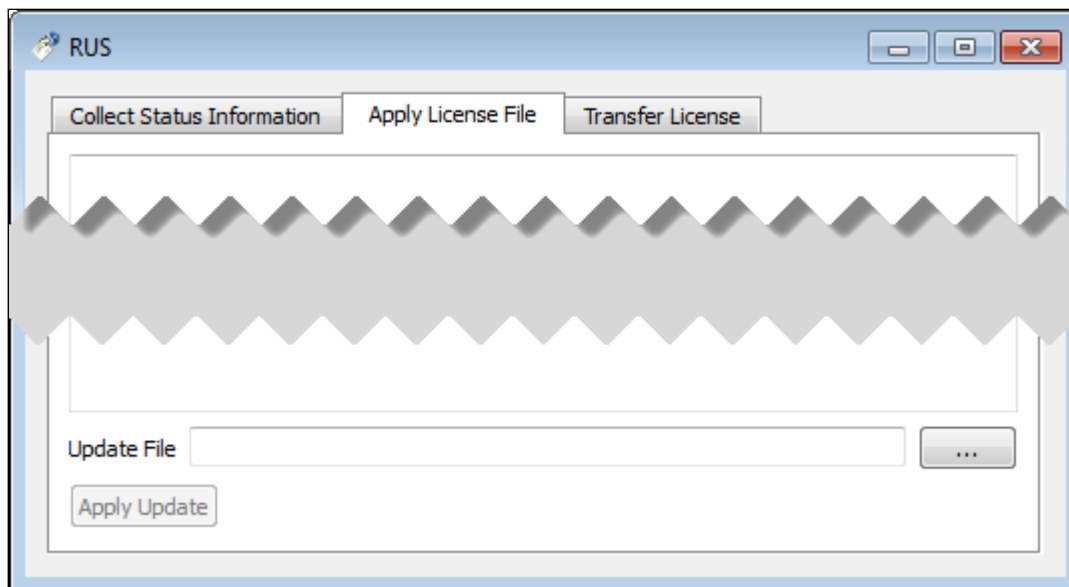
After pressing the **Collect Information** button, you will be prompted to enter a location for storing the fingerprint file. Select a location and filename and save the file.

Once a software license has been installed, current versions of the license update utility do not allow you to create another fingerprint without (temporarily) removing the installed software license. Therefore, save the created fingerprint file in case you need to create an additional software license for the same system (this will be the exception because in most cases the existing license can be updated with the required information). If you don't have the fingerprint file anymore and cannot create a second fingerprint because there is already a valid software license installed, please contact your Stromasys representative or VAR. If you have an obsolete/invalid software license on your system, you can remove it as described in *Removing a License*.

**Step 2:** Send the resulting fingerprint file to Stromasys orders administration using the email address that Stromasys will provide to you.

**Step 3:** After receiving the V2C file from Stromasys, copy the file to the system where the license needs to be installed, start the license update utility as described above, and install the new license:

1. To install the V2C file(s) for hardware **and** software licenses, select the tab **Apply License File**.



2. Press the "... " button next to the **Update File** field. This will open a file browser.

3. Select the V2C file to be applied.

4. Click on **Apply Update** to complete the operation.

**Please note:** This section describes the installation of a new software license. However, the installation procedure for a V2C file is identical when updating a hardware license.

**Important caveat:** When updating a hardware license, you will in most cases receive two V2C files, a **\*\_fmt.v2c** file and a **\*.v2c** file. The **\*\_fmt.v2c** file formats the dongle and the **\*.v2c** file contains the updated license data. In such cases the **\*\_fmt.v2c file** must be applied **first**.

## Installing a Network License

For a network license to be provided to a client host on the network, a license server must have been set up either with a network-enabled hardware license (red dongle) or a software license (software licenses are always network enabled).

If the license server also runs a Charon emulator product, follow the software installation steps in the user's guide. Additionally, the licensing handbook contains additional information about setting up a dedicated license server without installing the full product kit using the license-specific packages of a product kit.

For the client to access a license on a license server the following steps are required:

- Any firewall between license server and client must permit the necessary communication.
- The license server must be configured to allow access from the client.
- The client must be configured to allow access to the license server.

## Firewall Considerations

The following ports are used for the communication between license server and client hosts:

- On the **server side** (where network license has been installed), port 1947 must be open for incoming TCP and UDP traffic to allow client access to the license.
- On the **client side**, traffic is initiated using ports 30000 through 65535 as the source ports and port 1947 as the target port. If broadcast search for remote licenses is to be used, the client must also permit UDP traffic initiated from port 1947 of the license server to ports 30000 through 65535 of the client.

If a host on the network cannot find the license server even though the server is operational, you can **temporarily** disable the firewall to determine whether it blocks the traffic.

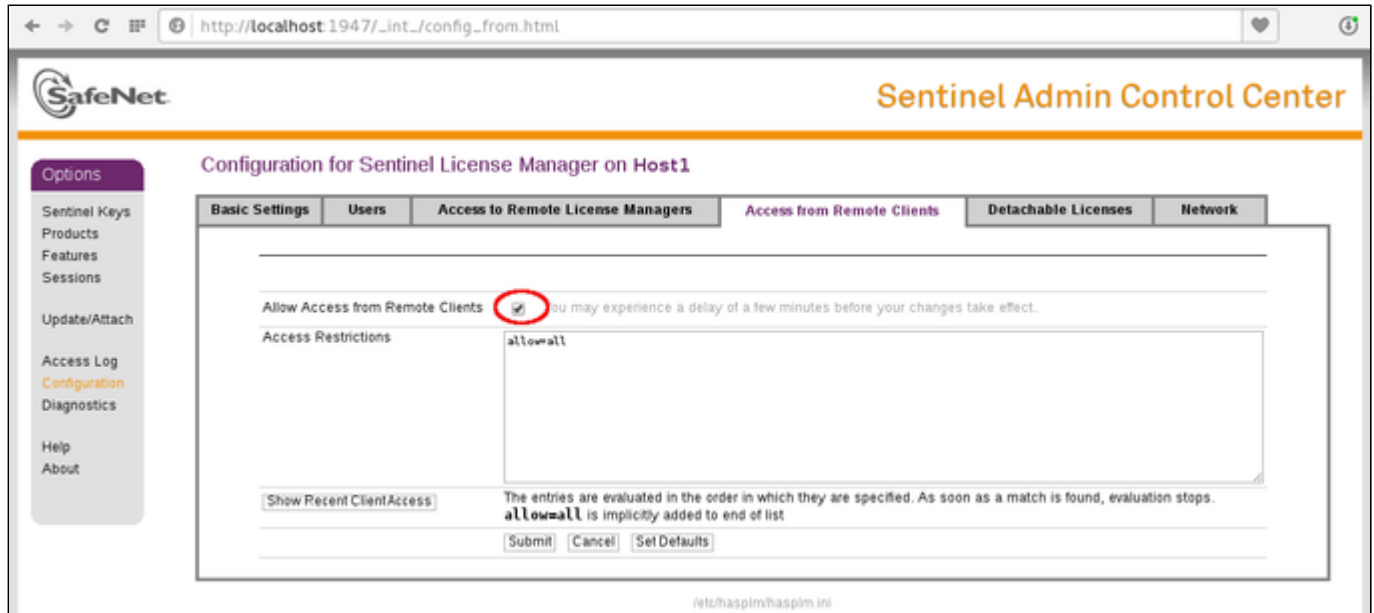
For details on how to configure the firewall in your network, please consult your operating system documentation and make sure to adhere to your company's security policies.

## Allowing Client Access on the License Server

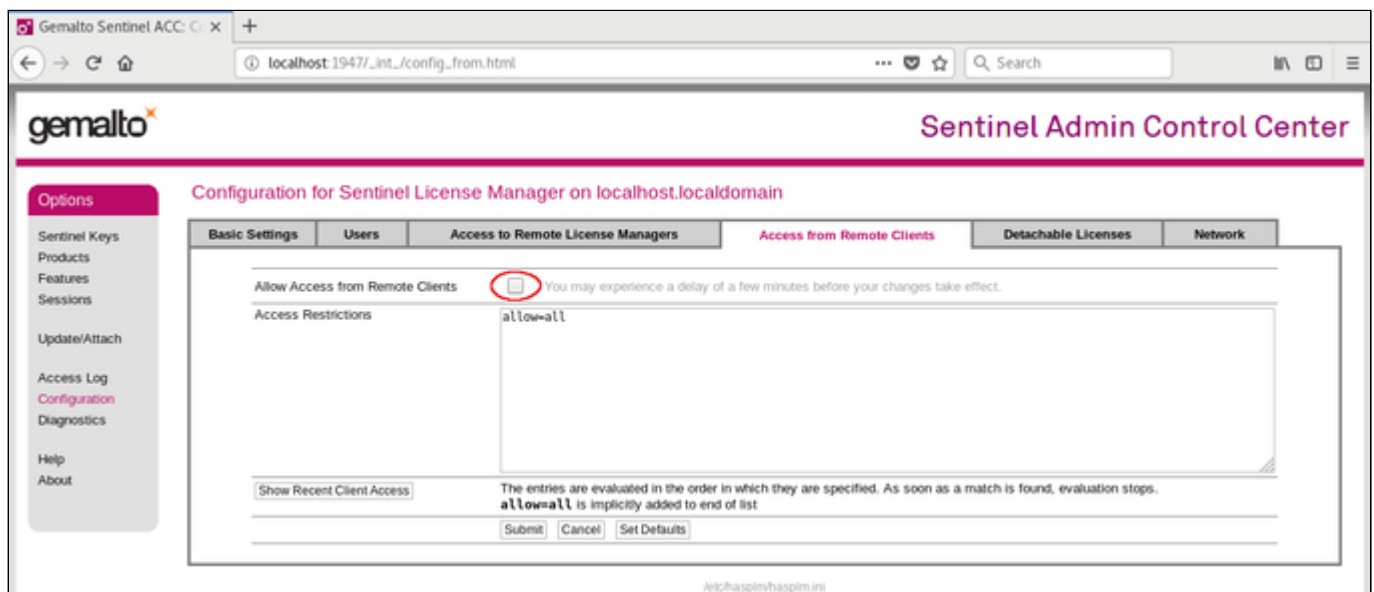
The Sentinel license manager on the license server can be configured to **allow** or **disallow** access from remote clients to the network licenses installed on the license server. To access this configuration option, perform the following steps **on the license server**:

1. Open a web-browser and go to the URL **http://localhost:1947/\_int\_/config\_from.html** (option: Configuration / Access from Remote Clients).
2. This opens a page similar to the following. **Please note**: newer Charon emulator products (e.g., Charon-AXP/VAX version 4.9) have newer versions of Sentinel ACC. The pages of these versions look different, but the functionality remains mostly the same.

Old ACC version:



New ACC version:



Or for versions starting with 8.x:

Allow Access from Remote Clients	<input type="radio"/> No one <input type="radio"/> Identifiable clients only <input type="radio"/> Anyone, but cloud licenses require identity <input checked="" type="radio"/> Anyone, and cloud licenses can be consumed without identity
Public Address for Access With Identity	<input type="text"/>
Store Identity Secrets	<input checked="" type="radio"/> Plain text <input type="radio"/> Encrypted with the storage key provided with Sentinel AdminAPI
Access Restrictions	<div style="border: 1px solid black; height: 100px;"></div>
<input type="button" value="Show Recent Client Access"/>	The entries are evaluated in the order in which they are specified. As soon as a match is found, evaluation stops. <b>allow=all</b> is implicitly added to end of list
<input type="button" value="Submit"/> <input type="button" value="Cancel"/> <input type="button" value="Set Defaults"/>	

### 3. Possible actions:

- To **allow access** from remote clients, *activate* the check-box next to the field **Allow Access from Remote Clients** or **Anyone and cloud licenses can be consumed without identity** and press **Submit** at the bottom of the page. To allow access from remote clients, network visibility on the "Network" tab must be set to **All Network Adapters**.
- To **refuse access** from remote clients, *clear* the check-box next to the field **Allow Access from Remote Clients** or select **No one**, and press **Submit**.
- **Access Restrictions** allow refining access rules, e.g., by specifying IP addresses. Please refer to Sentinel ACC help for details.

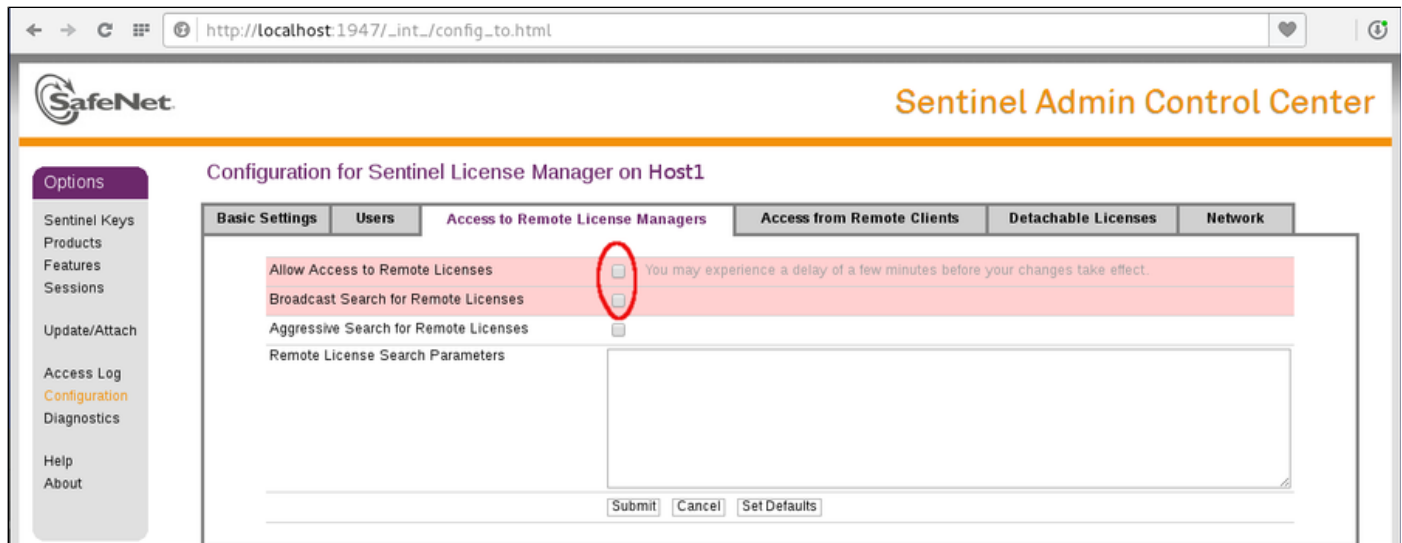
Sentinel ACC versions 8.x and higher have additional configuration options on this screen (mostly cloud related). These options are not relevant to Charon emulator products.

## Allowing Access to a License Server on the Client

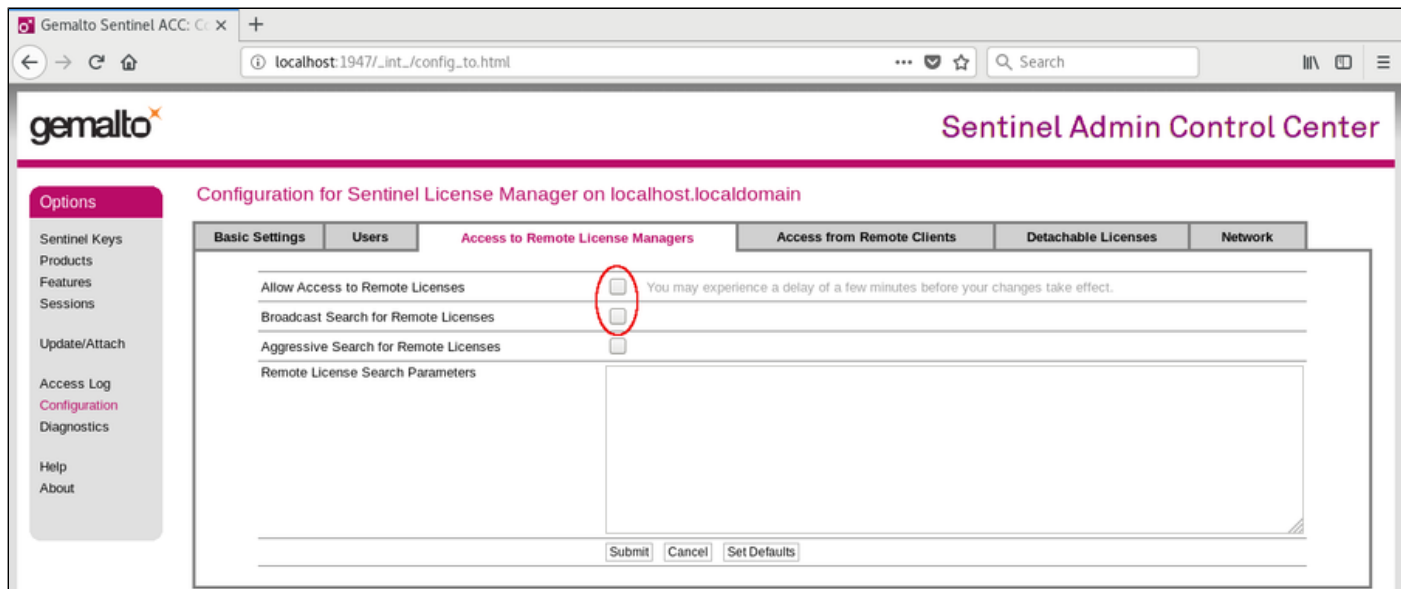
The Sentinel ACC can be configured to enable or prevent that the client host discovers network licenses and to change the options used to discover and access network licenses provided by a license server.

1. Open a web-browser on the **client host** and go to the URL **http://localhost:1947/\_int\_/config\_to.html** (option: Configuration / Access to Remote License Managers).
2. This will open a configuration page similar to the following. **Please note:** newer Charon emulator products (e.g., Charon-AXP/VAX version 4.9) have newer versions of the Sentinel license drivers. The Sentinel ACC pages of these versions look different, but the functionality remains mostly the same

Old ACC version:



New ACC version:



3. Possible actions:

- **Activate** the check-box next to the field **Allow Access to Remote Licenses** to **enable** access to license servers. Press **Submit** to save the setting.
- **Clear** the check-box next to the field **Allow Access to Remote Licenses** to **disable** access to license servers. Press **Submit** to save the setting.
- The option **Broadcast Search for Remote Licenses**, when activated, enables a broadcast search for license servers on the local network without having to enter the address of a license server.

**Please note:**

- If the option **Broadcast Search for Remote Licenses** is not enabled or cannot be used in the customer specific setting, you can enter specific IP addresses or host names that should be queried for network licenses in the **Remote License Search Parameters** field. Please refer to the Sentinel ACC help function for more information.
- To allow access to remote license managers, network visibility on the "Network" tab must be set to **All Network Adapters**.
- Starting with Charon-AXP/VAX 4.9 for Linux and Charon-AXP/VAX version 4.8 for Windows, these Charon emulator products do not follow the settings in the Sentinel ACC with respect to querying remote license servers and network visibility. They perform a **broadcast search** for network licenses even if this has been disabled in the local Sentinel ACC configuration. If this behavior has to be prevented for specific reasons, the access of the system to the license server has to be temporarily restricted or disabled, for example by blocking the relevant traffic in a firewall. Another possibility would be to block access to the network license at the license server side. Note that such methods can negatively impact other functions of the system or, in the case of blocking access to a network license on the server, even the functions on other license clients.

## Verifying License Availability

After installing a license on the system, verify the availability of the license as described in the section *View Existing Licenses*.

Check if the license shows the correct product, expiration date etc.

## Handling Multiple Keys

A Charon host system can have access to several local and remote license keys. Each license key can contain one or more product licenses.

The Charon emulator products can only use one **active** license at one time. Without additional configuration, they cannot make decisions about which product license or license key to use. If there is more than one available license key, the default license key will be used. The default license key is determined by the Sentinel software. As this can lead to undesirable effects, newer Charon emulator products have configuration parameters that allow the definition of a **primary** (or production) and a **secondary** (or backup) key, or the creation of a **list of license keys or product licenses sorted by priority**.

The following sections describe the options available to achieve a more deterministic license selection. If only one license is available, either locally installed or via the network, this section does not apply.

**Please note:** for Charon-AXP/VAX before version 4.9 and other older Charon products: the parameters described in this section can **only** be used to define a primary key and a backup key, or to specify the correct key if there is a conflicting license for a different Charon product on the same system. It is **not possible** with these versions of Charon to combine a local license (black dongle) with other licenses (local or network) to increase the number of concurrent instances of the same Charon product on one host system. Newer versions allow a more flexible use of multiple licenses. Both variants are described in the following sections.

## Backup License Characteristics

It is strongly recommended to order a backup key to recover immediately from damage or loss of the main license key. A backup license key can also help in situations where the Charon host hardware fails and the software must be moved to a different system, thus invalidating the original software license.

Backup keys typically use a counter (integer) value programmed in the key. This integer value corresponds to a number of hours the Charon software is allowed to run. Each time the Charon software checks the license (at start and then every hour), the value is decreased (by 1 hour). Please note that backup keys have may have additional restrictions:

- The runtime is typically limited to 720 hours (30 days). This is the time available to get a replacement license from Stromasys.
- A backup license key may be valid only until a certain date.

**Please note:** if you start and stop the emulator frequently (e.g., frequent runs with a duration of under one hour), the runtime may be significantly less than 30 days, because the license check during the start of an emulator will reduce the counter by one.

## Primary and Backup License for Charon-AXP/VAX prior to Version 4.9

If more than one key is visible to the system, you can define which is the primary and which the backup key. To do this, add the following parameters to the configuration file of the Charon instance:

```
set session license_key_id[0]=<primary-key-id>
set session license_key_id[1]=<secondary-key-id>
```

The entry with index 0 denotes the primary key, the entry with index 1 the backup key.

To identify the relevant key IDs, display the available licenses as described in section *View Existing Licenses*.

To get information about modifying the Charon configuration files, refer to the appropriate sections in the user's guide.



# Prioritized List of License Keys or License IDs for Charon-AXP/VAX 4.9 and Higher

## Prioritized List of License Keys

If more than one key is visible to the system, you can define a prioritized list of license keys. To do this, add the following parameter to the configuration file of the Charon instance:

```
set session license_key_id = "<key-id prio1>, <key-id prio2>,...[,any]"
```

The **first entry** in the list has the highest priority. The emulator instance will try to use this key first. **If it is not available**, it will try the second key in the list and so on. If the parameter **any** is added to the end of the list, the emulator instance will search for any keys not part of the list should none of the listed keys be available. If the parameter is not defined, an unqualified search for a suitable license will be performed and the first one found (if any) will be used. If no usable license is found, the emulator will stop.

To identify the relevant key IDs, display the available licenses as described in section *View Existing Licenses*.

To get information about modifying the Charon configuration files, refer to the appropriate sections in the user's guide.

## Prioritized List of Product License IDs

Every license key can contain one or more product sections. Newer licenses can contain a product license ID identifying each product section. A prioritized list of product license IDs can be added to the emulator instance configuration file to specify which product sections to use and which should have the highest priority. To do this, add the following parameter to the configuration file of the Charon instance:

```
set session license_id = "<lic-id prio1>, <lic-id prio2>,..."
```

The emulator instance will scan the available license keys for the listed product license IDs. Then it will try to use the first defined product section. If it is not available, it will try the next and so on. If none of the listed product license IDs are found, the emulator will stop.

To identify the relevant product license IDs, display the available licenses as described in section *View Existing Licenses*.

To get information about modifying the Charon configuration files, refer to the appropriate sections in the user's guide.

## Update an Existing License

If you need to update an **existing hardware or software** license, for example because the time limit on the license has expired or to upgrade to a new product version, perform the following tasks:

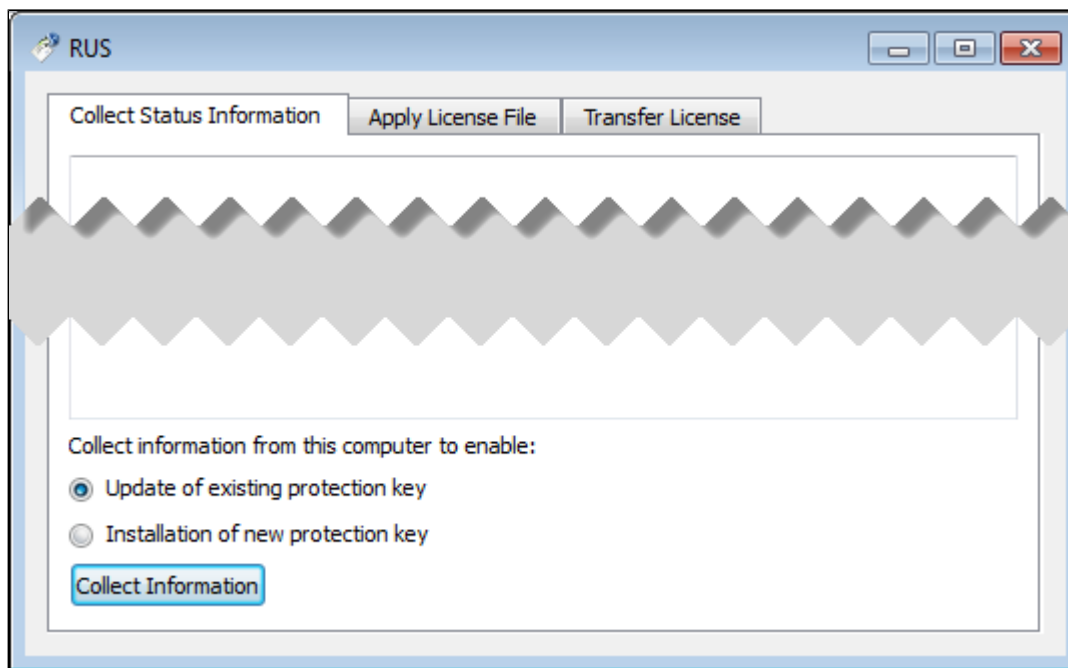
1. Generate the C2V file for the existing license. This step is the same for a hardware license or an existing software license. This Customer-to-Vendor (C2V) file contains the license characteristics necessary for creating the license update.
2. Send the C2V file to Stromasys. Stromasys will use the data to create the necessary license update. You will receive one (software license) or two (most hardware license updates) V2C files (the Vendor-to-Customer file).
3. Apply the license data from the V2C file(s). This will install and activate the update for your license.

These tasks are described below:

**Step 1:** Start the license update utility as described above and generate the C2V file:

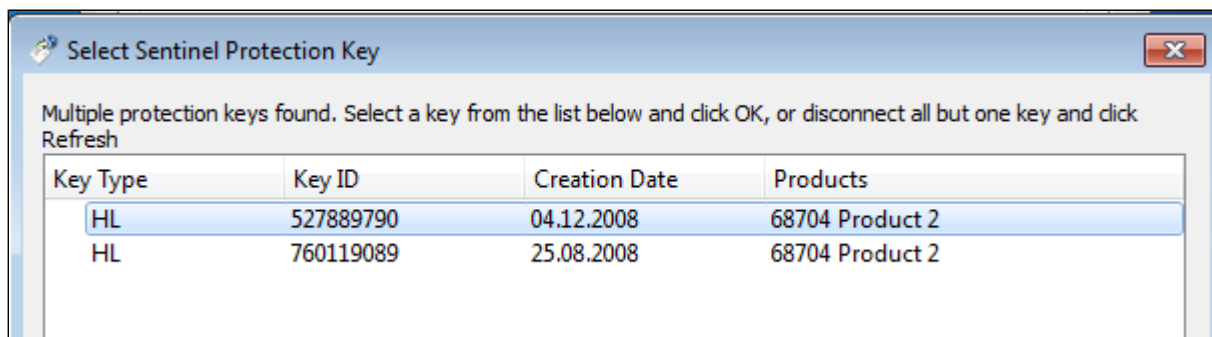
The License Update Utility (HASPRUS) looks the same, independent of the product variant used.

The following image shows the **necessary options** for generating a C2V file for an existing license: select the tab **Collect Status Information** and the option **Update of existing protection key**.



After pressing the **Collect Information** button, you will be prompted to enter a location for storing the C2V file. Select a location and filename and save the file.

If there is more than one license key available on the system, the tool will allow you to select the appropriate key to update as shown in the screenshot below:



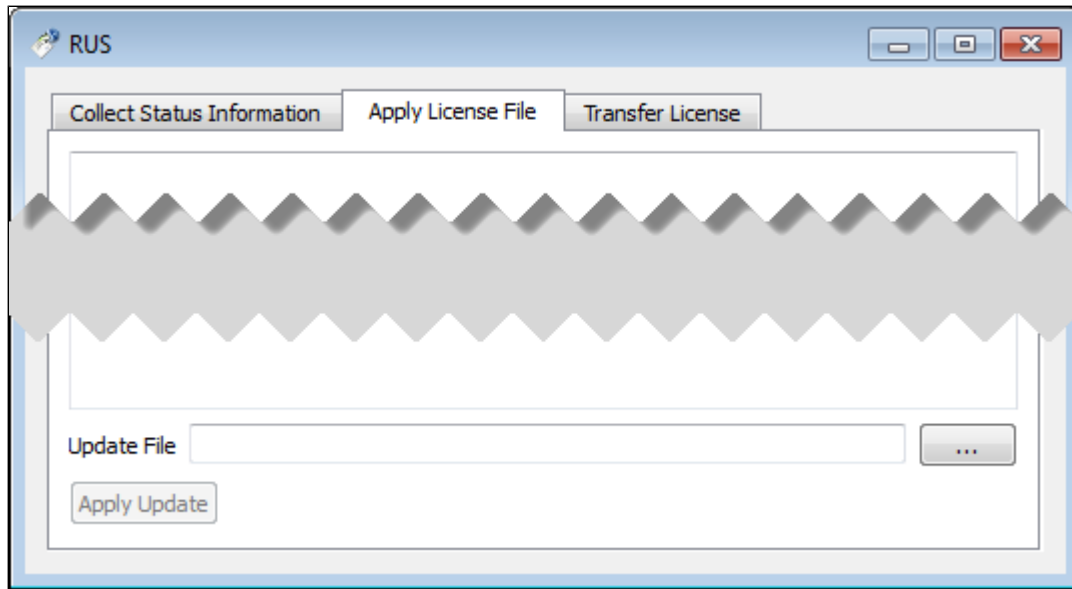
**Step 2:** Send the resulting C2V file to Stromasys orders administration using the email address that Stromasys will provide to you.

**Step 3:** After receiving the V2C file(s) from Stromasys, copy the file(s) to the system where the license needs to be installed, start the license update utility as described above, and install the new license:

**Important caveat:** if a hardware USB key is to be updated, you will in most cases have received **two** files: a \*\_fmt.v2c file and a \*.v2c file. The \*\_fmt.v2c file formats the dongle and the \*.v2c file contains the updated license data. In such cases the \*\_fmt.v2c file must be applied **first**.

**Important caveat:** if a hardware USB key is to be updated, you will in most cases have received **two** files: a \*\_fmt.v2c file and a \*.v2c file. The \*\_fmt.v2c file formats the dongle and the \*.v2c file contains the updated license data. In such cases the \*\_fmt.v2c file must be applied **first**.

1. To install the V2C file(s) for hardware **and** software licenses, select the tab **Apply License File**.



2. Press the "..." button next to the **Update File** field. This will open a file browser.

3. Select the V2C file to be applied.

4. Click on **Apply Update** to complete the operation.

If you received two V2C files, perform these steps for both files (remember to **first** apply the \*\_fmt.v2c file).

Alternatively, you can use the Sentinel ACC to apply V2C files (use the section Update/Attach). Refer to the license handbook for more details.

## Removing a License

If a license needs to be **removed** from a system or the access to the license needs to be **disabled**, the steps are different for the different license types and are described in the next sections:

- Removing a hardware license from a system
- Removing a software license from a system
- Removing access to a network license from a system

**Important:** before removing a license from a system or disabling access to a license, take the following points into consideration:

- Make sure that there is a valid replacement license for all emulator instances depending on the license to be removed, or make sure the emulated systems have been cleanly shut down,
- Make sure that this step will not negatively impact your production environment.

## Removing a Hardware License from a System

To remove a hardware license from a system, simply unplug it from the system.

## Removing a Software License from a System

**Please note:**

The old method for removing a software license from a system depended on the license transfer functionality. **This functionality is no longer available in the current versions of the Charon products.** The only way to remove a non-transferable obsolete software license is the method described below. On Microsoft Windows, you can use the **Charon License Utility** (see [CHARON on Windows - Charon License Utility](#)) to perform the necessary steps.

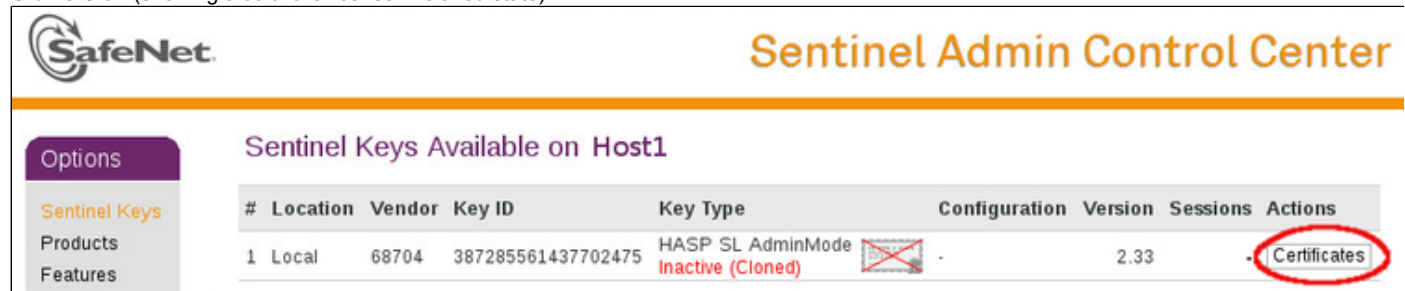
To remove an **obsolete software license or a software license in cloned state** (a software license goes into a cloned state, for example, if a VM is cloned to a different Virtual Machine UUID) from the system, perform the following basic steps:

- Start the Sentinel Admin Control Center (ACC) (for local system: URL <http://localhost:1947>).
- Identify the software license that is to be removed.
- Identify the name of the certificate file and the path of the base directory where it is stored.
- Remove the certificate file(s).
- Restart the Sentinel runtime service.


The following example shows the required steps in detail:

1. Go to <http://localhost:1947/> to access the Sentinel HASP ACC and select **Sentinel Keys** on the left.
2. In the Sentinel Admin Control Center (ACC), go to **Sentinel Keys** and locate the target "Sentinel SL AdminMode" license to be removed.
3. Click on the **Certificates** button at the right side of the SL description:

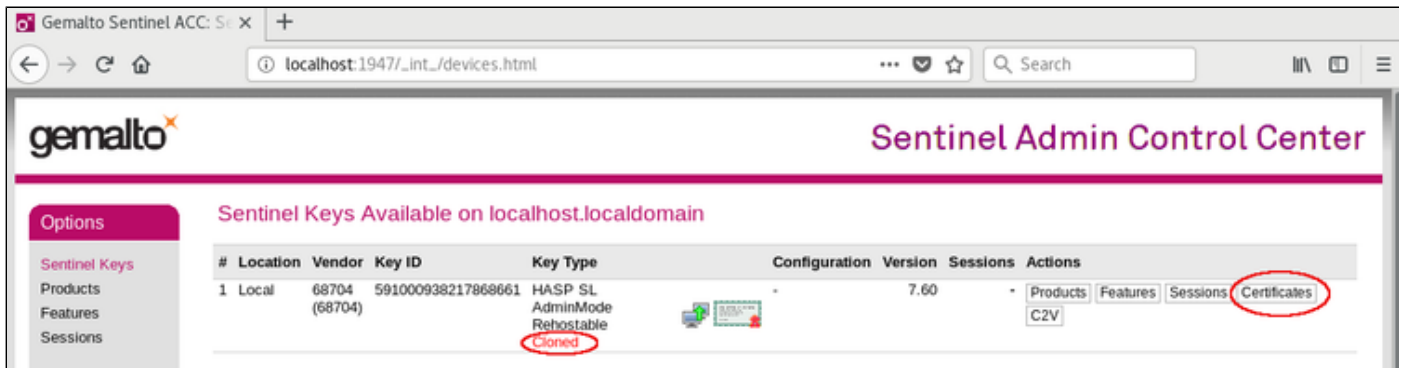
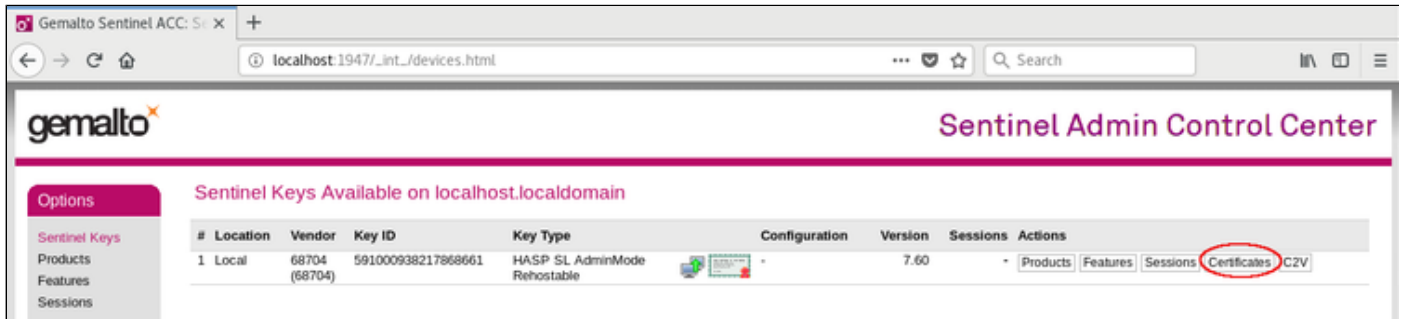
**Old version** (showing a software license in cloned state):



The screenshot shows the Sentinel Admin Control Center interface. On the left, there is a navigation menu with 'Options', 'Sentinel Keys', 'Products', and 'Features'. The main content area is titled 'Sentinel Keys Available on Host1' and contains a table with the following data:

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

**New version** (showing an active software license and a cloned software license):

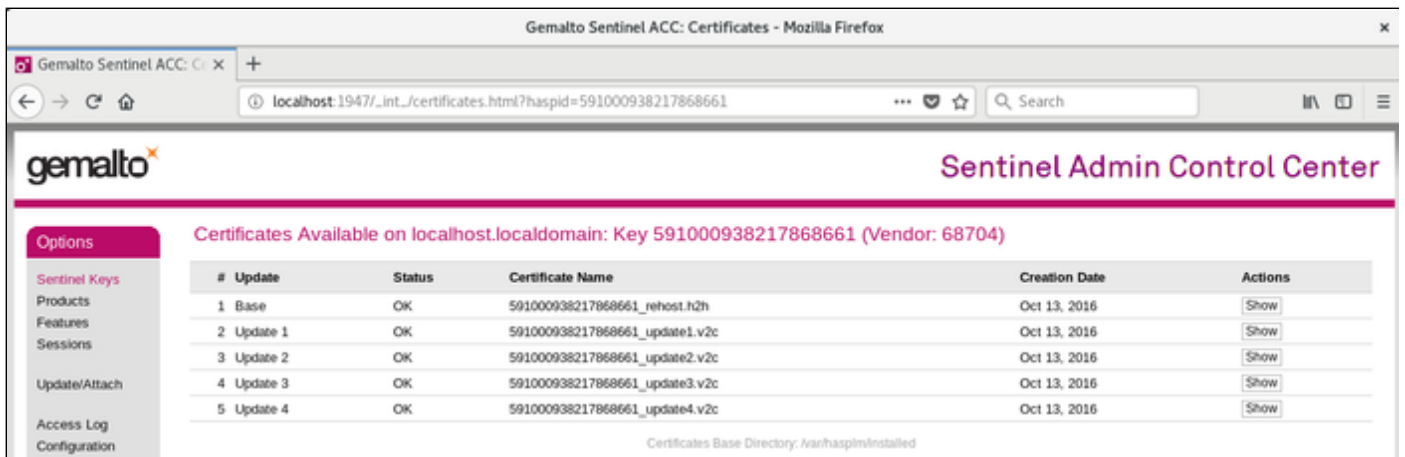


4. Note the name of the corresponding certificate file and the path to the certificates base directory as shown in the "Certificates" section:

Example using old ACC version:



**Please note:** the certificate display screen can look slightly different depending on the type of software license and the actions performed on this license in the past. The following example shows the **certificates of a software license after several updates**. The screenshot is based on Sentinel ACC version equal or higher than version 7.60.



## 5. Remove the certificate file(s) using operating system tools.

a. In the first example above (Windows), there is one file to be removed. It is in `C:\Program Files (x86)\Common Files\SafeNet Sentinel\Sentinel LDK\installed\68704`.

b. In the second example (Linux) there are several files created by a series of license updates. On Linux, the certificate files are typically stored in `/var/hasplm/installed/68704/`.

c. **All certificate files in the identified directory must be removed.**

## 6. After removing the certificate file, restart the Sentinel license service or reboot the system. To restart the service, proceed as follows:

a. On Windows, restart the "**Sentinel LDK License Manager**" service (hasplms) either using the `services.msc` program or via a command line (as administrator) then enter the `sc stop hasplms` command followed by `sc start hasplms`.

**Important:** any running emulators will be stopped if the license service is forced to stop despite existing service dependencies (i.e., running emulators). So any running guest systems **must be cleanly shutdown** before restarting the license service in order to avoid data loss.

b. On Red Hat Enterprise Linux 7.x and 8.x (and compatible Linux systems) restart the aksusbd service: # `systemctl restart aksusbd`  
This also will automatically restart the `hasplmd` service (separate service for this daemon was introduced in newer runtime driver versions).

c. On Red Hat Enterprise Linux 6.x, restart the aksusbd service: # `service aksusbd restart`

## 7. Start the Sentinel ACC again to verify that the software license has been removed.

# Removing Access to Network Licenses from a System

There are three different ways to remove a network license. They have different implications and are described below:

## Full Removal of Network License

To completely remove a network license, follow the steps above for hardware or software licenses respectively. The steps must be performed on the license server.

**Please note:** this will remove the license for all emulator hosts on the network using this license and also for the local host. Make sure that this is in fact intended.

## Block Access of Remote Emulator Hosts to Network License

You can use the Sentinel Admin Control Center on the license server

- to **allow** access to a network license for all or specific hosts on the network, or
- to **block** access from remote hosts completely.

Please refer to the section *Controlling Access to Network License on Server Side* in *Installing a Network License* for more information.

## Block Access to Network Licenses on Client System

If access of the local system to network licenses is not desired, this can be disabled in the Sentinel ACC for certain products and product versions. Blocking access at the server side is the safer way.

Please refer to the section *Controlling Access to Network License on Client Side* in *Installing a Network License* for more information.

# Manual Sentinel HASP License Driver Installation

This is normally not required as the license drivers are part of the Charon emulator product installation kits. Also, Stromasys provides a license update utility that simplifies this task.

## New License Driver Installation

Before installing the 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 the most recent version, must be installed. The Charon installation performs this operation automatically.

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

Installation of a Charon regular or network license consists of:

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

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

1. Login as Administrator
2. Open the Charon kit folder
3. Switch to the `hasp_install` subfolder
4. Unzip the archive `haspdinst.zip` located in this folder
5. Open **cmd.exe** and switch to the folder where the files were unzipped
6. Execute the following commands:

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

7. Extract the contents of this archive `hasplib.zip` to the same directory.
8. Copy the file `haspvlib_68704.dll`
  - to `C:\Program Files (x86)\Common Files\Aladdin Shared\HASP` (in case of a x64 host) or
  - to `C:\Program Files\Common Files\Aladdin Shared\HASP` (in case of a x86 host)

Note that the following operations have to be performed when installing a network license:

- *On the server side (where the network license will reside):* open port 1947 for both TCP and UDP
- *On the client side,* if broadcast search for remote licenses is to be used, UDP traffic from port 1947 of the license server to ports 30000-65535 of the client must be permitted.
- *Both on server and client sides:* setup default gateway

Please consult with your 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 as part of the Charon emulator kit).

## Replacing the Currently installed License Drivers

Replacement of the currently installed Sentinel Run-time can be needed in case of installation of specific run-time provided by Stromasys. To proceed, remove the current version of Sentinel run-time:

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

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

Install the new Sentinel run-time:

- **cd** to the directory where the target run-time resides
- Issue the following command:  

```
...> haspdinst.exe -install -cm
```
- **cd** to the **hasp\_install** directory
- Extract the contents of this archive **hasplib.zip** to the same directory.
- Copy the file **haspvlib\_68704.dll**  
to C:\Program Files (x86)\Common Files\Aladdin Shared\HASP (in case of a x64 host) or  
to C:\Program Files\Common Files\Aladdin Shared\HASP" (in case of a x86 host).



# CHARON-VAX and CHARON-PDP for Windows utilities

## General description

CHARON-VAX / CHARON-PDP provides the following set of utilities:

Utility	Description
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.
MkDskCmd	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.
License expiration check	Used to warn user about expiration of CHARON license
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.
HOSTPrint	Used to print CHARON output to Windows printers.
VT-30TV	TV-30TV screen emulation.
CHARON Guest Utilities for OpenVMS	Used to manage virtual tapes and CHARON performance.

# Host Device Check

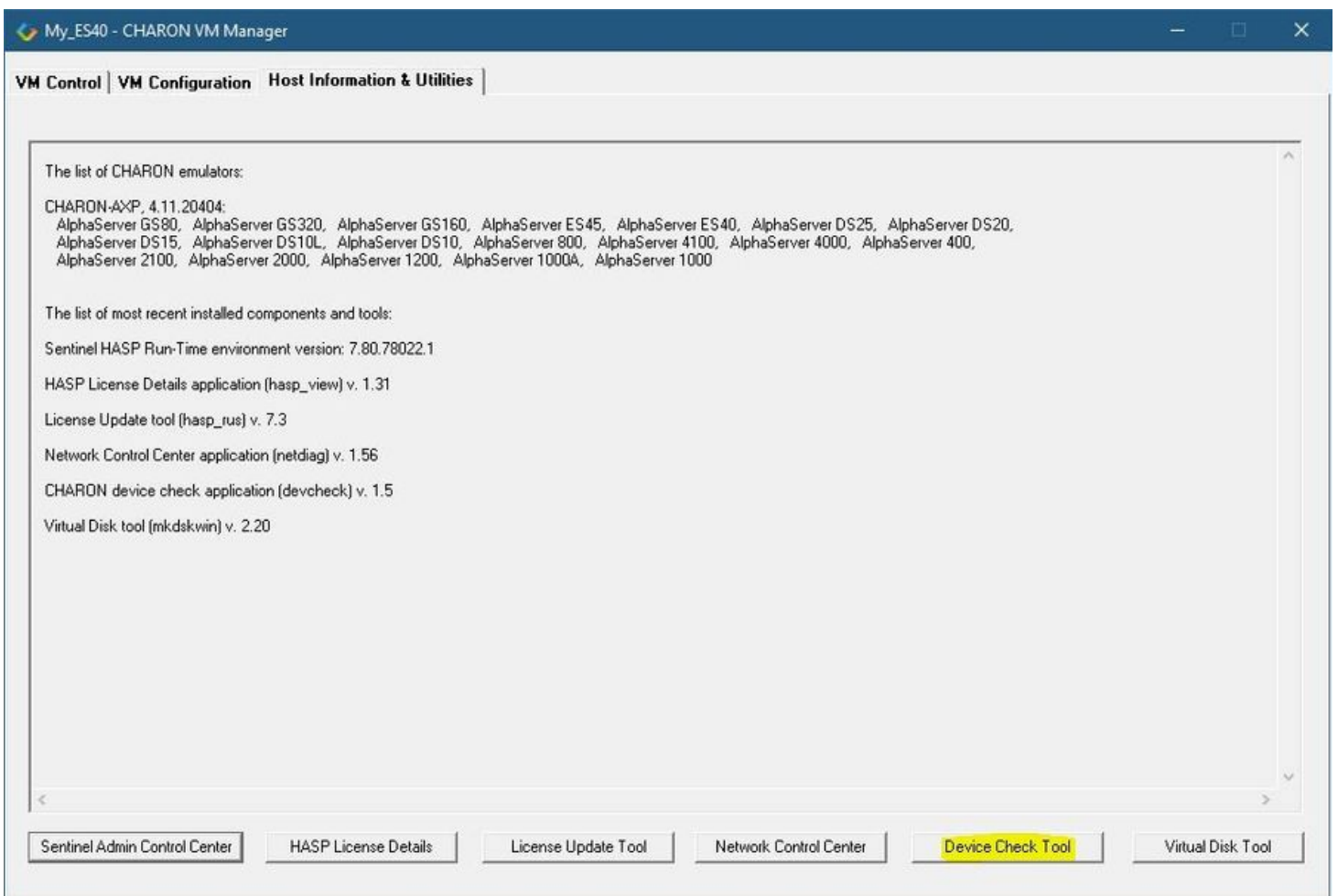
## Table of Contents

- [General Description](#)
- [Reviewing available physical disks](#)
- [Reviewing all the available host resources](#)
- [Collecting the configuration strings](#)

## General Description

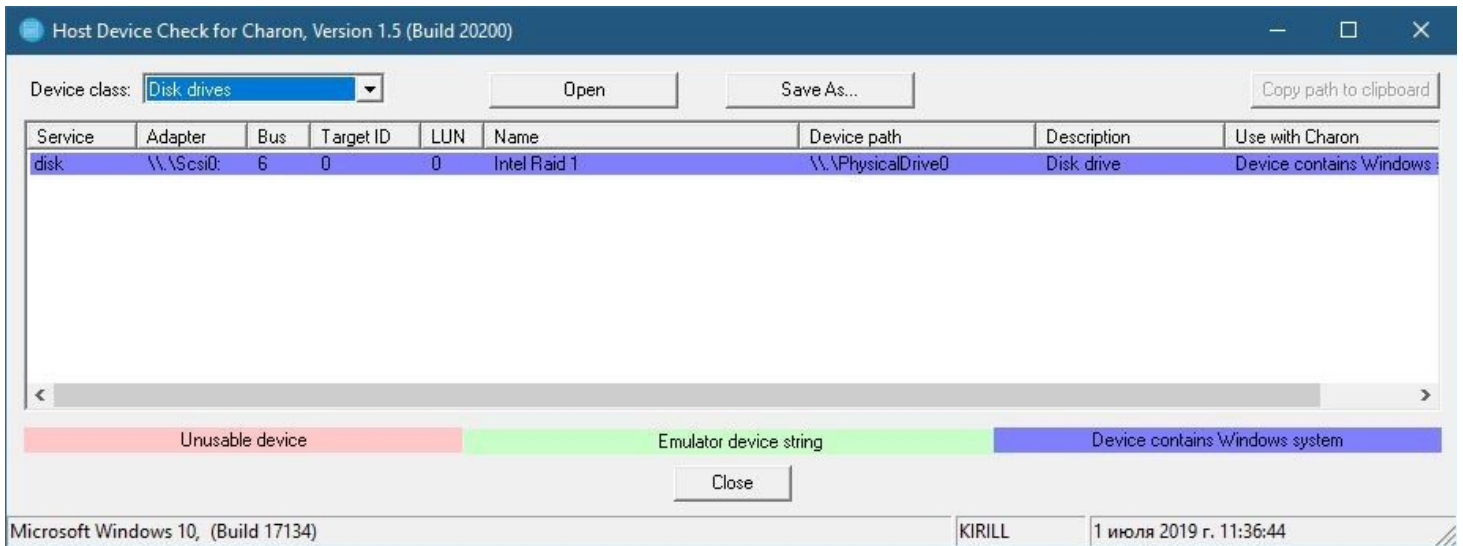
The "Host Device Check" utility is used to review system resources that can be mapped to CHARON.

Open the CHARON Virtual Machine Manager, switch to the "Host Information & Utilities" tab and press the "Device Check Tool" button:



## Reviewing available physical disks

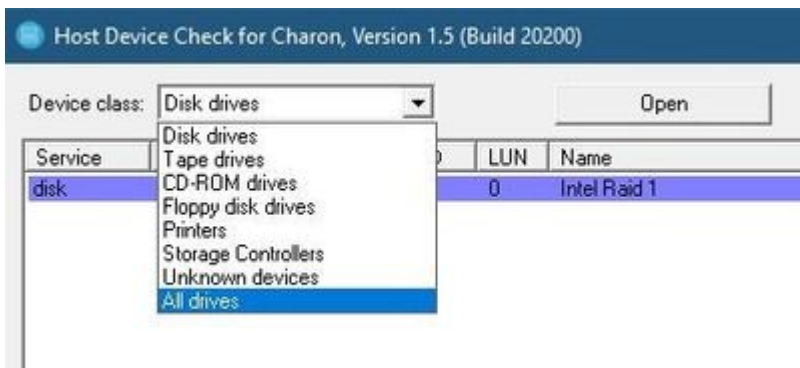
By default the "Host Device Check" utility reports the available physical disks:



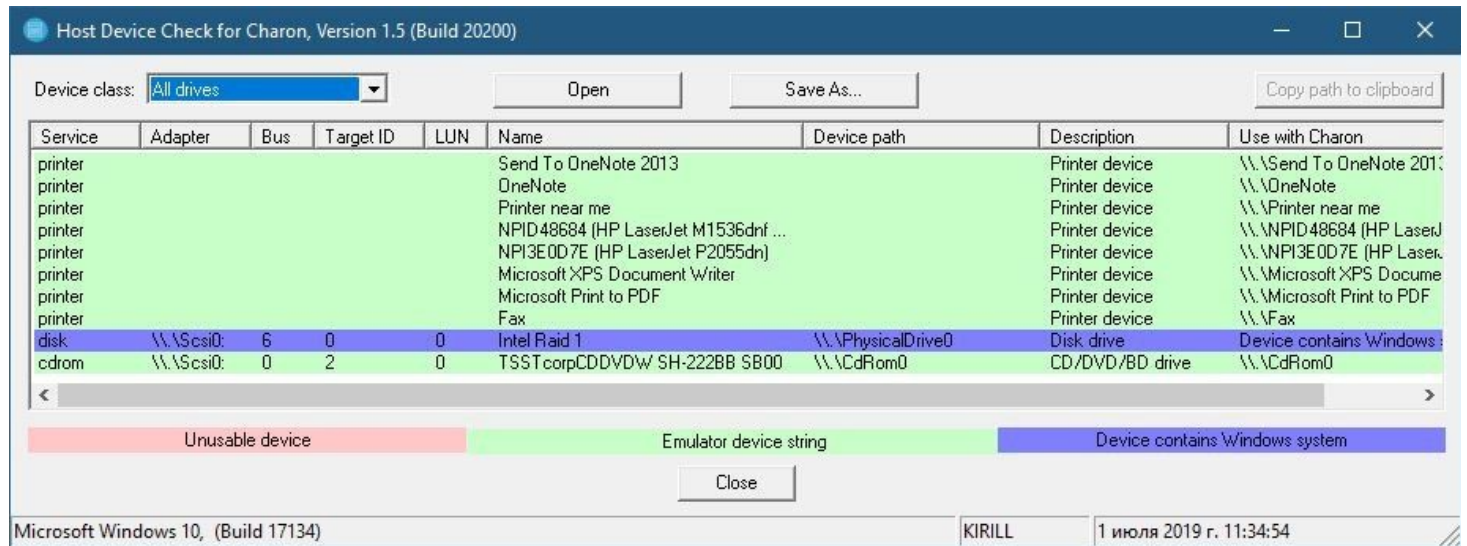
⚠ Please note that the unusable devices and the devices containing Windows system (see corresponding colors above) **must not** be mapped to CHARON!

## Reviewing all the available host resources

Select "All drives" in the "Device class" drop-down list:



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

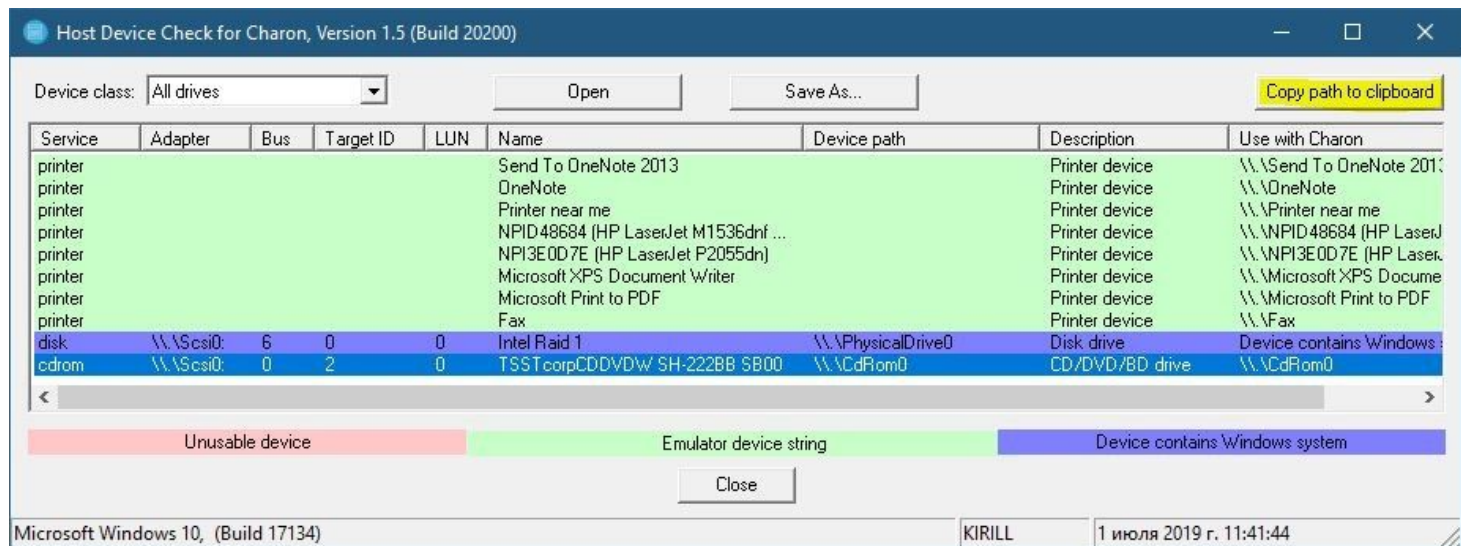


It is also possible to choose all the other categories to narrow the possible mapping options list.

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 (see below).

## 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, it can then be pasted to the CHARON configuration file (using the "CTRL-V" keys combination for example).

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

# Network Control Center

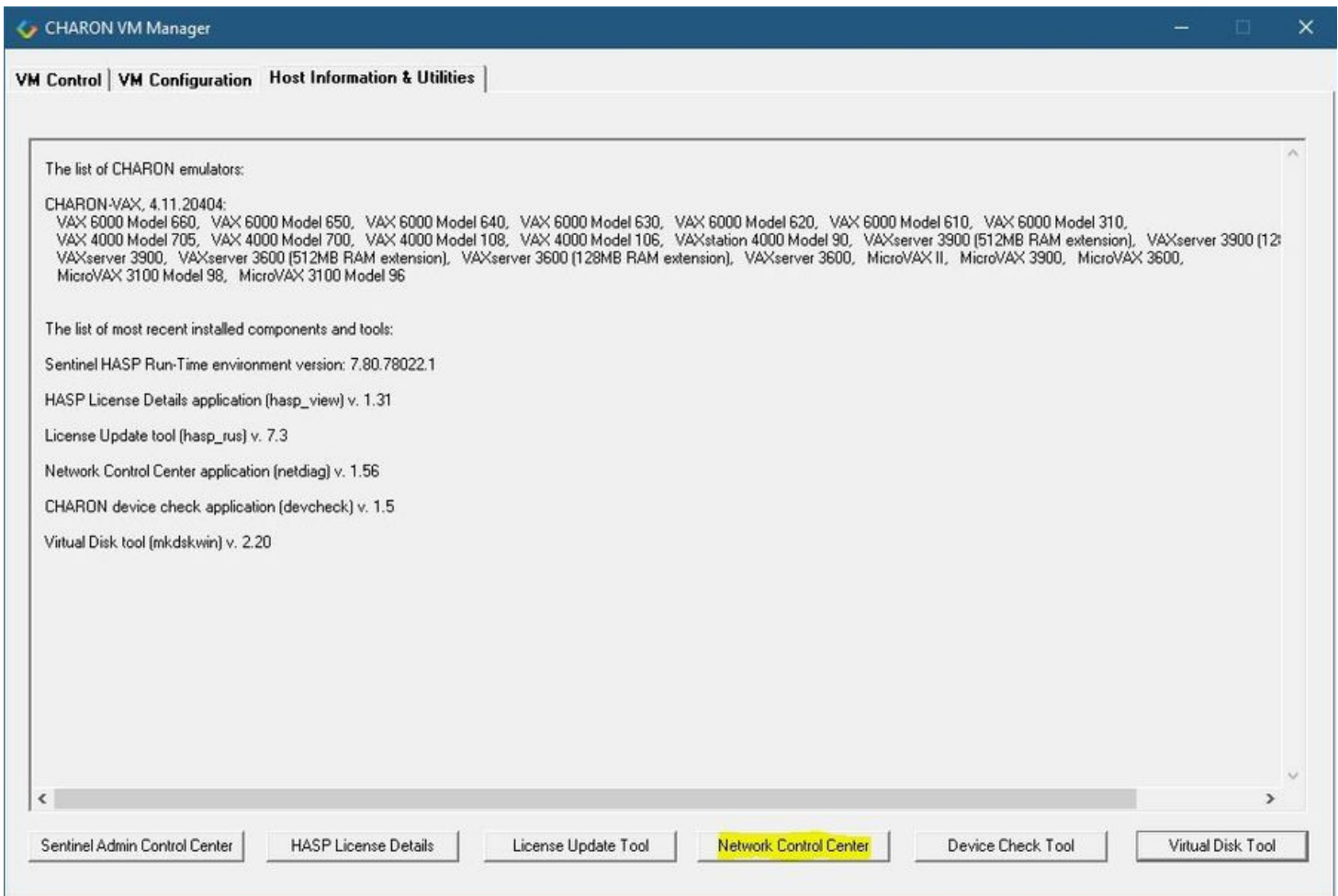
## Table of Contents

- [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](#)
  - [Configuration type 1](#)
  - [Configuration type 2](#)
  - [Configuration type 3](#)
- [Monitoring the CHARON network activity](#)

## General Description

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

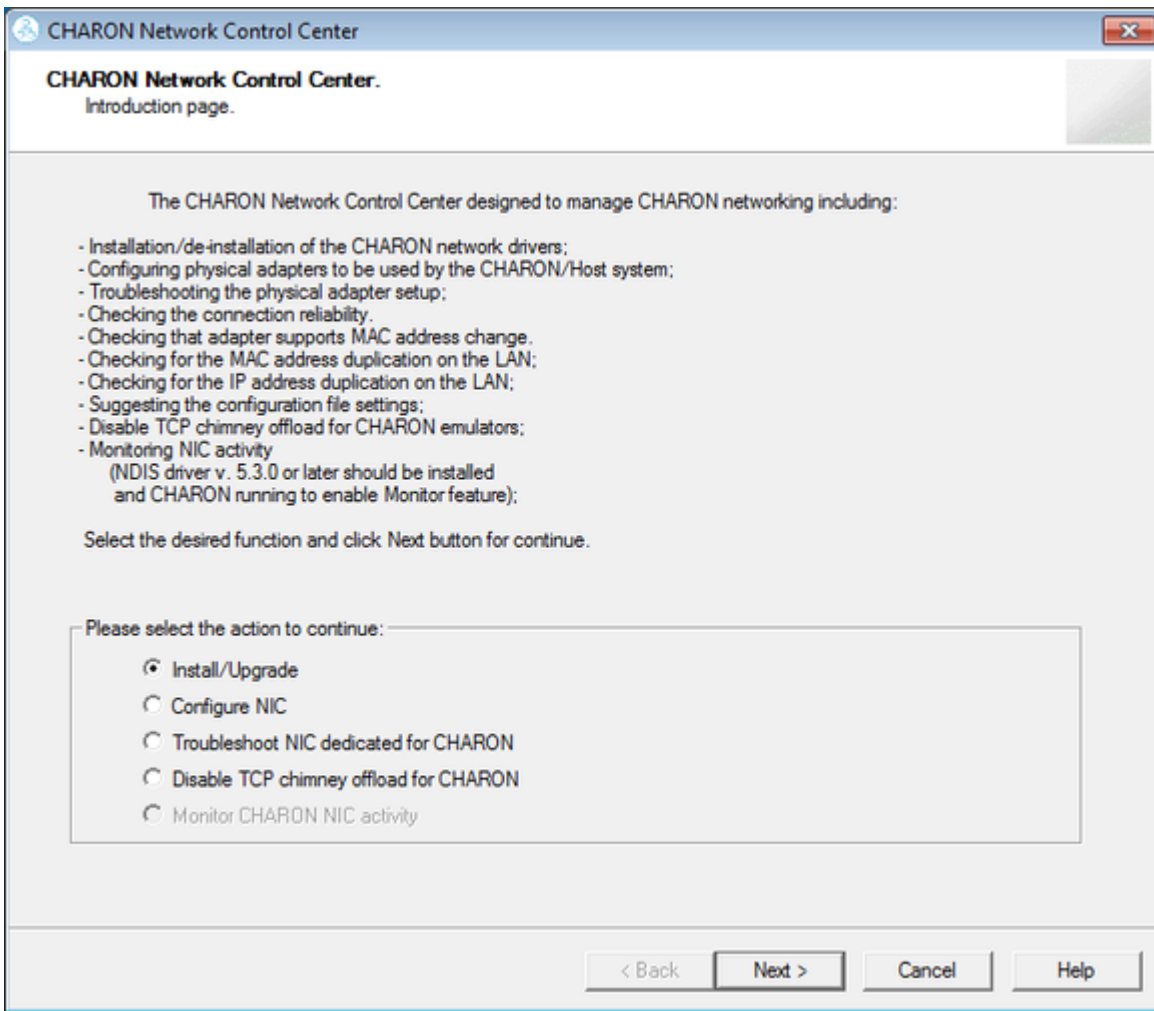
Open the CHARON Virtual Machine Manager, switch to the "Host Information & Utilities" tab and press the "Network Control Center" button:



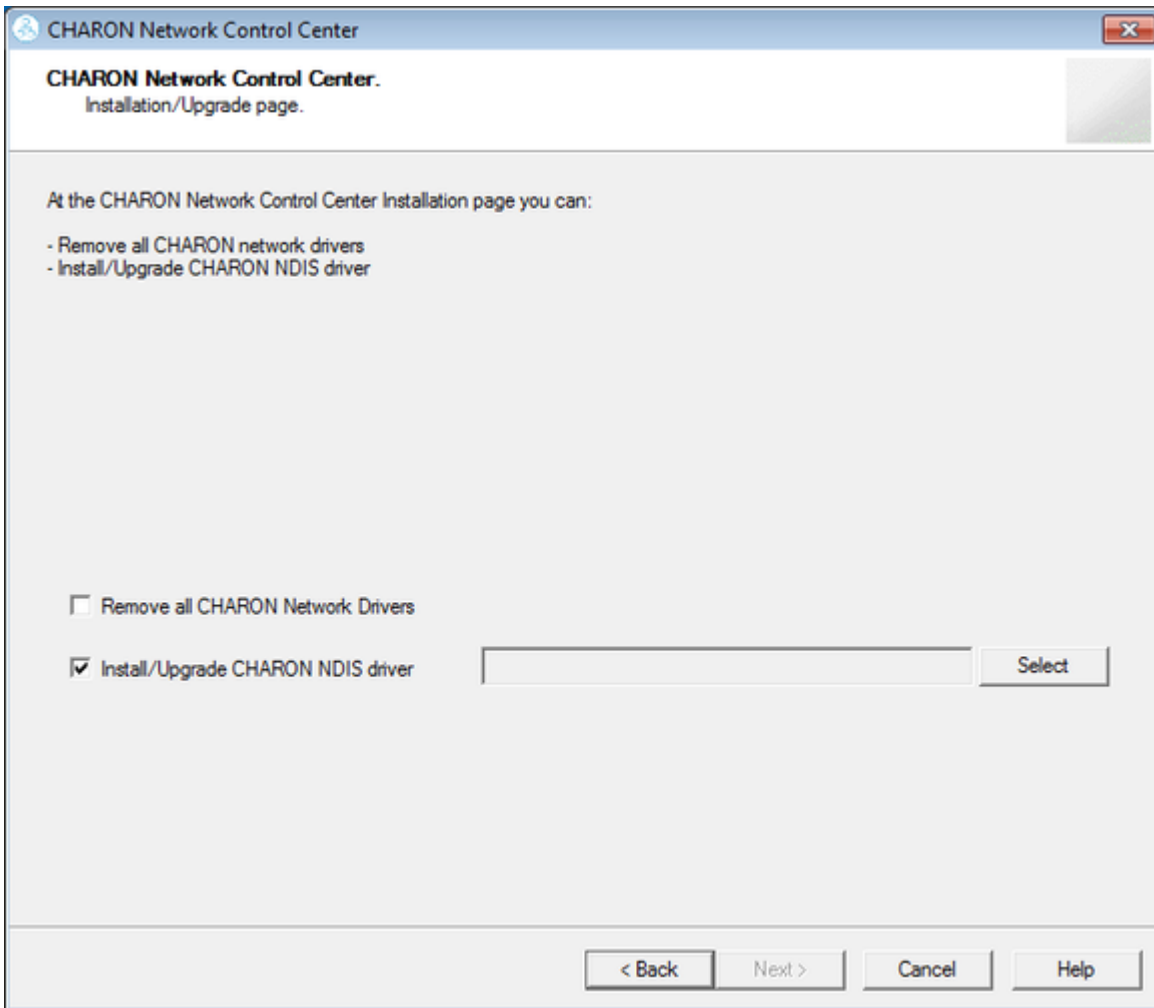
## Installation of CHARON network drivers

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

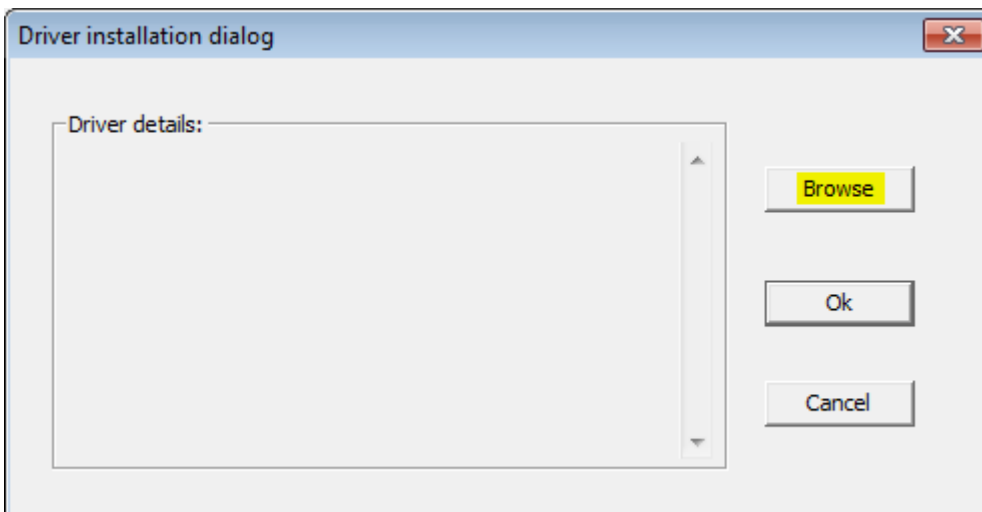
When it is needed to install a modified driver, use the following procedure. Select "Install/Upgrade" and press the "Next" button:



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

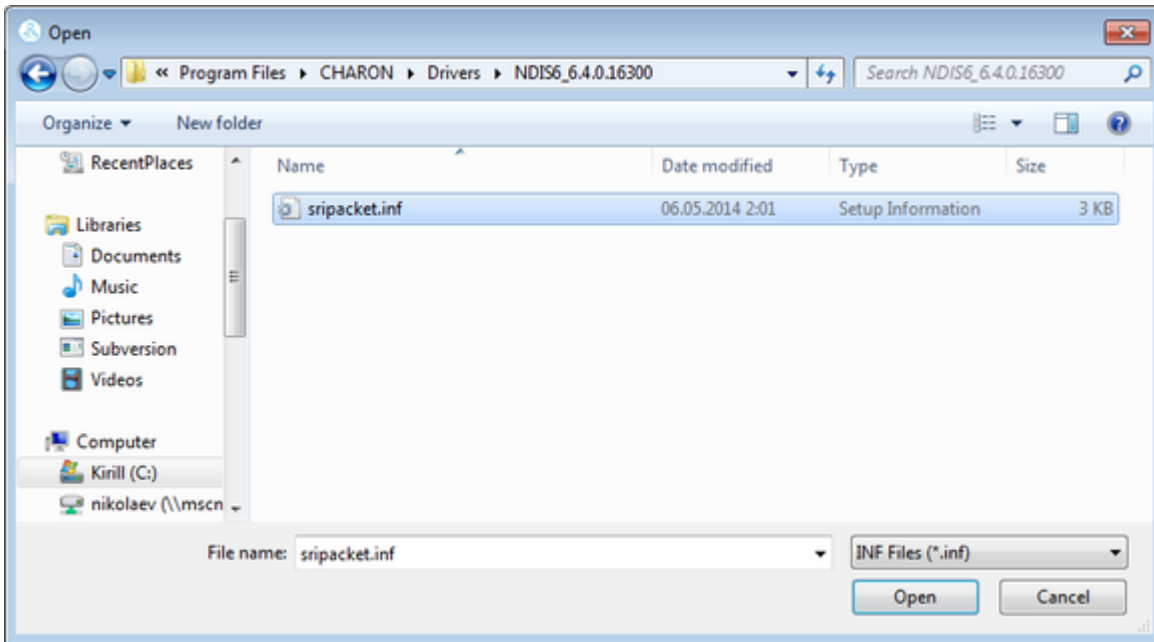


Press the "Browse" button:



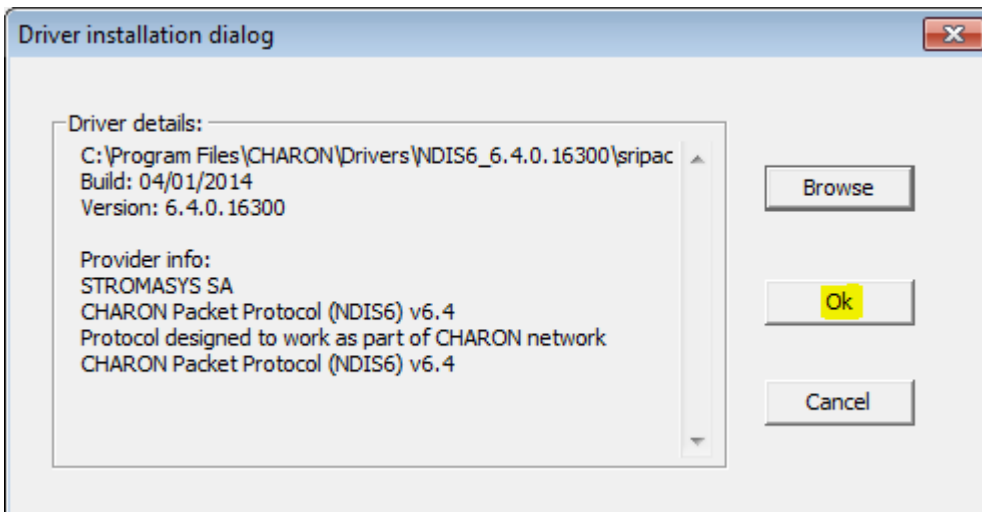


Browse for the target "sripacket.inf" file (by default it is located in the "C:\Program Files\CHARON\Drivers\NDIS6\_X.X.XXXXX" directory), select it and press the "Open" button:



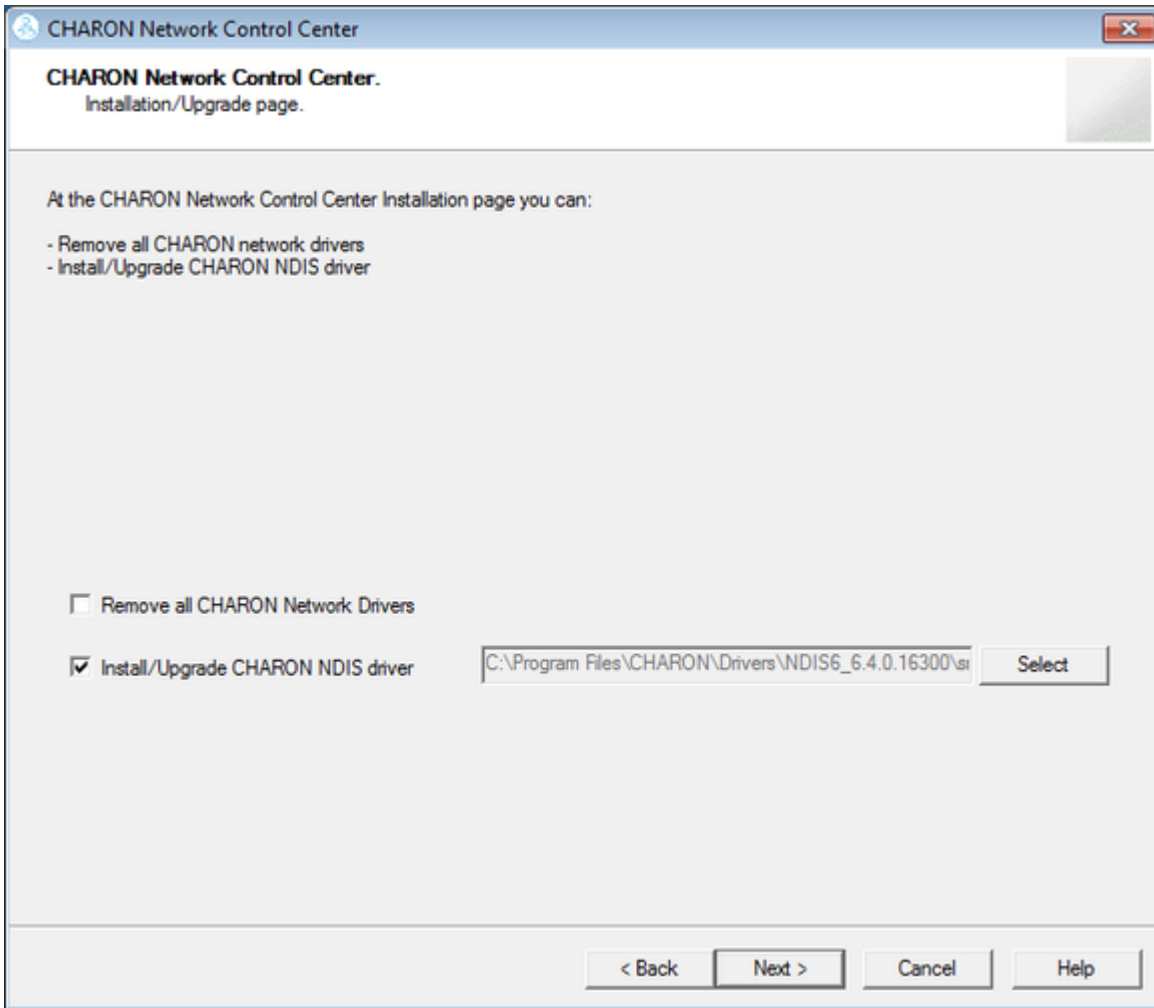
If the CHARON network driver has been acquired directly from STROMASYS, put it in a temporary directory and choose the "sripacket.inf" file from this directory in the dialog above.

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





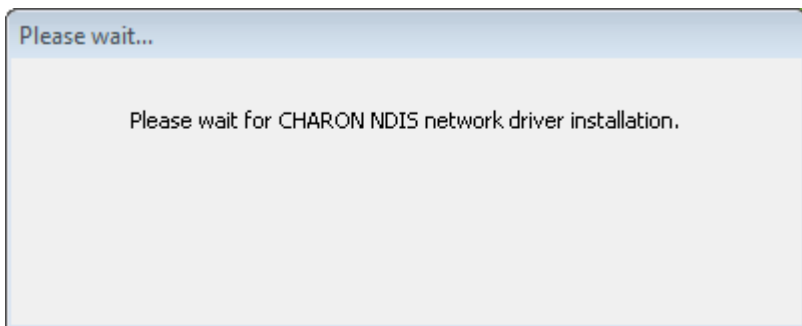
The Network Control Center will display the previous dialog with the path to the CHARON network driver. Press the "Next" button in the popup below to continue:



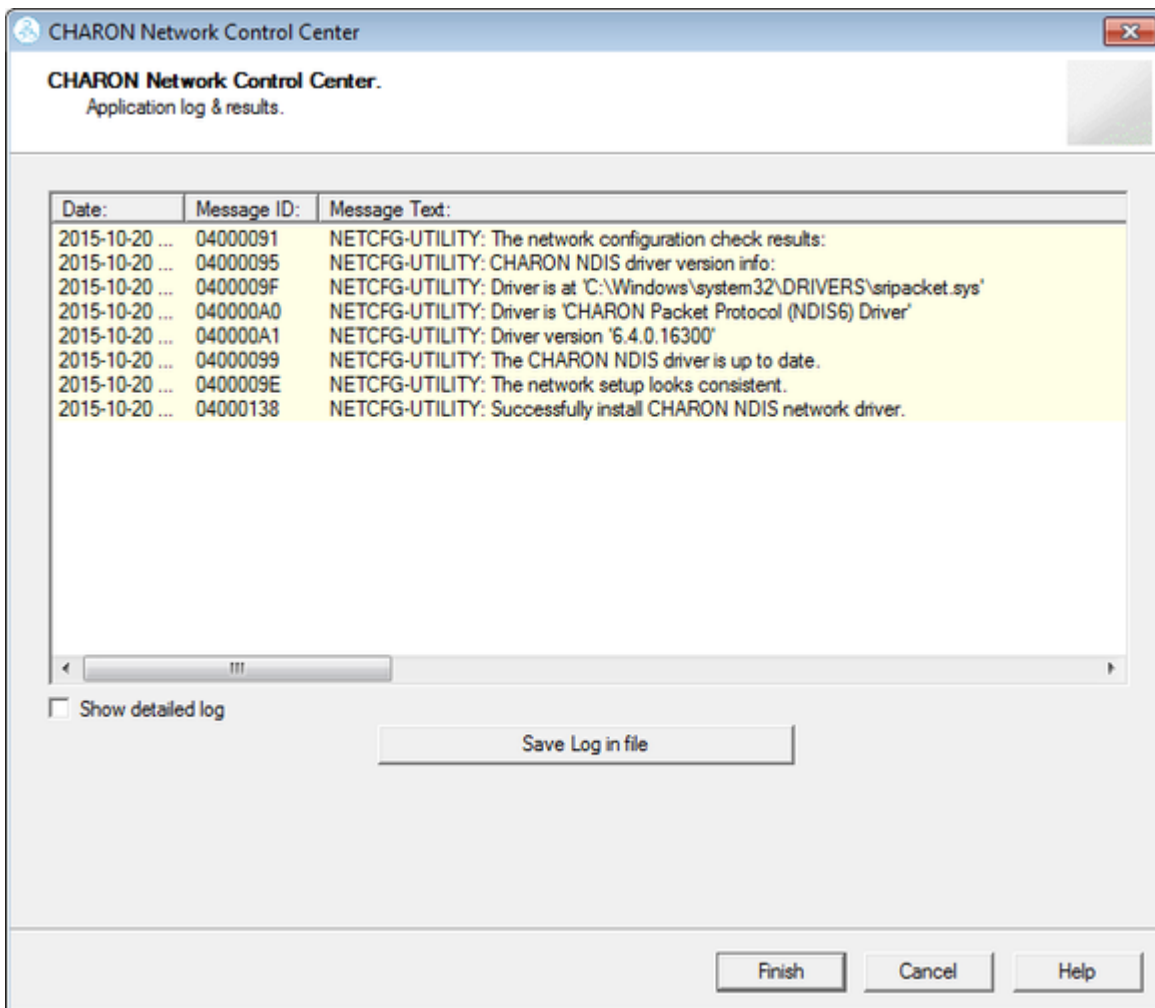
The utility will begin the CHARON network driver installation. If Windows Security asks you to confirm the driver installation, press the "Install" button:



The driver installation will resume:



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



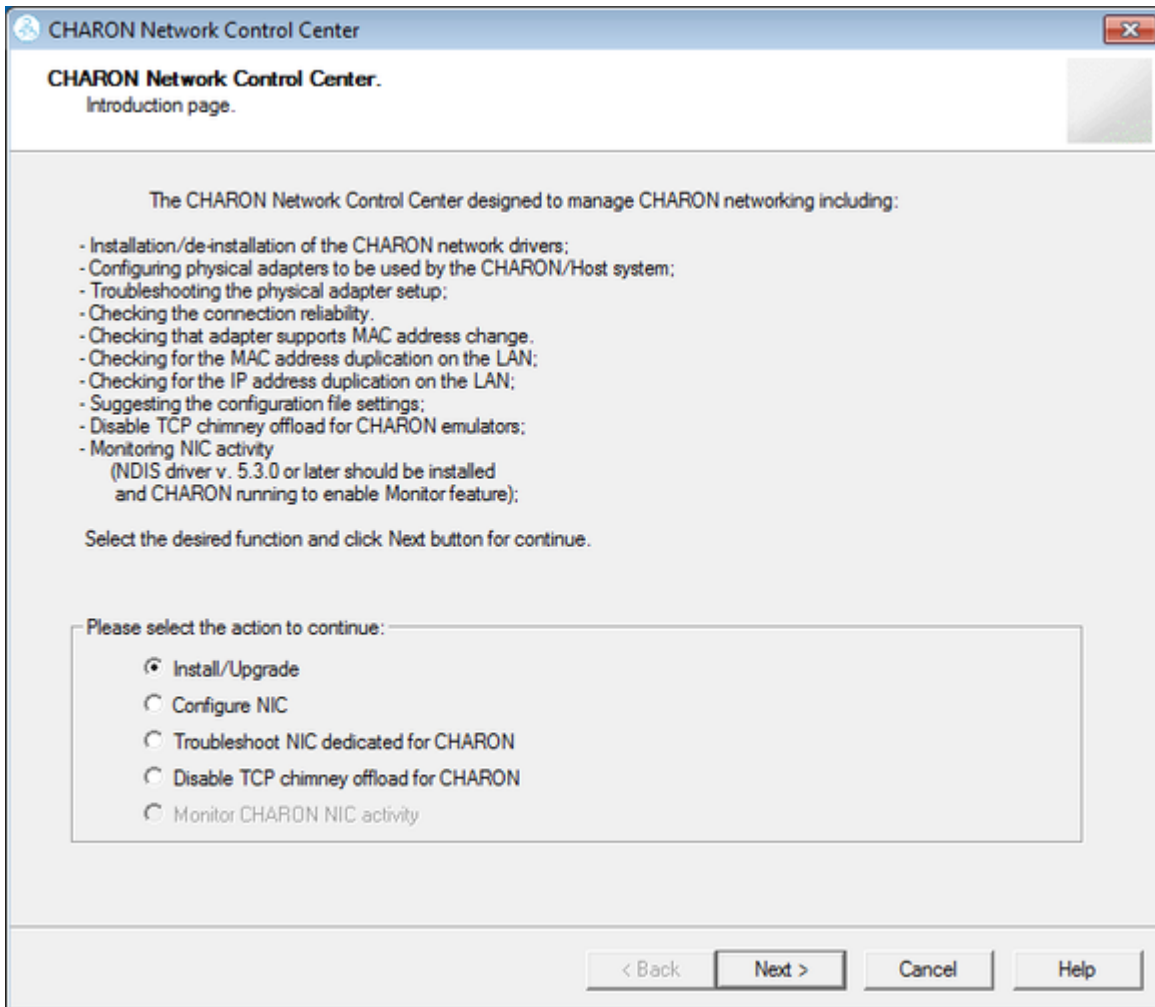
Review the log, make sure it is correct. You can save the log to a file by pressing the "Save Log in file" button in case the log file has to be sent to Stromasys Customer Support.

Select "Show detailed log" to display more detail.

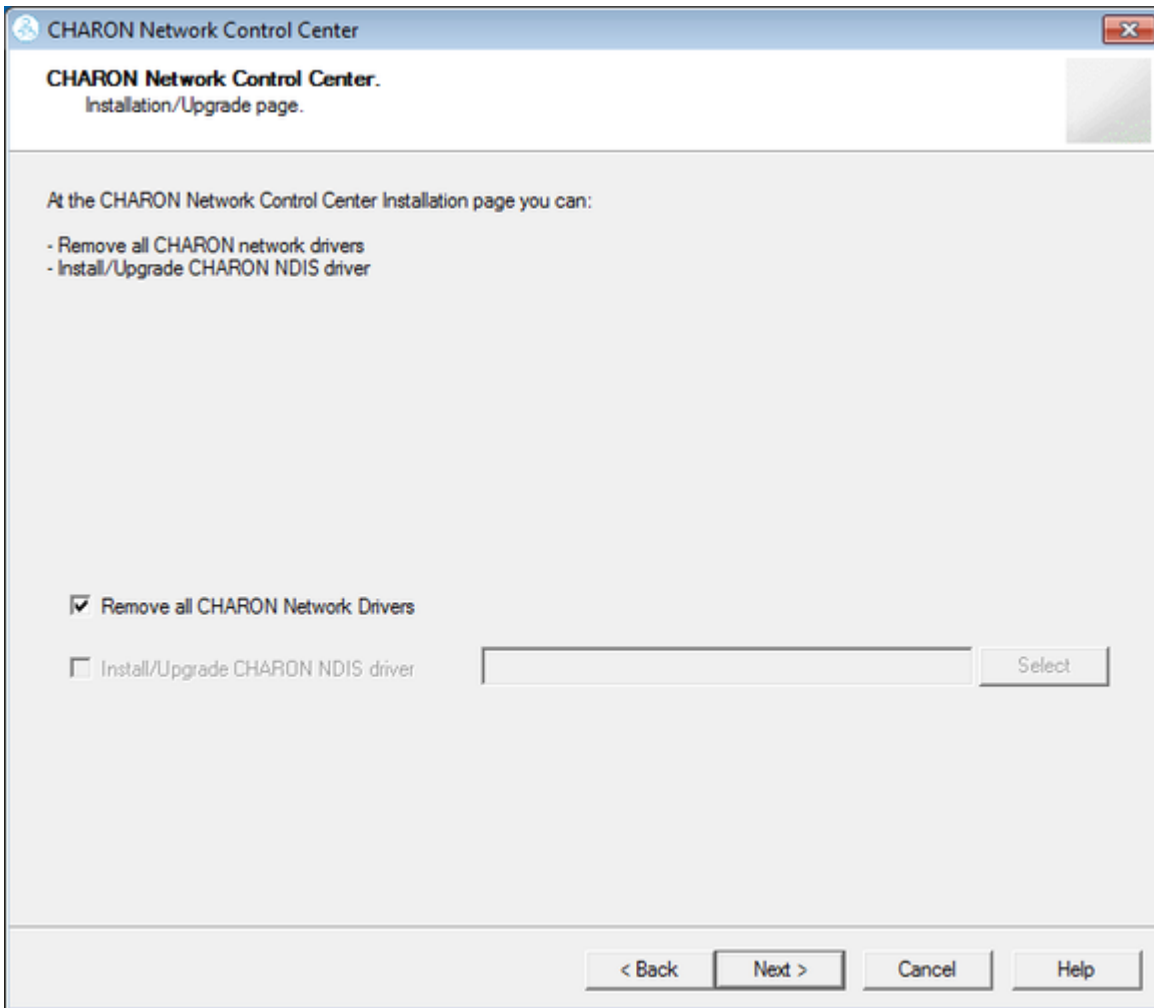
Press the "Finish" button to exit.

## Deinstallation of the CHARON network driver

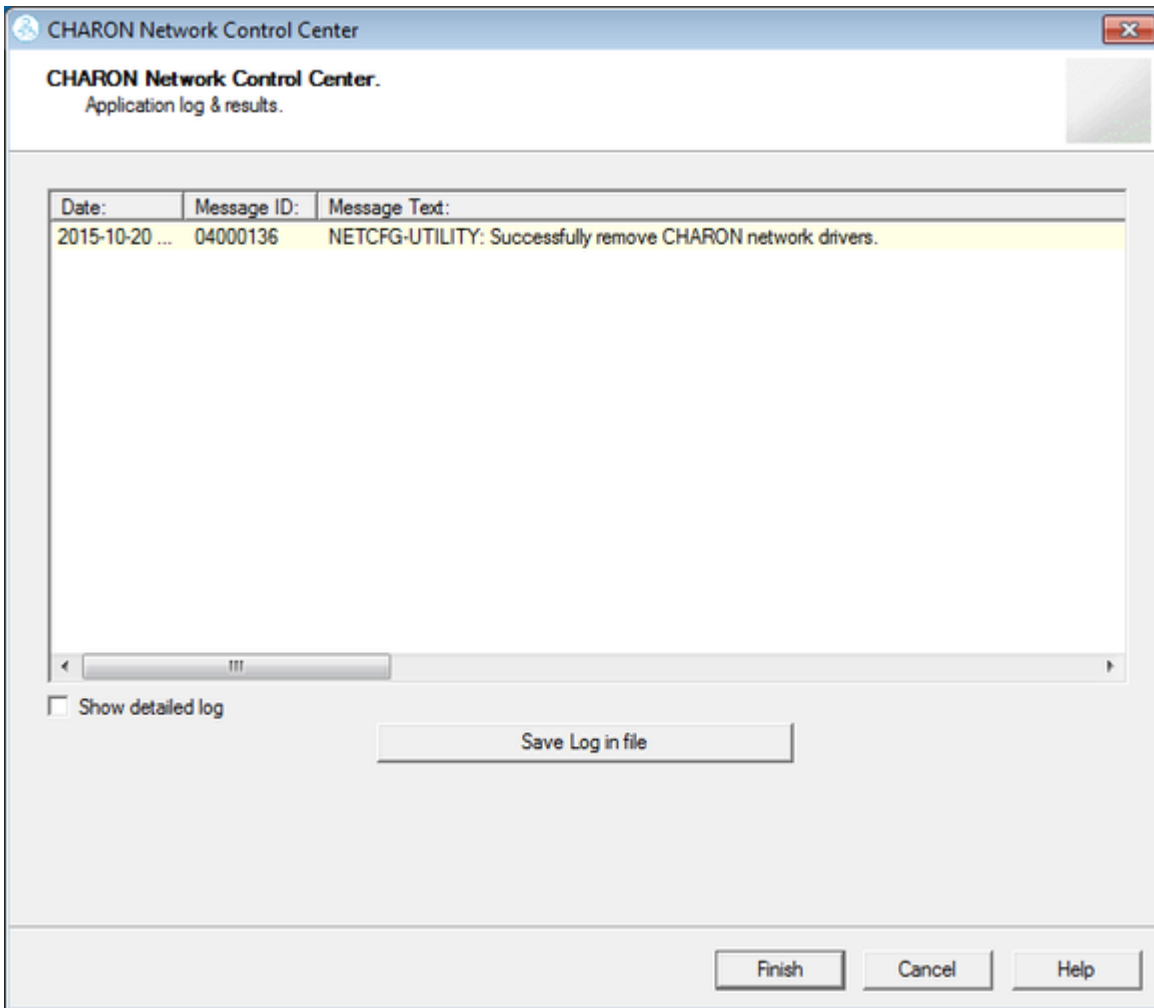
Start the utility and select "Install/Upgrade"; press the "Next" button:



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

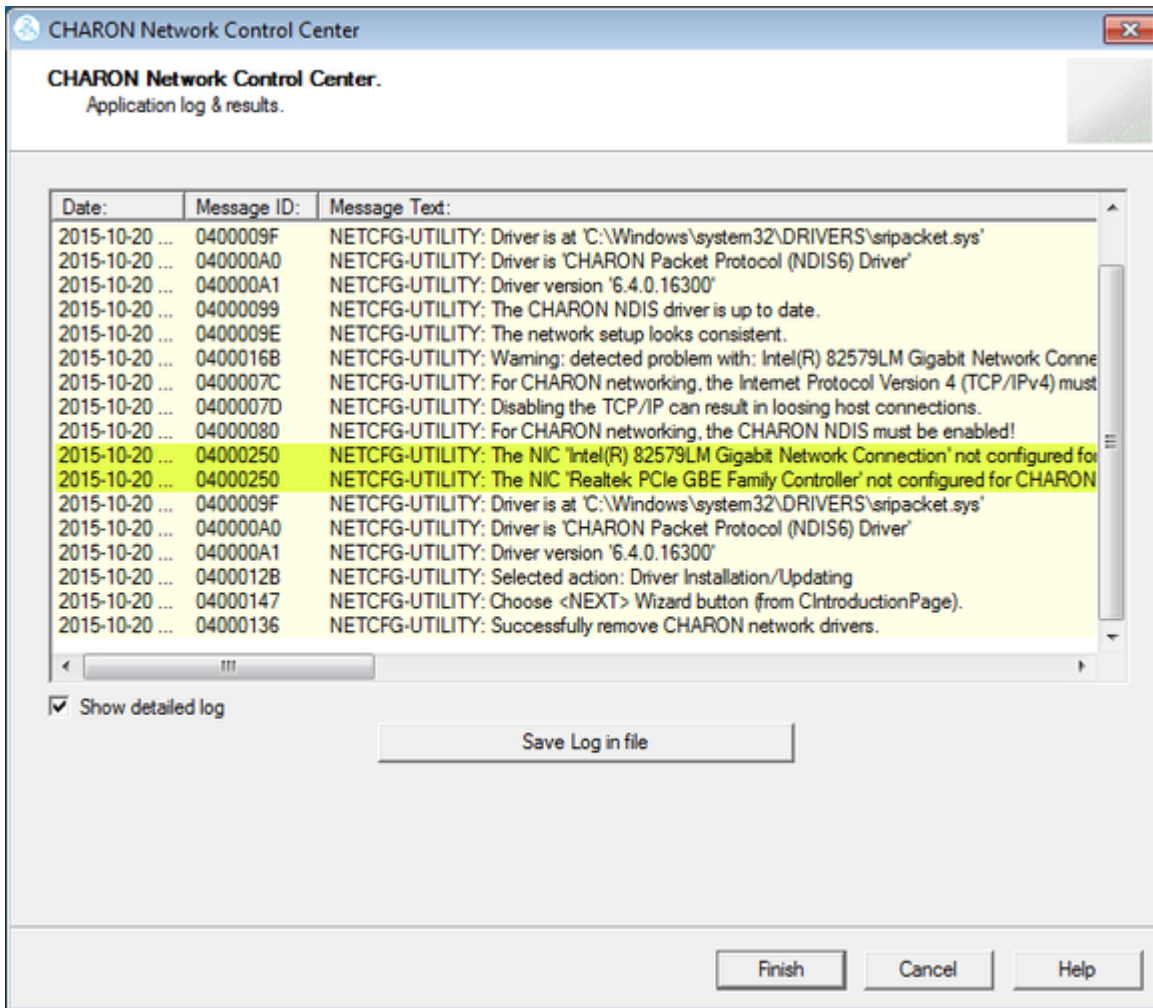


The utility will report that the CHARON drivers have been removed:



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

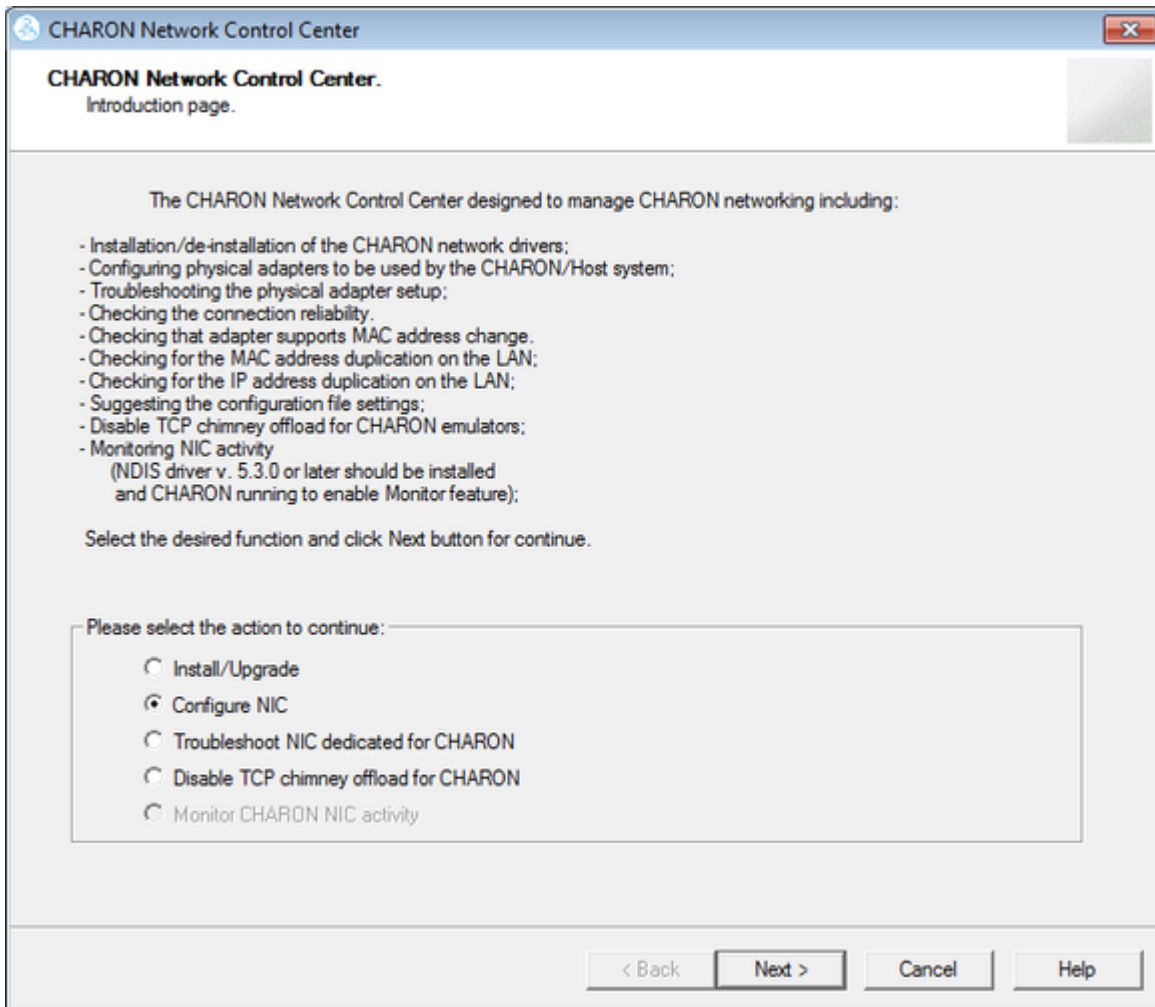
Select "Show detailed log" to display more detail.



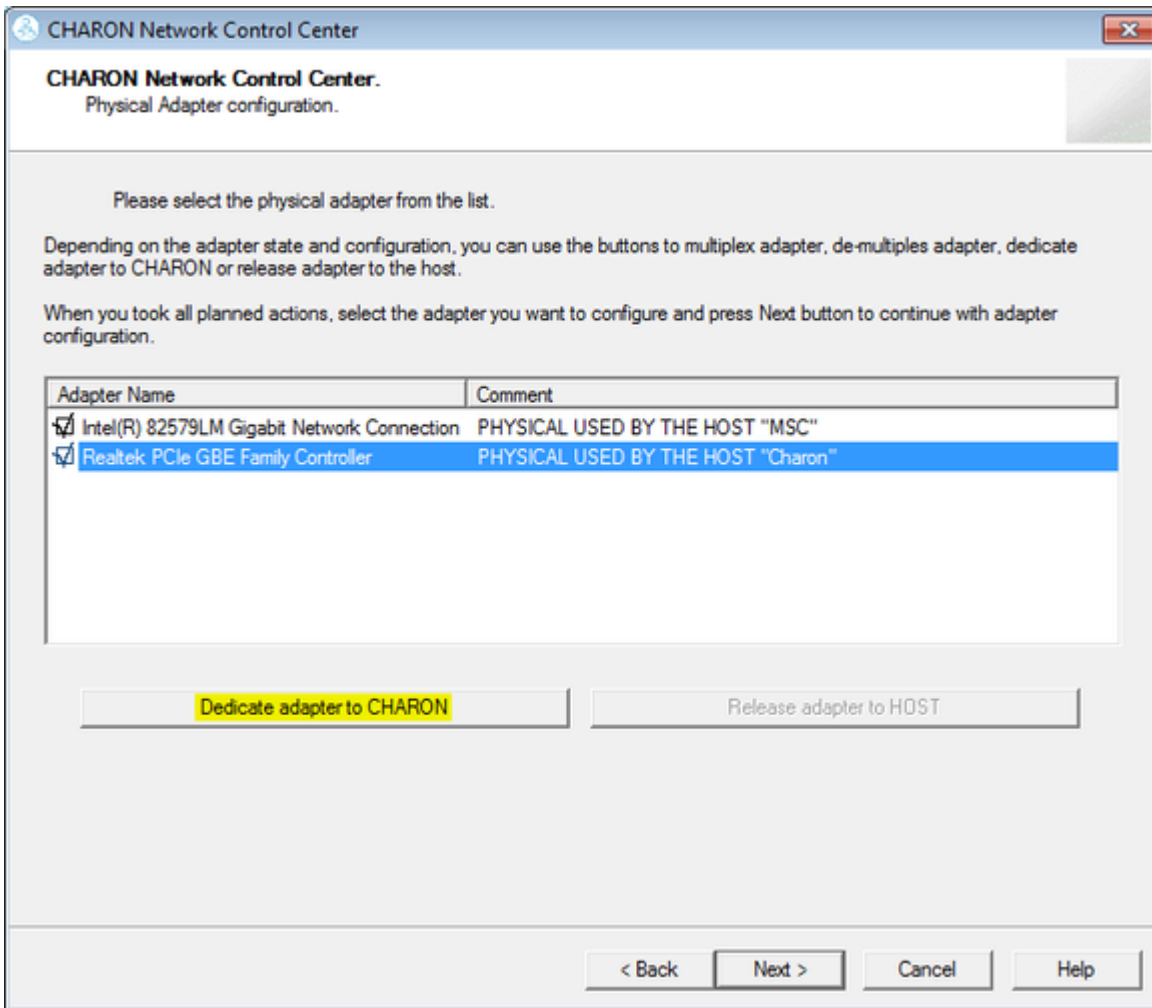
Press the "Finish" button to exit.

## Configuring the host network interfaces for CHARON

Start the utility, select "Configure NIC" and press the "Next" button:



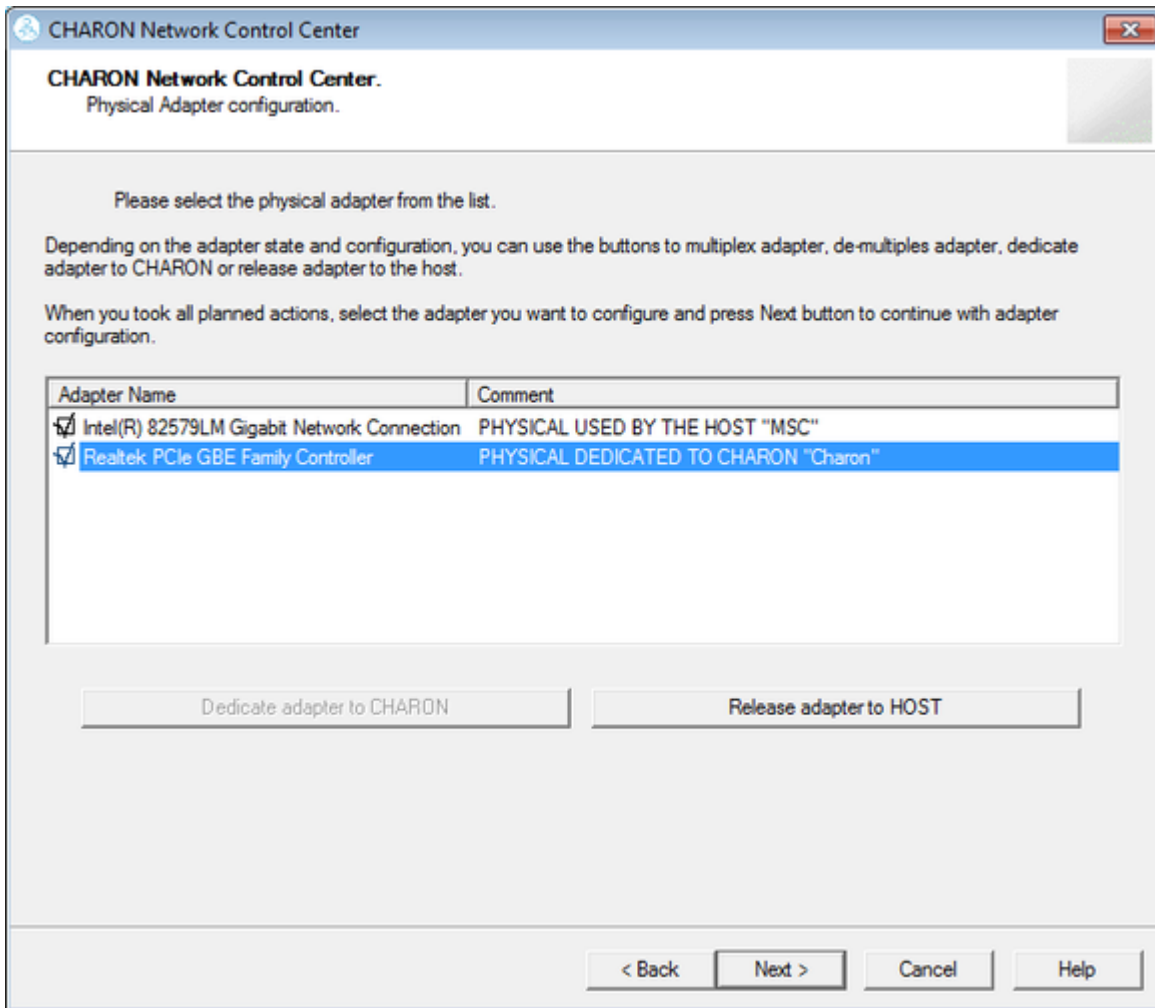
Select the host interface to be dedicated to CHARON (in the example below it is "Charon"), press "Dedicate adapter to CHARON":



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 the "Next" button.



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

**CHARON Network Control Center.**  
Proposals for configuration file records.

The CHARON Network Control Center Config File suggestion step.

Please select the network interface you plan to use (depending on emulated model) and check the suggested configuration file settings with one you are using. The settings are suggested for primary adapter only.

For secondary adapter configuration and adapter options (see the your system's 'User Guide' for details).

QBUS systems (MicroVAX 3600, MicroVAX 4000 106/108, ...) - XQA  DEQNA  DELQA  DESQA

SCSI systems (MicroVAX 3100 96/98, MicroVAX 4000 106/108, ...) - EZA  SGEC

VMI systems (VAX 66x0) - EXA  DEMNA

**CHARON-AXP (AlphaServer DS10, DS20, ES40, GS80, GS160, ...) - EWA**  DE500AA  DE500BA  DE435

CHARON-AXP (AlphaServer DS10, DS20, ES40, GS80, GS160, ...) - EIA  DE602

DEBNI systems (VAX 81 or VAX 6000) - DEBNI  DEBNI

PMAD-AA (TurboChannel add-on card for VAX 4000-90A)  PMADAA

NIC connection name.  
load DE500AA/dec21x4x EWA interface=EWA0  
load packet\_port/chnetwork EWA0 interface="connection:Charon"

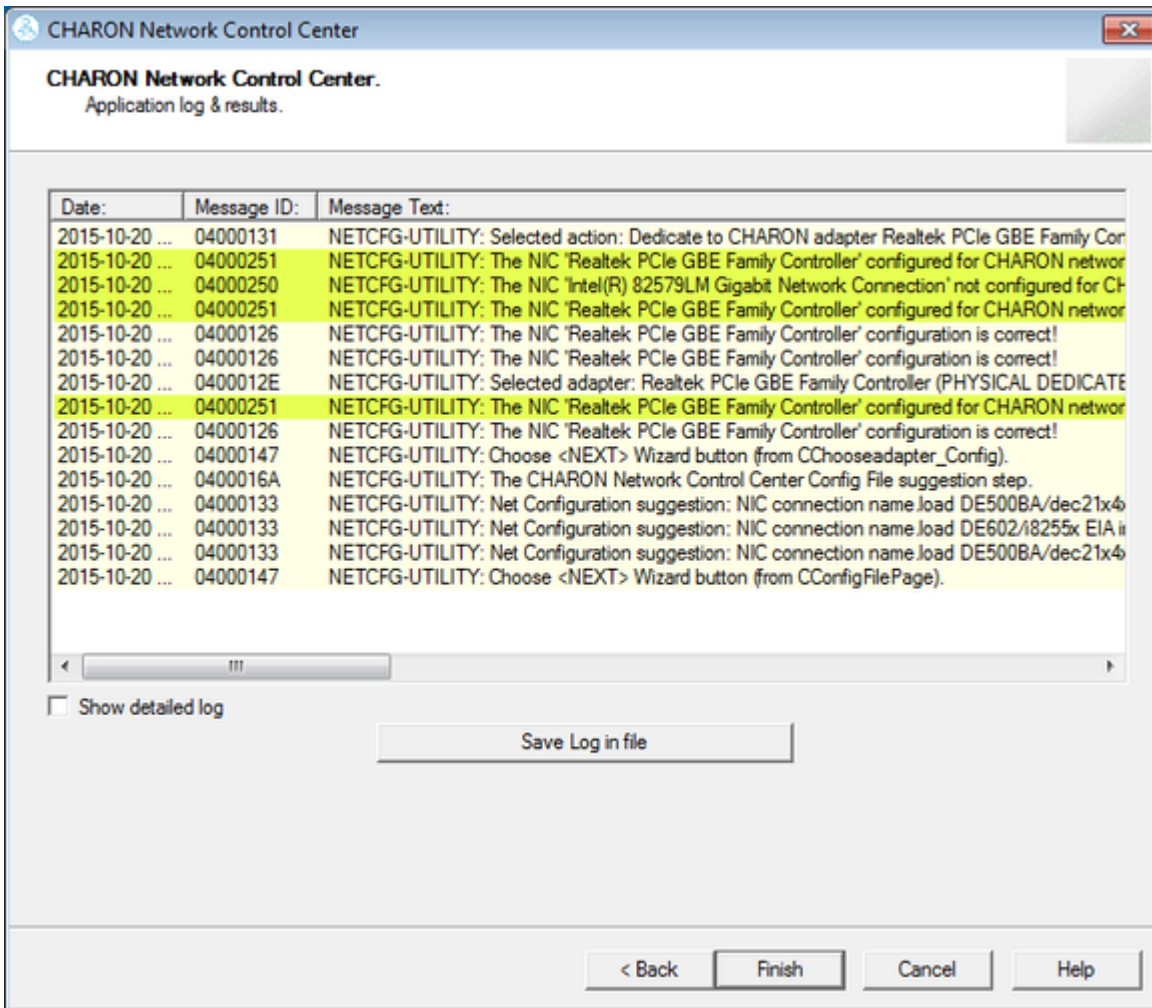
Copy suggestions to clipboard

< Back   Next >   Cancel   Help

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

```
...
load DELQA/DEQNA XQA interface=XQA_0
load packet_port/chnetwork XQA_0 interface="connection:Charon"
...
```

Press the "Next" button to see the log file:



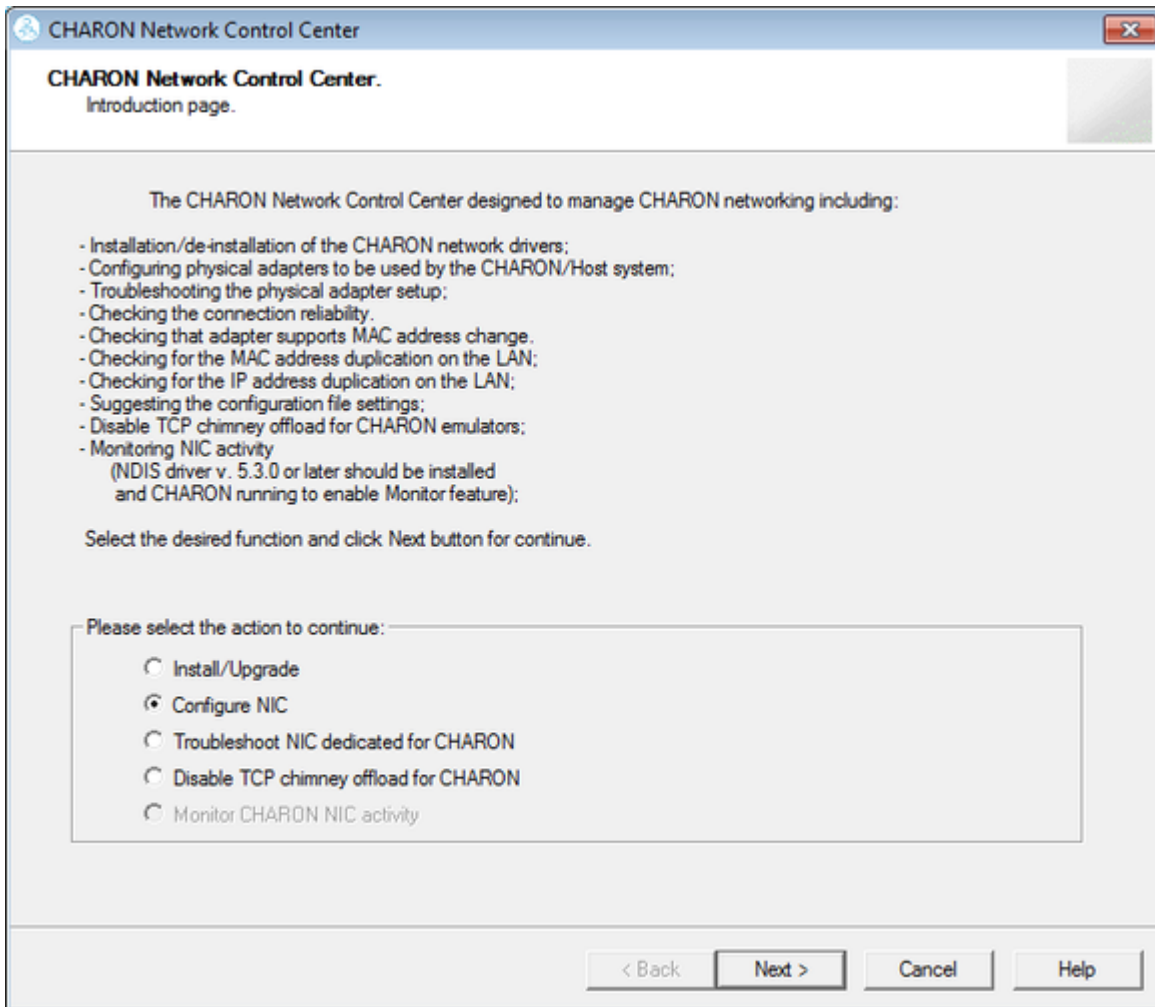
Review the log for errors. Save the log to a file by pressing the "Save Log in file" button in case the log file has to be sent to Stromasys Customer Support.

Select "Show detailed log" to display more detail.

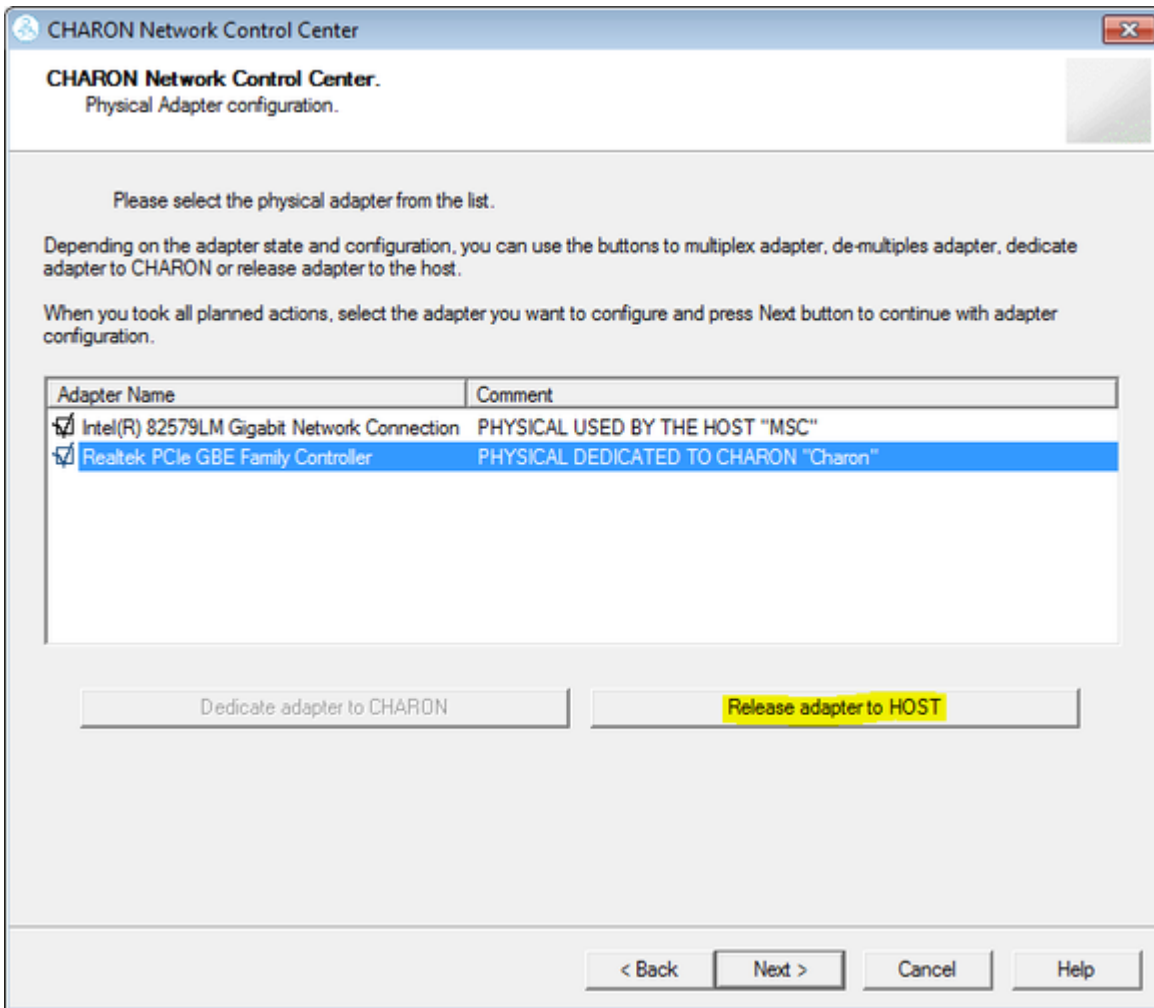
Press the "Finish" button to exit.

## Release of the host network interfaces

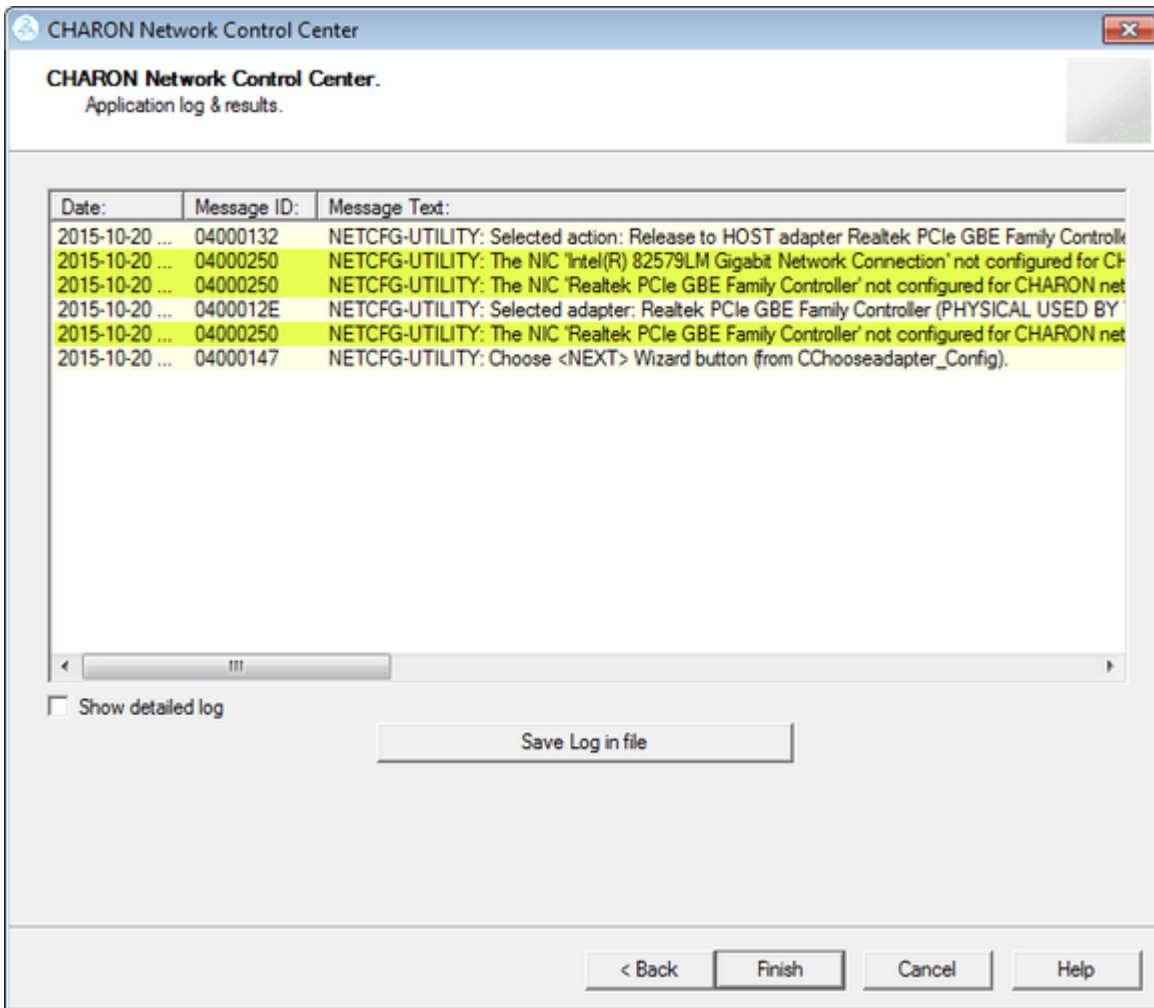
Start the utility, select "Configure NIC" and press the "Next" button:



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



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

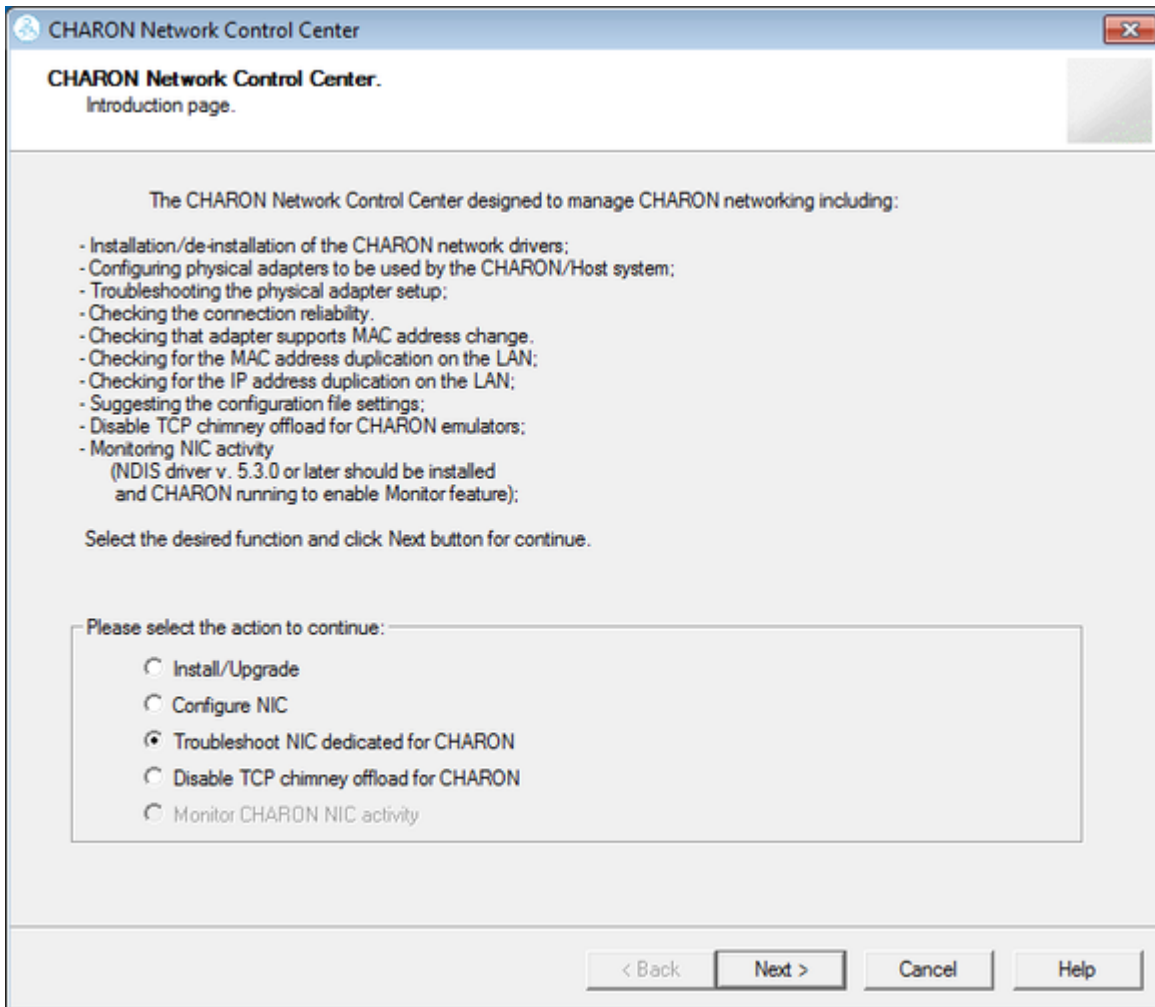


Select "Show detailed log" to display more detail.

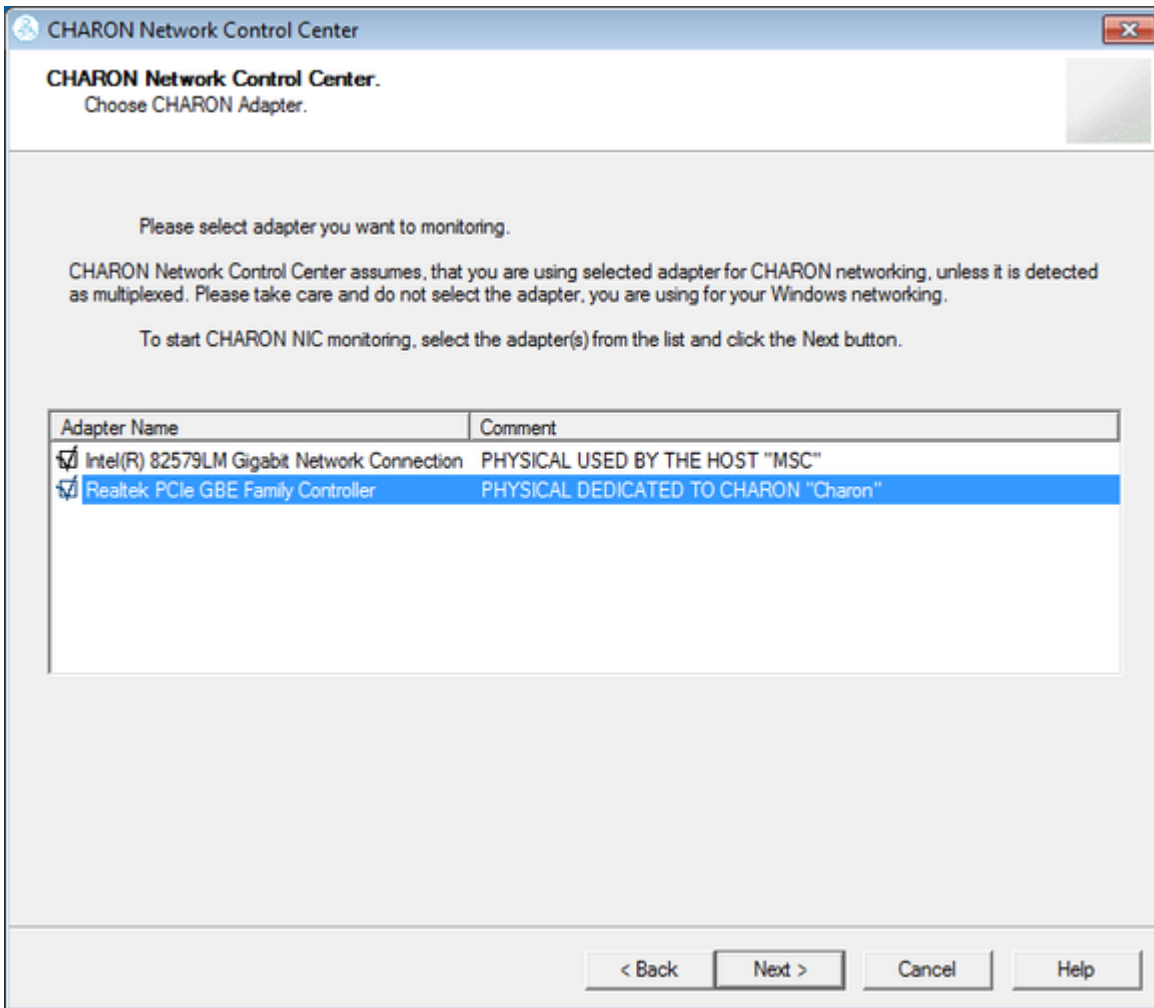
Press the "Finish" button to exit.

## Troubleshooting the CHARON network interfaces configuration

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

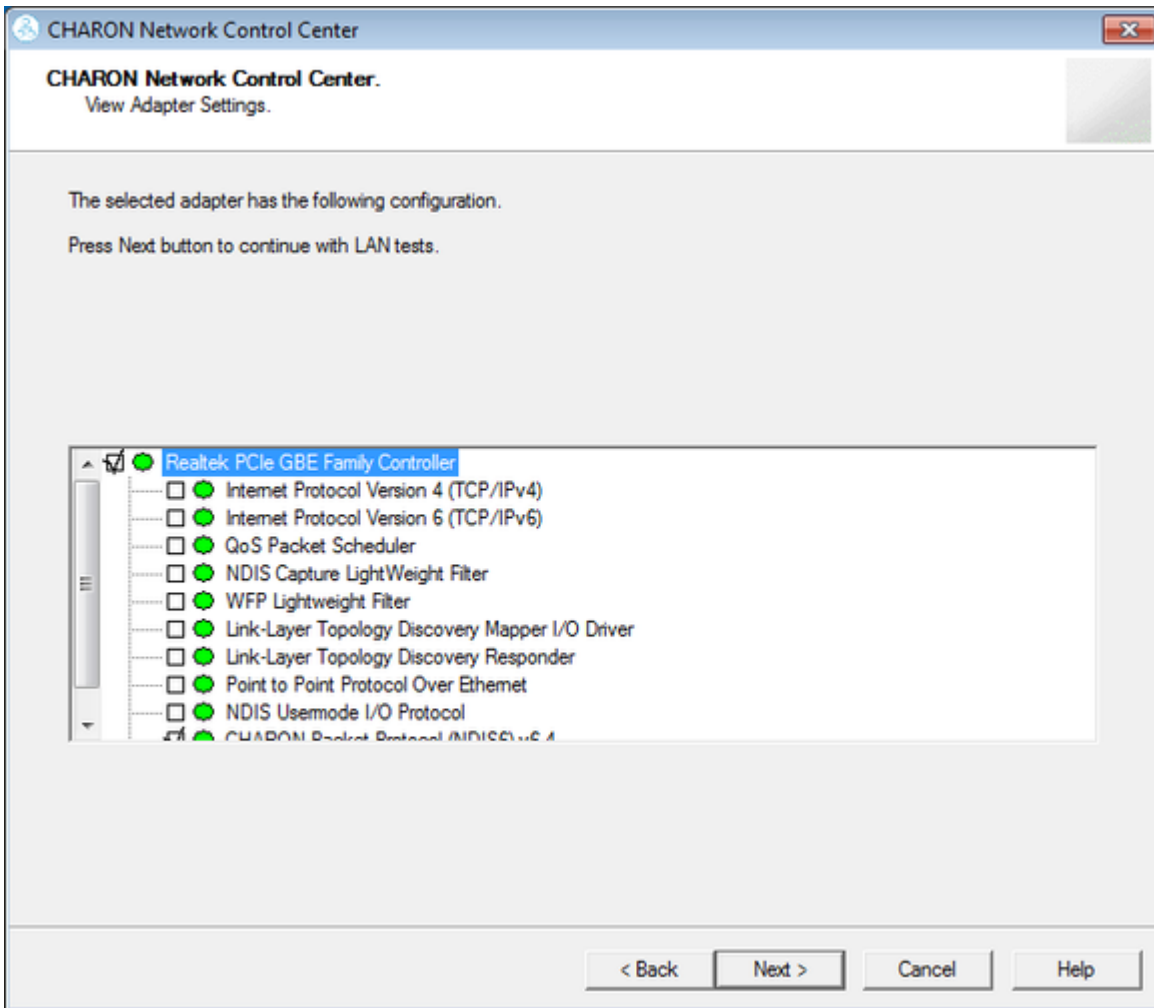


Select the target interface and press the "Next" button:

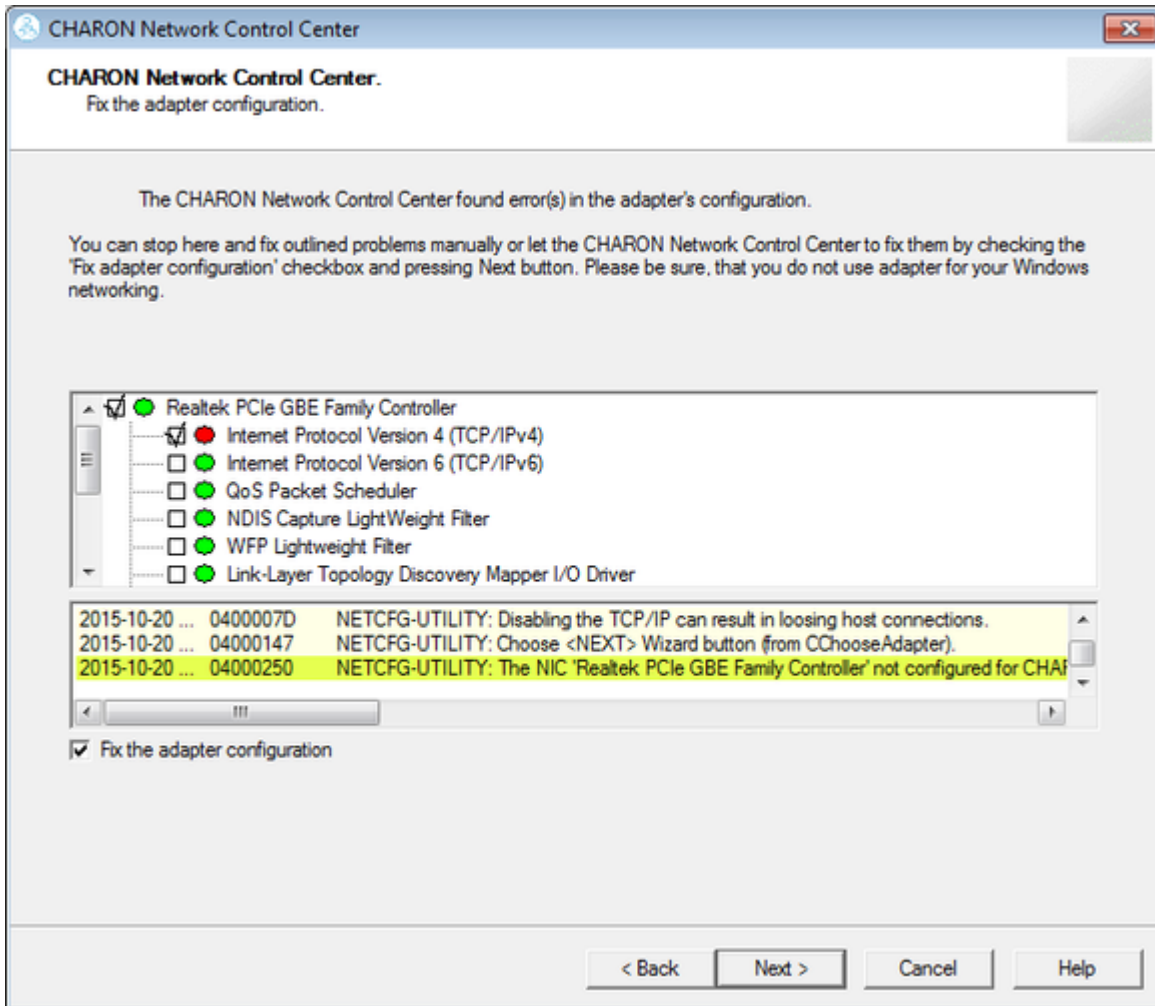




Review the status of the interface:



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



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

The screenshot shows a Windows-style dialog box titled "CHARON Network Control Center". The main content area contains the following text:

**CHARON Network Control Center.**  
Step 1 LAN tests.

The CHARON Network Control Center Step 1 LAN tests.

To start the Step 1 tests, please define the DECnet address you are using or plan to use with your CHARON installation. If you do not use the DECnet or want to skip the tests check the "Skip tests" checkbox below.

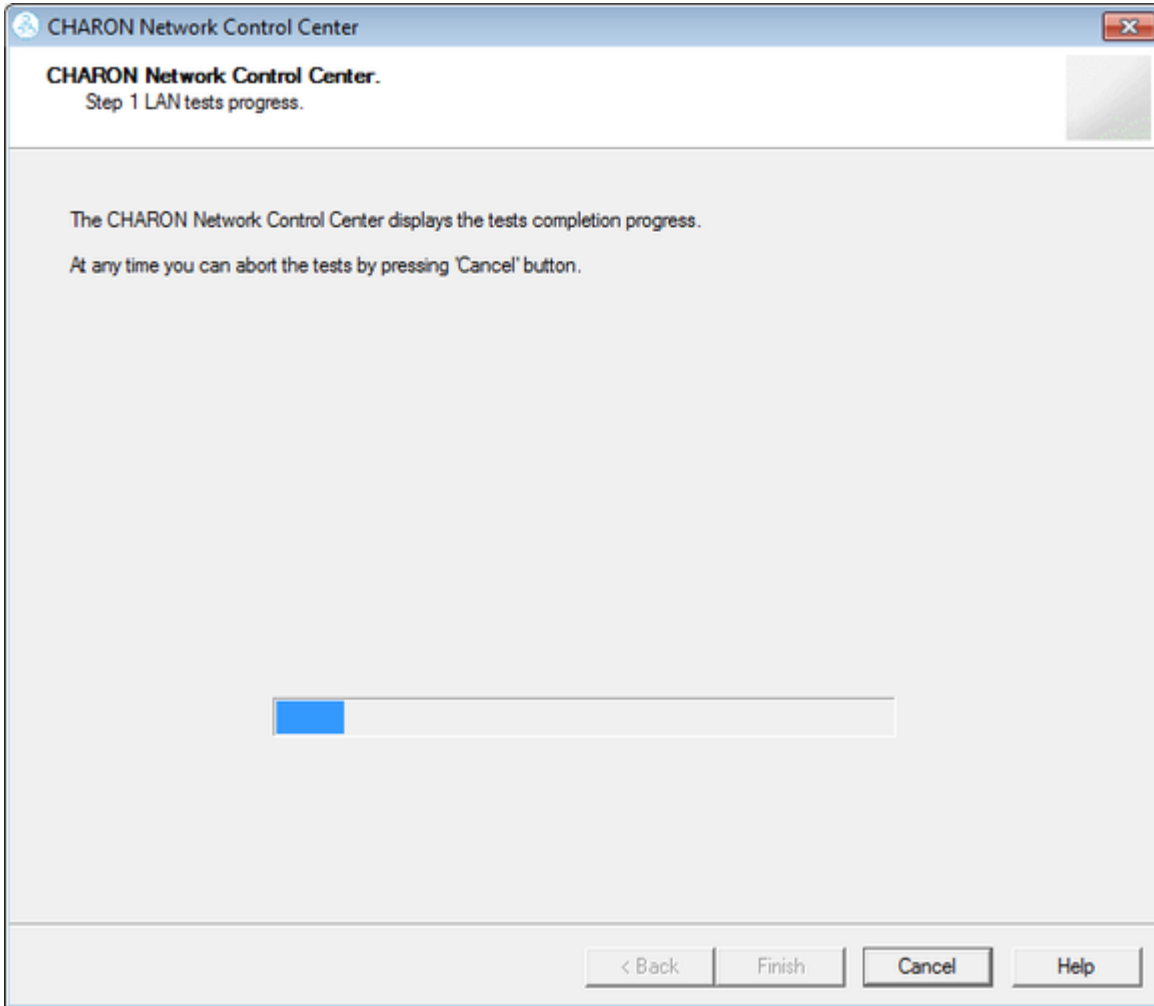
The whole Step 1 LAN tests took approximately 3 minutes to complete.

Desired DECnet address   

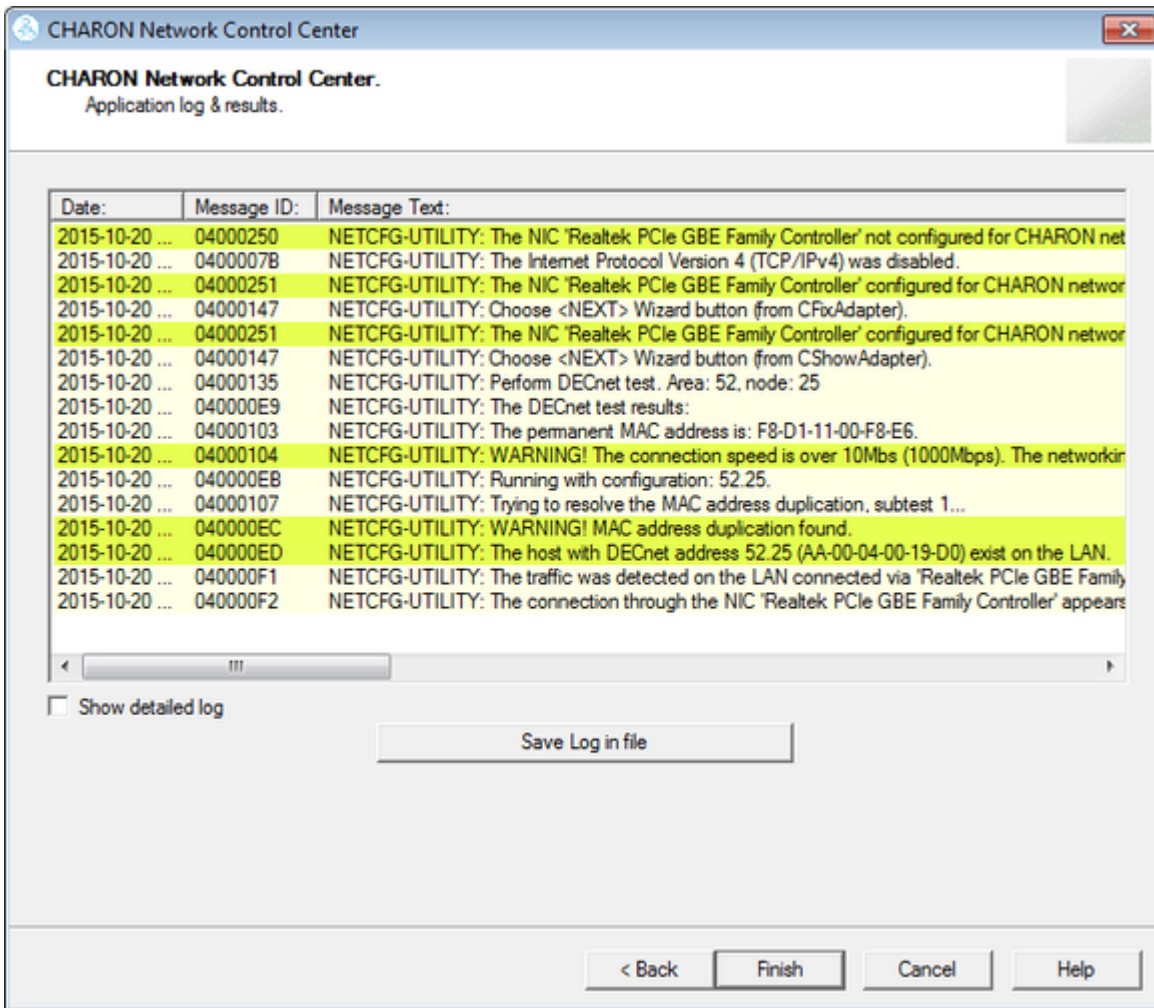
Skip tests

At the bottom of the dialog, there are three buttons: "< Back", "Next >", and "Help".

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 the "Next" button:



Once the checking is completed, if there are 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 title is "CHARON Network Control Center." followed by the subtitle "Step 2 LAN tests." Below this, there is a large text area containing the following information:

- A heading: "The CHARON Network Control Center Step 2 LAN tests."
- Instructions: "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: "WARNING: The IP address duplication test can result in the loosing connections by the Windows host that using the IP address you specify."
- Duration note: "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."

At the bottom of the text area, there are two input fields:

- "Desired IP address (or subnet address)" with a text box containing ". . ."
- "Subnet mask" with a text box containing ". . ."

Below the input fields is a checkbox labeled "Skip tests" which is currently unchecked.

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

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

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.

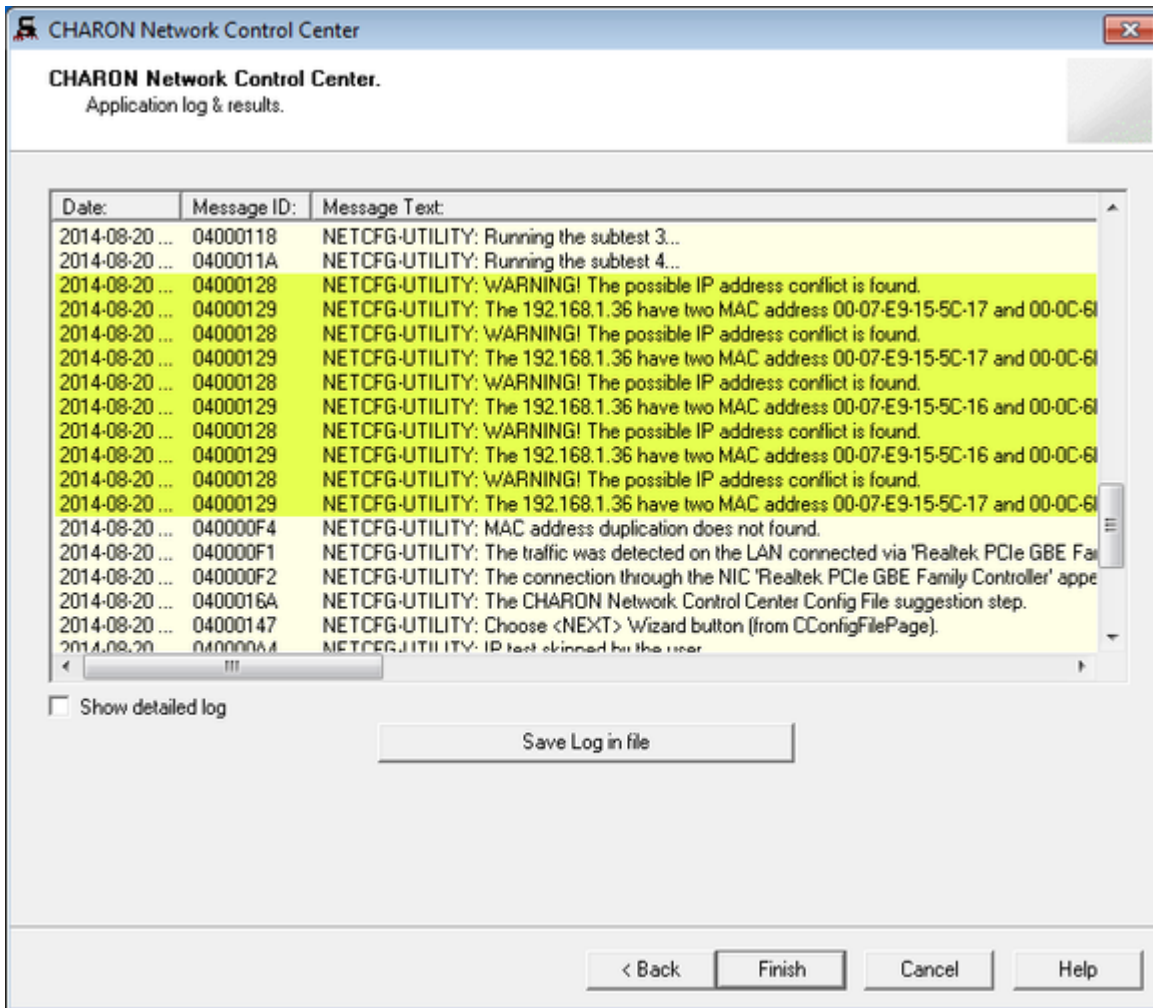
Desired IP address (or subnet address)

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.

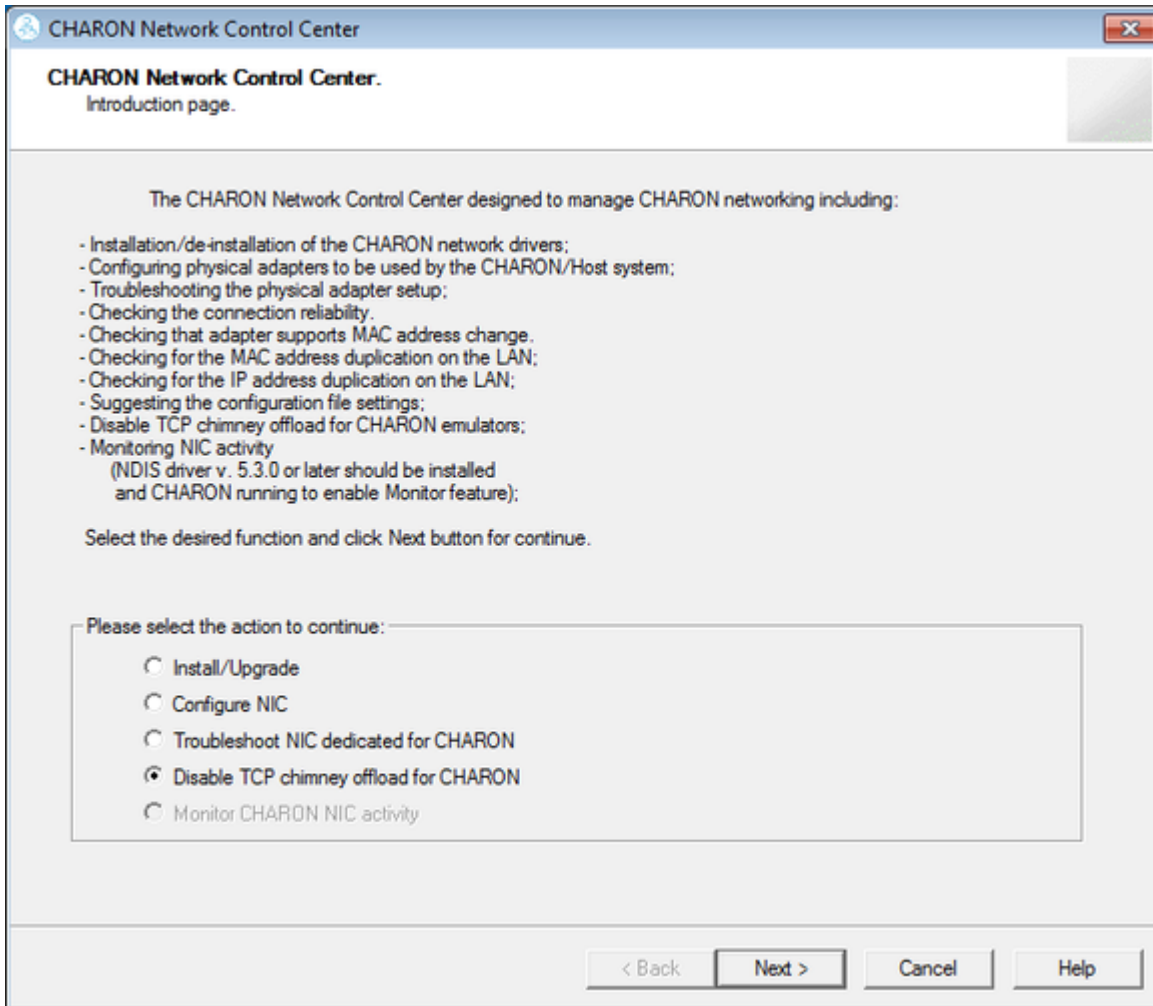
Press the "Finish" button to exit.



## Disable "offload" parameters

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

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








The "offload" parameters will be reset for all CHARON emulators (executable files).

Press the "Finish" button to exit.

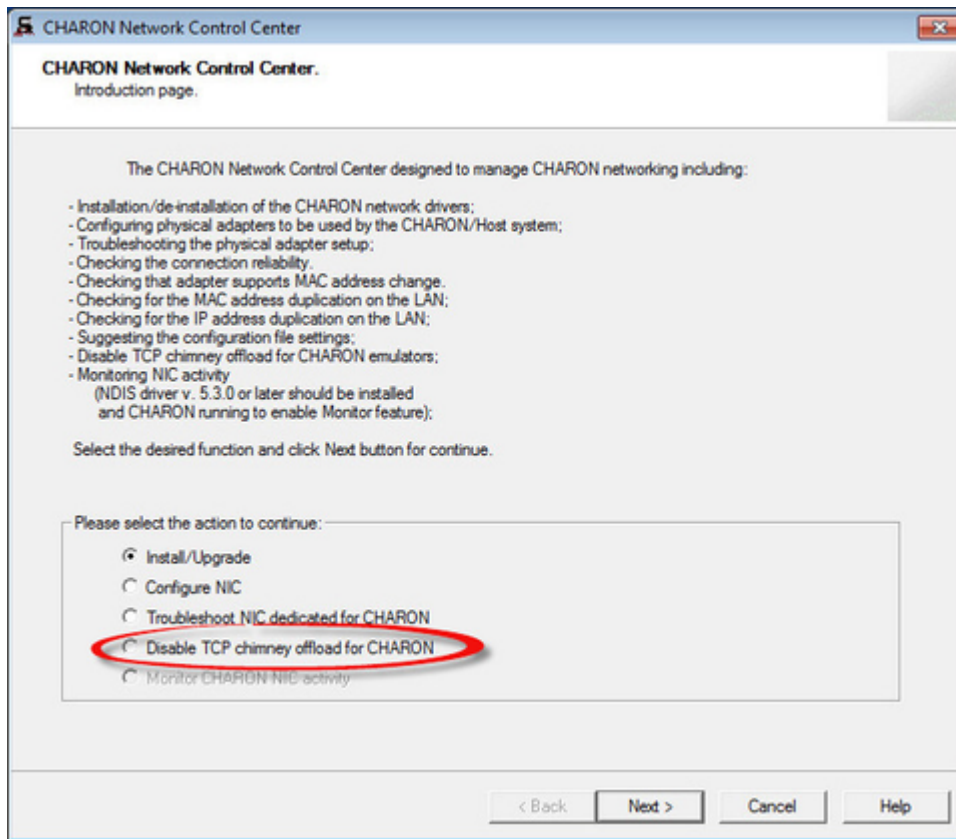
**!** Please note to restart the CHARON host to enable this settings.

Additional information regarding TCP Chimney Offload:

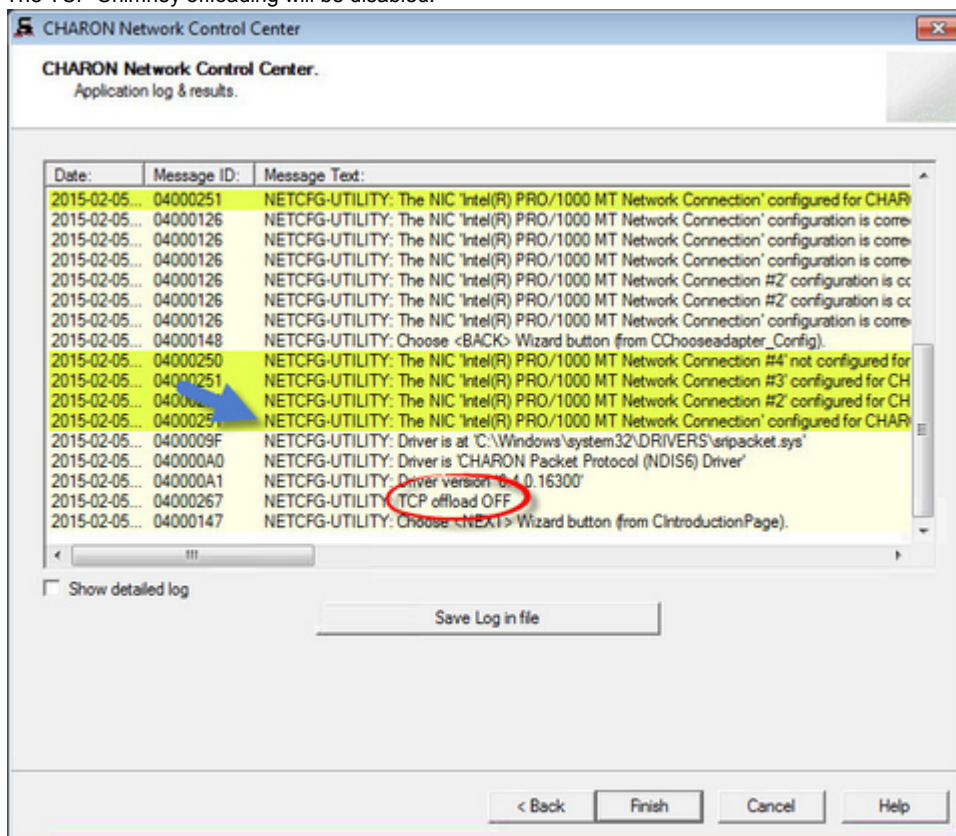
Operating system	Notes	Charon version	Configuration steps
Windows Server 2019	TCP Chimney offload is disabled by default (deprecated technology)	<ul style="list-style-type: none"> <li>Charon-AXP/PDP/VAX V4.11 Build 204-03 and above</li> </ul>	None
Windows Server 2008 R2	TCP Chimney offload must be disabled. <b>Note:</b>	<ul style="list-style-type: none"> <li>CHARON-VAX V4.7 and CHARON-AXP V4.7 Build 171-10 and above (patched)</li> <li>Charon-AXP/PDP/VAX V4.8 Build 183-02 and above (patched)</li> </ul>	Configuration type 1
Windows Server 2012 R2	<ul style="list-style-type: none"> <li>TCP Chimney Offload is a networking technology that helps transfer the workload from the CPU to a network adapter during network data transfer.</li> <li>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.</li> </ul>	<p><b>Older versions of Charon-AXP/PDP/VAX</b></p> <p> Do not use the network control center utility to disable TCP chimney offload. Please use only the manual settings described in this chapter</p>	Configuration type 2
Windows Server 2016			
Windows 7			
Windows 8 & 8.1			
Windows 10	<p> Running guests will have to be powered off/on for the new settings to be taken into account.</p> <p> To check TCP Chimney offload is disabled, use the following commands:</p> <pre>C:\Windows\system32&gt;netsh int tcp show global Querying active state...  TCP Global Parameters ----- Receive-Side Scaling State      : enabled Chimney Offload State          : disabled NetDMA State                   : disabled Direct Cache Access (DCA)     : disabled ...</pre> <p> To revert back to the original settings manually, use the following command:</p> <pre>C:\Windows\system32&gt;netsh int tcp set global chimney=automatic</pre>		
Older versions of Windows	 Running guests will have to be powered off/on for the new settings to be taken into account	Charon-AXP/PDP/VAX versions supported on these old versions of Windows	Configuration type 3

## Configuration type 1

- Charon-AXP/VAX V4.7:
  - open the **Start** menu, select **"All Programs"** then **"CHARON"**, select your Charon product version, **"Utilities"** then **"Network Controler Center version x.xx"**.
- Charon-AXP/PDP/VAX V4.8:
  - Double click the **"CHARON Virtual Machine Manager"** icon on desktop or select this utility in the tray menu; select **"Host Information and Utilities"** section and press **"Network Control Center"** button.
- From the **"Network Control Center"** utility main page, select the **"Disable TCP chimney offload for CHARON"** option:



- Click on the "Next" button.
- The TCP Chimney offloading will be disabled:



- Click on the **"Finish"** button to exit.

### Configuration type 2

- Open a command line windows with Admin rights
- Issue the following command:

```
C:\> netsh int tcp set global chimney=disabled
```

- Delete all occurrences of Charon executables in the following table if any as described below:

```
C:\> netsh interface tcp show chimney
...
C:\> netsh interface tcp delete chimneyapplication ^
More? application="C:\Program Files\CHARON\Build_16400\x64\as1000.exe"
```

### Configuration type 3

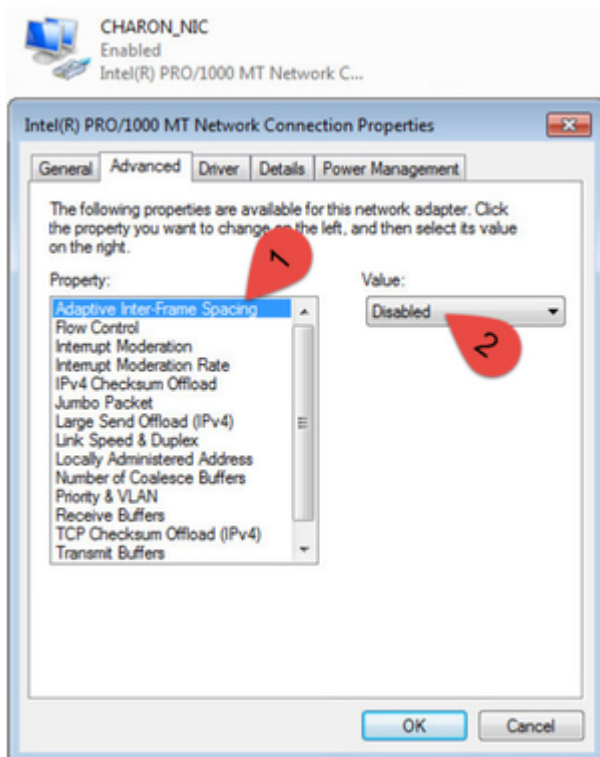
Intelligent packet processing should be switched off for the network adapters dedicated to CHARON-AXP (naming of the parameters depends on the network adapter driver):

- |   |   |
|---|---|
| <ul style="list-style-type: none"> <li>• Adaptive interframe spacing</li> <li>• Flow control</li> <li>• Interrupt moderation</li> <li>• Interrupt moderation rate</li> <li>• Ipv4 checksum offload</li> <li>• Jumbo packet</li> </ul> | <ul style="list-style-type: none"> <li>• Large send offload</li> <li>• Disable Priority and VLAN</li> <li>• TCP checksum offload</li> <li>• UDP checksum offload</li> </ul> |
|---|---|

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.

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

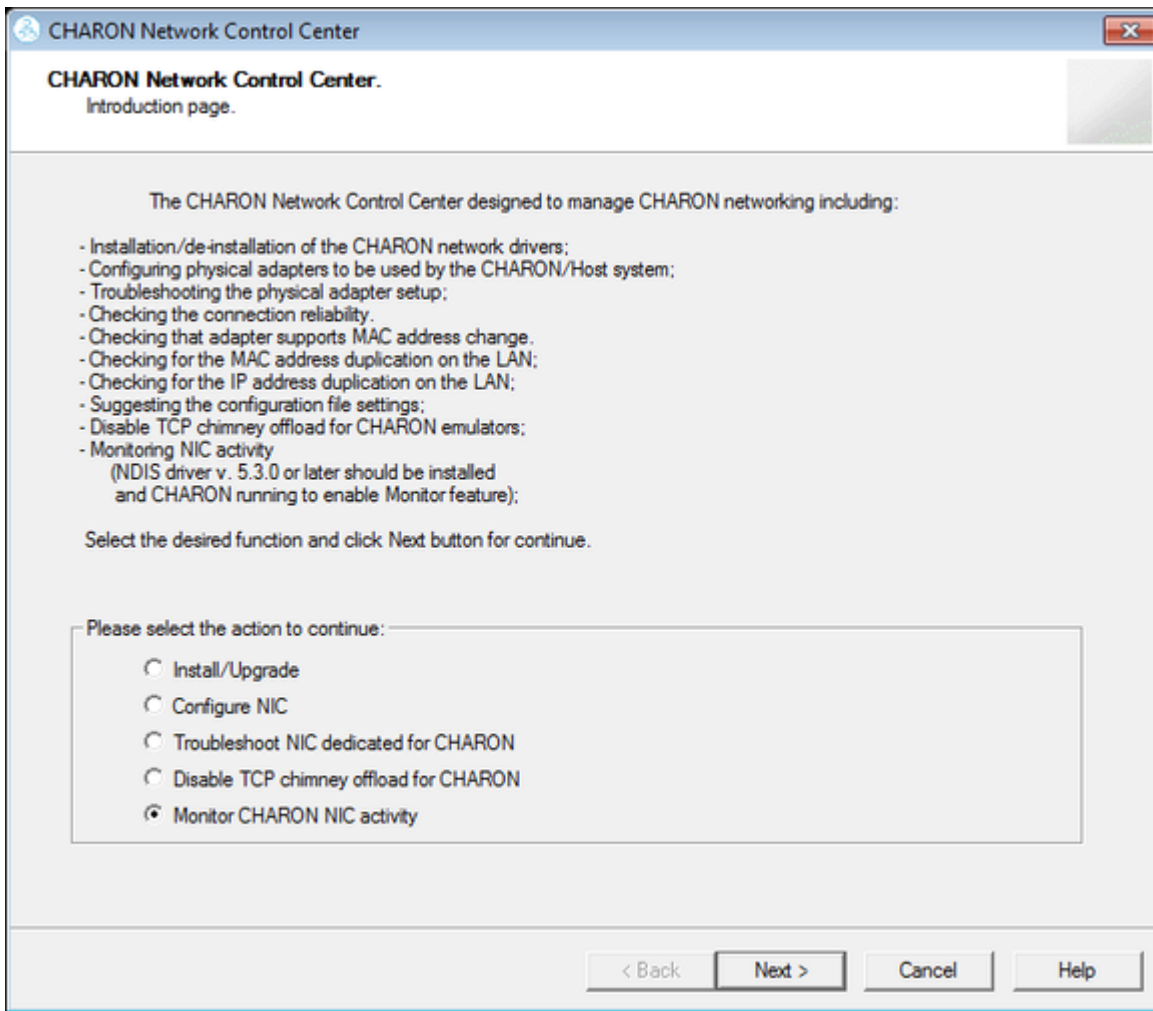
#### Example:



## 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, select "Monitor CHARON NIC activity" and press the "Next" button:



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

CHARON Network Control Center  
CHARON Network Control Center.  
Choose CHARON Adapter.  
  
Please select adapter you want to monitoring.  
CHARON Network Control Center assumes, that you are using selected adapter for CHARON networking, unless it is detected as multiplexed. Please take care and do not select the adapter, you are using for your Windows networking.  
To start CHARON NIC monitoring, select the adapter(s) from the list and click the Next button.  

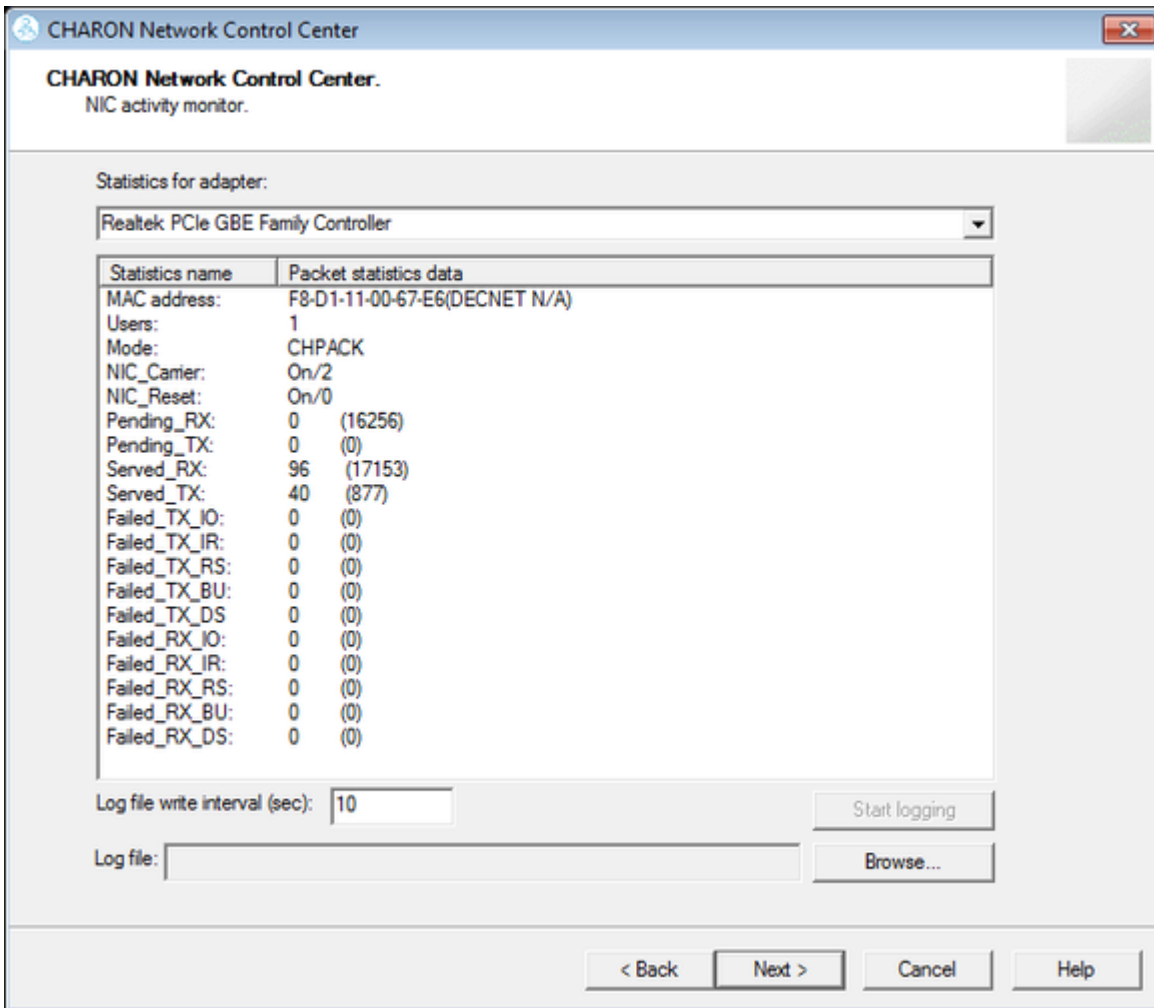
Adapter Name	Comment
<input checked="" type="checkbox"/> Intel(R) 82579LM Gigabit Network Connection	PHYSICAL USED BY THE HOST "MSC"
<input checked="" type="checkbox"/> Realtek PCIe GBE Family Controller	PHYSICAL DEDICATED TO CHARON "Charon"

Reset Adapter Statistics

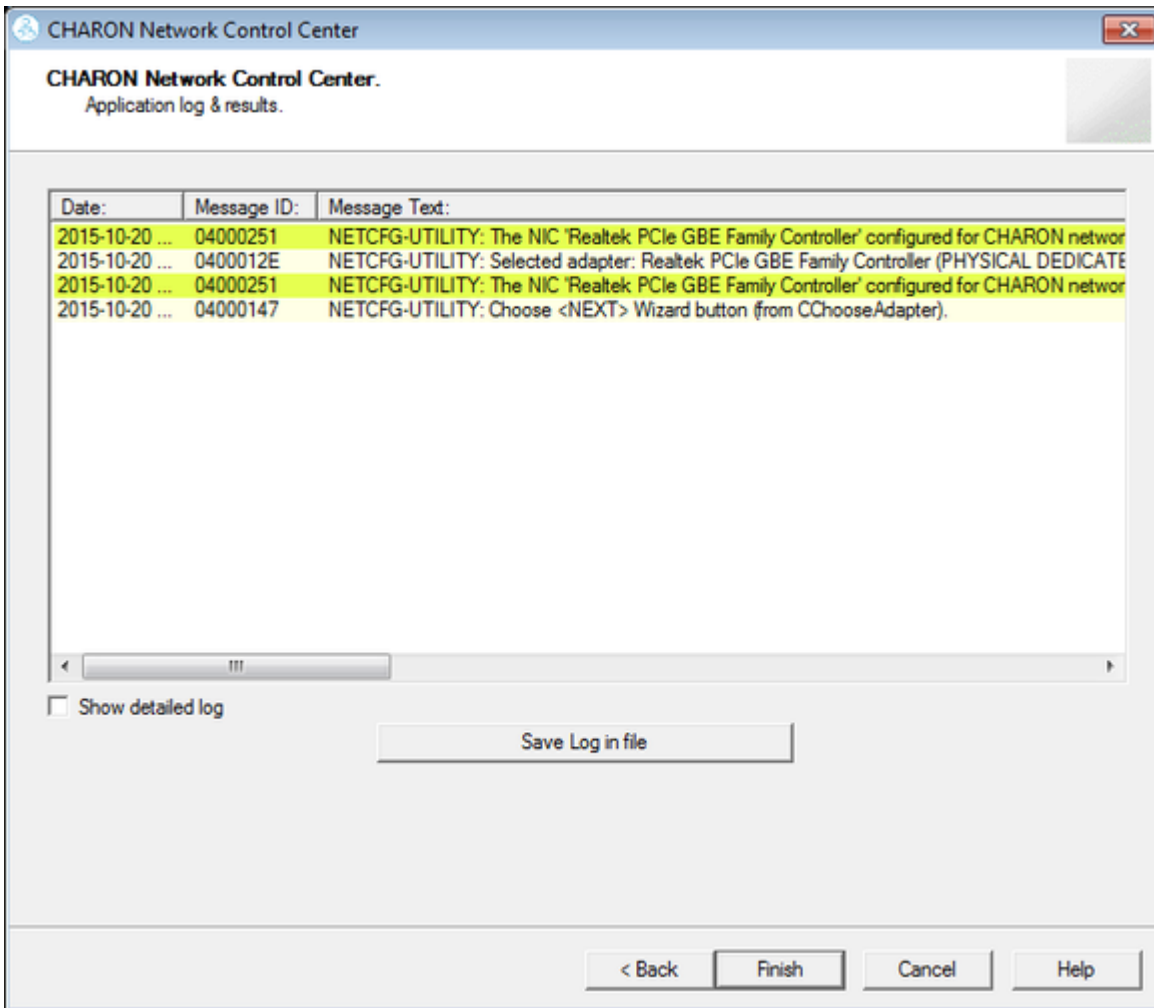
< Back   Next >   Cancel   Help

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



Select "Show detailed log" for more detail.

Press the "Finish" button to exit.



# MkDisk

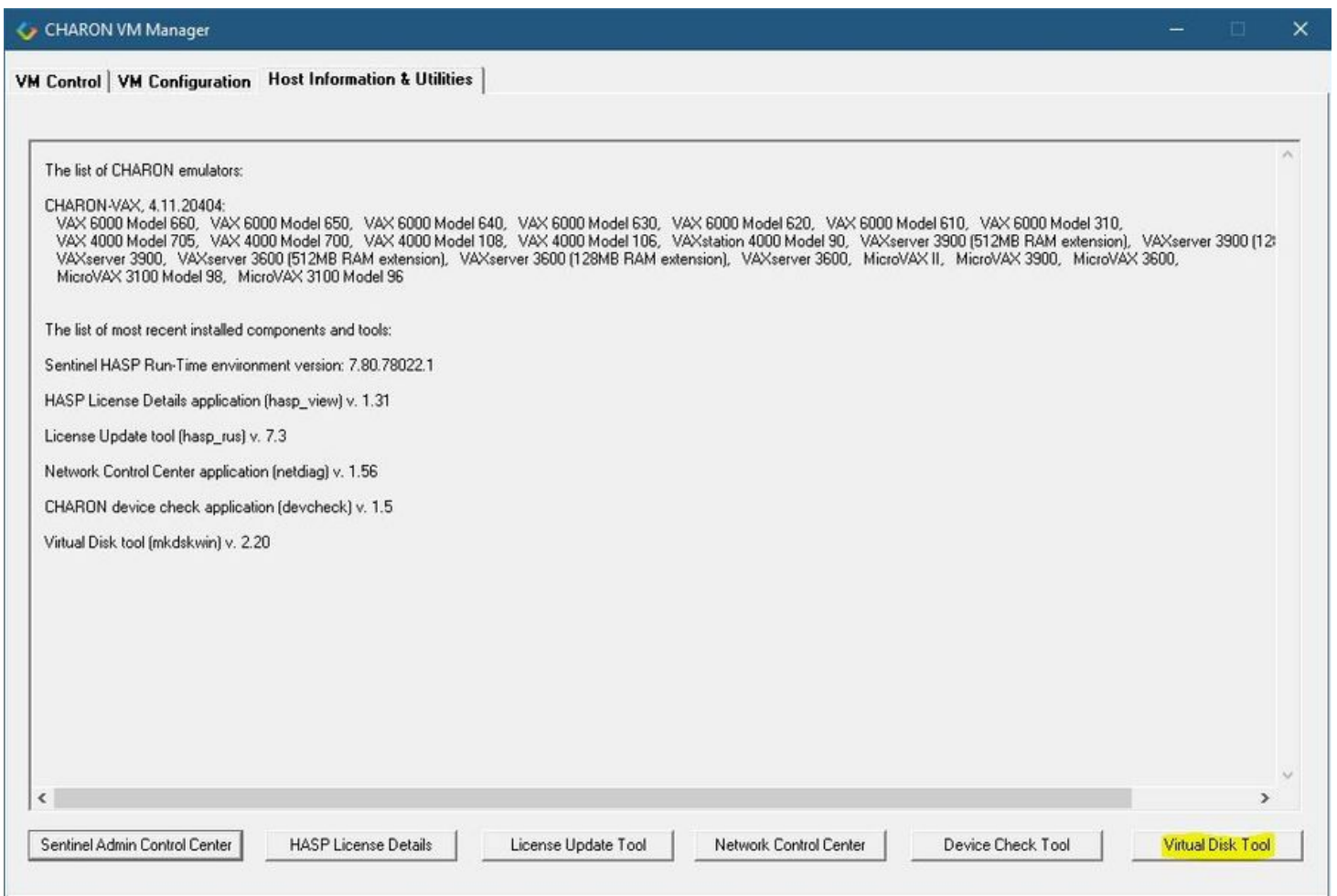
## Table of Contents

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

## General Description

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

Open the CHARON Virtual Machine Manager, switch to the "Host Information & Utilities" tab and press the "Virtual Disk Tool" button:



The main dialog of the utility will appear:

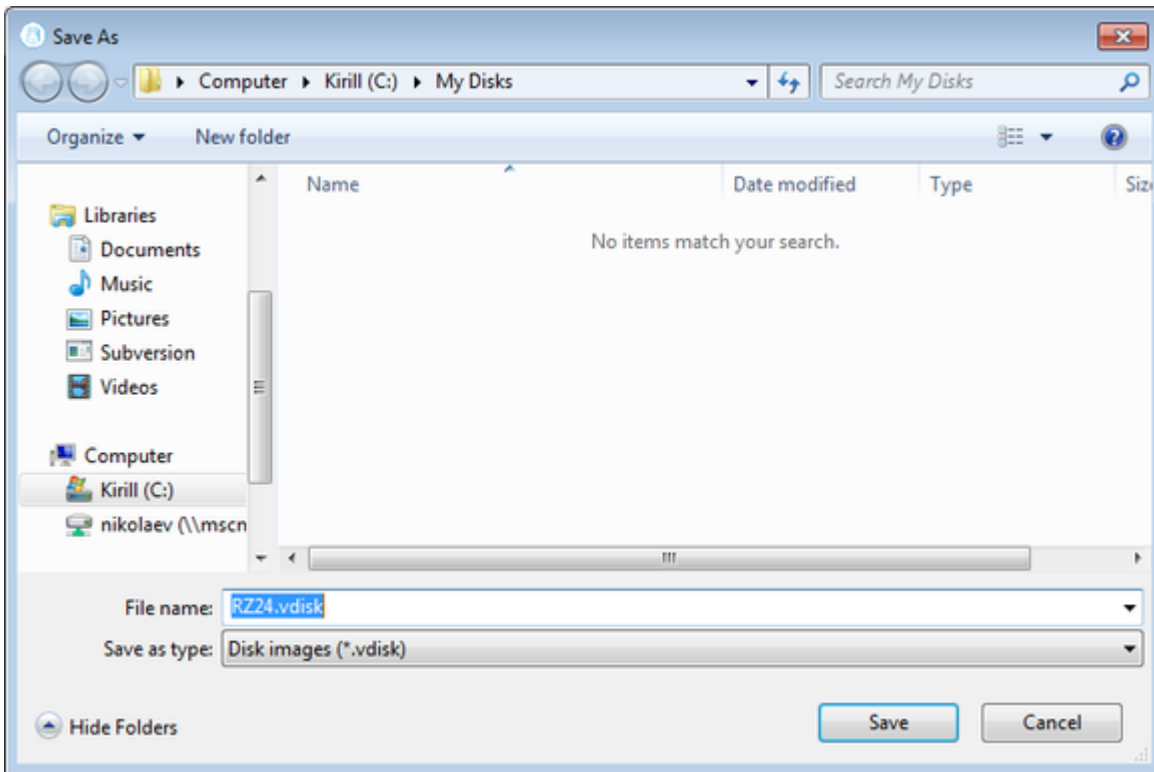


## Creating empty disk container

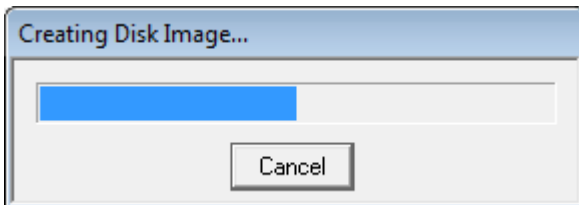
Select "VAX" or "PDP-11" in the "Select System" drop-down box, "MSCP" in the "All Controllers" drop-down box and choose the desired disk type in the "Select Disk Type" drop-down box to create an empty disk container:



Press the "Create Disk Image" button 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 the "Save" button. 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.

## Creating disk metadata

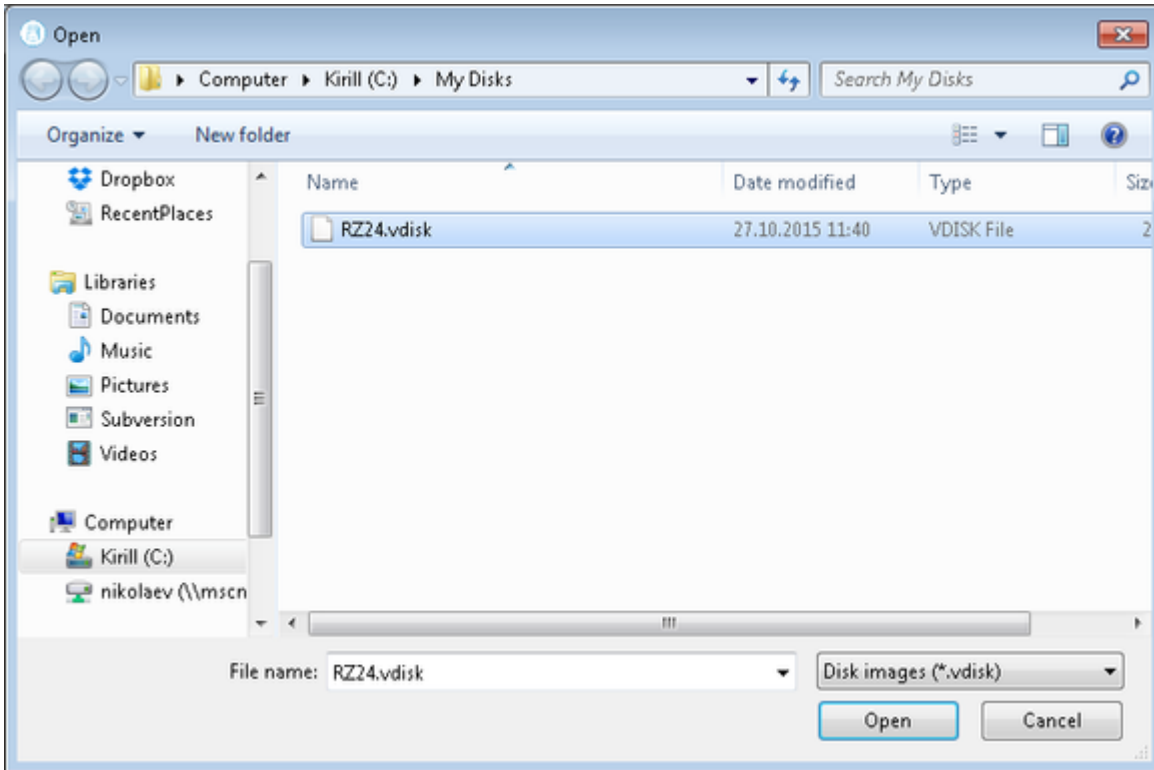
The disk image metadata file contains important information about the target disk image structure - it is used by CHARON, automatically created with the same name with a different extension.

This functionality is needed if you have a disk image of a certain type and you want to generate a metadata file for it.

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



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 the "Open" button. The utility will display a dialog for adjusting/correcting the disk image parameters:

The screenshot shows a "Metadata configuration dialog" window with a close button (X) in the top right corner. The dialog is divided into several sections:

- Geometry:** Contains five input fields: "Bytes per sector" (512), "Sectors per track" (38), "Tracks per cylinder" (8), "Cylinders per unit" (1348), and "Sectors per unit" (409792).
- SCSI:** Contains five input fields: "Vendor" (DEC), "Product" (RZ24), "Revision" (0200), "UDID" (empty), and "WWID" (empty).
- MSCP:** Contains two input fields: "Media type class" (DK) and "Media type name" (RZ24).
- SCS:** Contains two input fields: "Node name" (RZ24) and "System ID" (0).
- Special:** Contains a "Controller ID" dropdown menu (MSCP), a "Description" text box (DEC RZ24 WINCHESTER 209 MB 3.5"), and "Bad block table" radio buttons (YES and NO, with NO selected).

At the bottom of the dialog are two buttons: "OK" and "Cancel".

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

## Creating custom disk image

Select "Custom" in the "Select System" drop-down list, 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 the "Create Disk Image" button to proceed:

The screenshot shows the MKDISK utility window. At the top, there is a menu bar with "File" and "Help". Below the menu bar is the Stromasys logo and the text "stromasys legacy server emulation". The main interface consists of several input fields and buttons:

- A dropdown menu for "Select System" with "VAX" selected.
- A dropdown menu for "Select Model" with "MSCP" selected.
- A dropdown menu for "Select System" with "Custom" selected.
- A text input field labeled "Enter desired size in blocks and the block size".
- A "Disk Properties" section with the following fields:
  - Size: 27.0 Gb
  - Number of blocks: 5665667
  - Sectors: (empty)
  - Block size: 512
- Three buttons: "Create Disk Image", "Custom Disk...", and "Create Metadata...".
- A URL link: [WWW.STROMASYS.COM](http://WWW.STROMASYS.COM)

Specify the location and name of the target disk container and press the "Save" button. The utility will create the disk image and will open the "Metadata Configuration dialog" window (for the .avdisk) → click OK to continue.



## 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:

The screenshot shows a dialog box titled "Supported disks" with a close button (X) in the top right corner. The dialog contains a table with four columns: "Device", "Description", "Platform", and "Size (Kb)". The table lists various disk models from RZ22 to RZ59, all with a platform of "PDP,VAX". The sizes range from 51216 Kb to 8877807 Kb. A vertical scrollbar is visible on the right side of the table. An "OK" button is located at the bottom right of the dialog box.

Device	Description	Platform	Size (Kb)
RZ22	DEC RZ22 WINCHESTER 5.2 MB 3.5"	PDP,VAX	51216
RZ23	DEC RZ23 WINCHESTER 104 MB 3.5"	PDP,VAX	102432
RZ23L	DEC RZ23L WINCHESTER 121 MB 3.5"	PDP,VAX	118872
RZ24	DEC RZ24 WINCHESTER 209 MB 3.5"	PDP,VAX	204896
RZ24E	DEC RZ24 WINCHESTER 209 MB 3.5"	PDP,VAX	204827
RZ24L	DEC RZ24L WINCHESTER 245 MB 3.5"	PDP,VAX	239976
RZ25	DEC RZ25 WINCHESTER 425 MB 2.5"	PDP,VAX	416268
RZ25E	DEC RZ25 WINCHESTER 425 MB 3.5"	PDP,VAX	416052
RZ26	DEC RZ26 WINCHESTER 1.5 GB 3.5"	PDP,VAX	1025430
RZ26F	DEC RZ26F WINCHESTER 1.5 GB 3.5"	PDP,VAX	1025548
RZ26L	DEC RZ26L WINCHESTER 1.5 GB 3.5"	PDP,VAX	1025548
RZ27	DEC RZ27 WINCHESTER 1.6 GB 3.5"	PDP,VAX	1562704
RZ28	DEC RZ28 WINCHESTER 2.1 GB 3.5"	PDP,VAX	2055240
RZ28	DEC RZ28 WINCHESTER 2.1 GB 3.5"	PDP,VAX	2055240
RZ29	DEC RZ29 WINCHESTER 4.3 GB 3.5"	PDP,VAX	4190040
RZ35	DEC RZ35 WINCHESTER 825 MB 3.5"	PDP,VAX	832314
RZ55	DEC RZ55 WINCHESTER 332 MB 5.25"	PDP,VAX	330480
RZ56	DEC RZ56 WINCHESTER 665 MB 5.25"	PDP,VAX	660960
RZ57	DEC RZ57 WINCHESTER 1.0 GB 5.25"	PDP,VAX	987255
RZ58	DEC RZ58 WINCHESTER 1.38 GB 5.25"	PDP,VAX	1349587
RZ59	DEC RZ59 WINCHESTER	PDP,VAX	8877807

# MkDskCmd

## Table of Contents

- General Description
- Creating disk images
- Resizing disk images

## 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" from Administrator in the Start menu and switch to the CHARON installation folder then the "\Build\_XXX\xxx" child folder where the utility is located.

## Creating disk images

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


```
...> mkdskcmd -list
```

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

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

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

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

-  A file "rz22.avdisk" will also be created. This file helps CHARON 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

mkdisk for CHARON utility v. 1.17
Copyright: STROMASYS, 2020

Usage:
    mkdiskcmd [Options]

Options:
    -h, -help - display help screen

    -o, -output <file> - specify output file name

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

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

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

    -a, -avtable <file> - specify AVDISK table file

    -r, -resize <file> [<disk-name>]
        - resize the disk image
        <file> - file name of the disk image to be resized
        <disk-name> - name of the disk from the Disk table

        <file> will either have the specified number of blocks added to the
        end or be truncated at the new smaller size.

        To specify a custom disk size, use the following parameters:
        -resize <file> -blsize <value> -blcount <value>

    -s, -shrink - mandatory parameter when resizing to smaller disk

    -l, -list - to display AVDISK table

    -q, -quiet - run in quiet mode

Return value:
    0 - Success
    Non zero - Failure

Examples:
mkdiskcmd -help
mkdiskcmd -list
mkdiskcmd -avtable mkdisk.vtable -output rk07.vdisk -disk rk07
mkdiskcmd -output disks\custom.vdisk -blsize 512 -blcount 16384
mkdiskcmd -r 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.


## Resizing disk images

The "MkDskCmd" utility is able to resize (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.

### Notes:

- It is not possible to add more free space dynamically. The virtual machine must be stopped before performing this operation.
- Resizing a disk image requires the operating system running on the Charon virtual machine to be able to handle Dynamic Volume Expansion. Please refer to the documentation of your operating system version. If this is not supported, please create a new virtual disk then backup and restore the existing data.

 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 -resize <source disk file name> <source disk parameters> [-shrink] [-k]
```

where:

- <source disk file name> - the file name of the disk image to be resized
- <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 -resize "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 -resize <source disk file name> -blsize <number> -blcount <number>
```

### Example:

```
...> mkdskcmd -r "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 resized and its actual availability to CHARON. This delay can reach up to several minutes in the case of very large disk resizes. This is because the host operating system needs time for the actual allocation of the enlarged file on HDD.

# HASP View

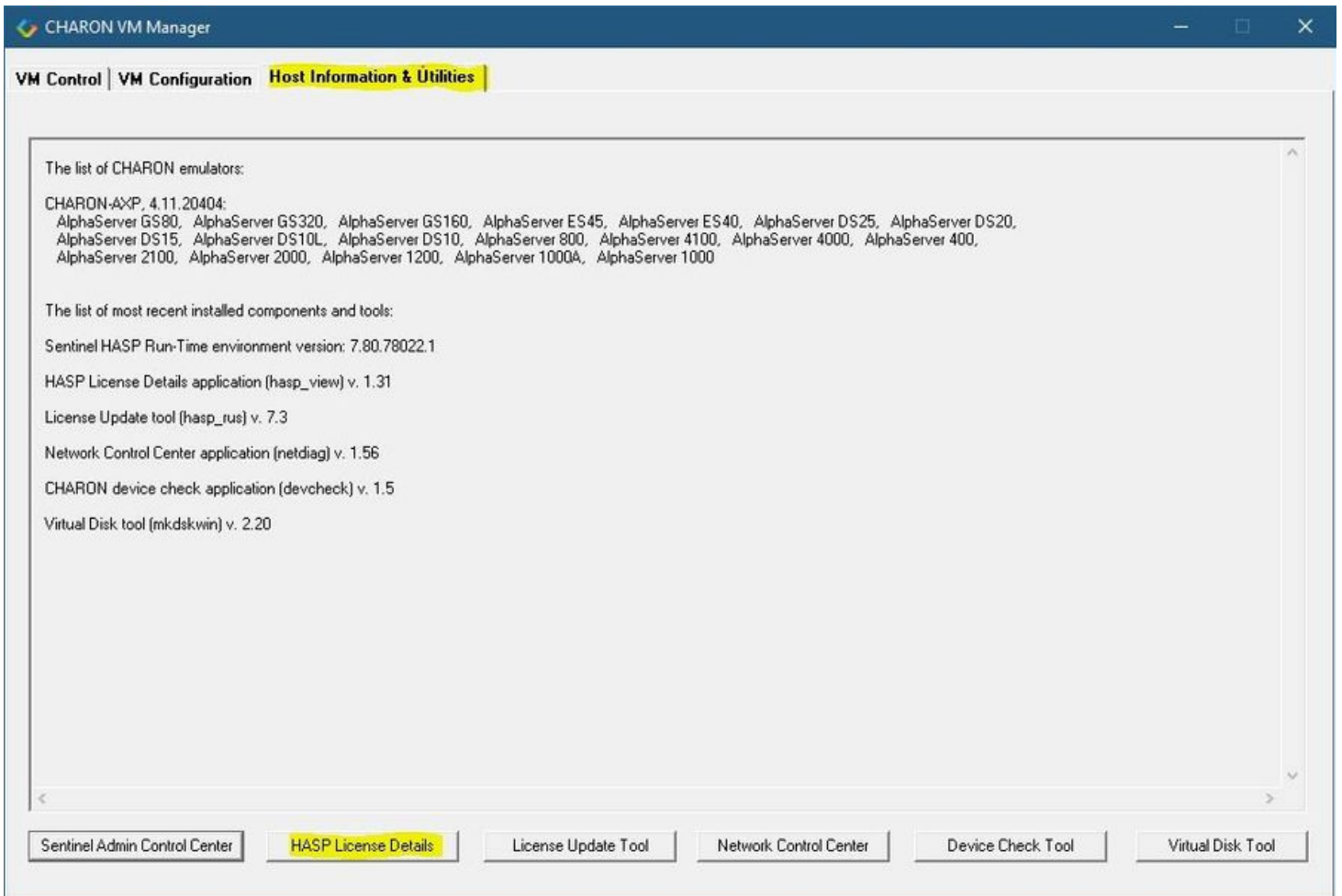
## Table of Contents

- General Description
- Getting CHARON licenses content

## General Description

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

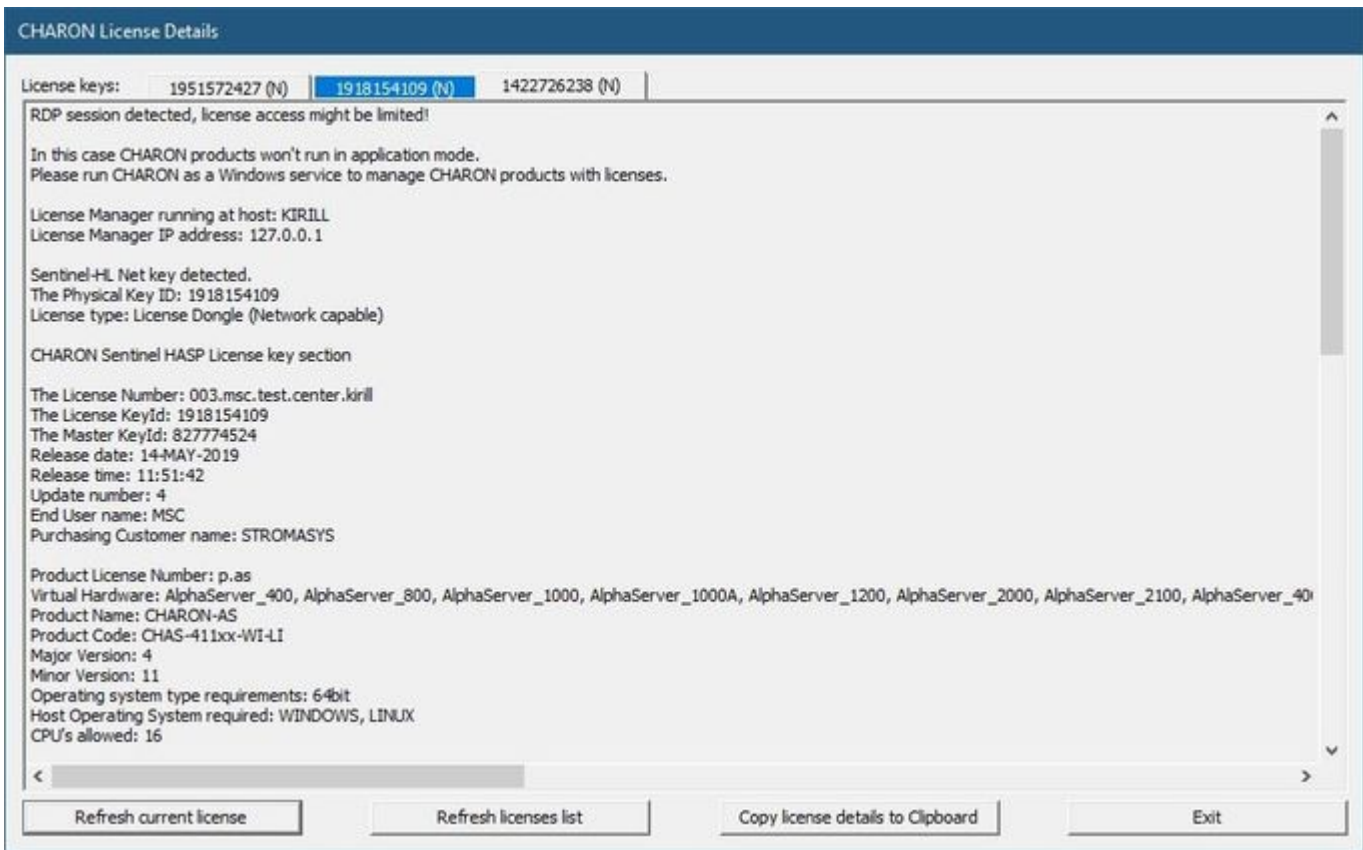
Open the CHARON Virtual Machine Manager, switch to the "Host Information & Utilities" tab and press the "HASP License Details" button:



## Getting CHARON licenses content

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

### Example:



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

Press the "Refresh key list" button to look for newly added and 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 if needed.

Press the "Exit" button to exit from the utility.

# License Update Service

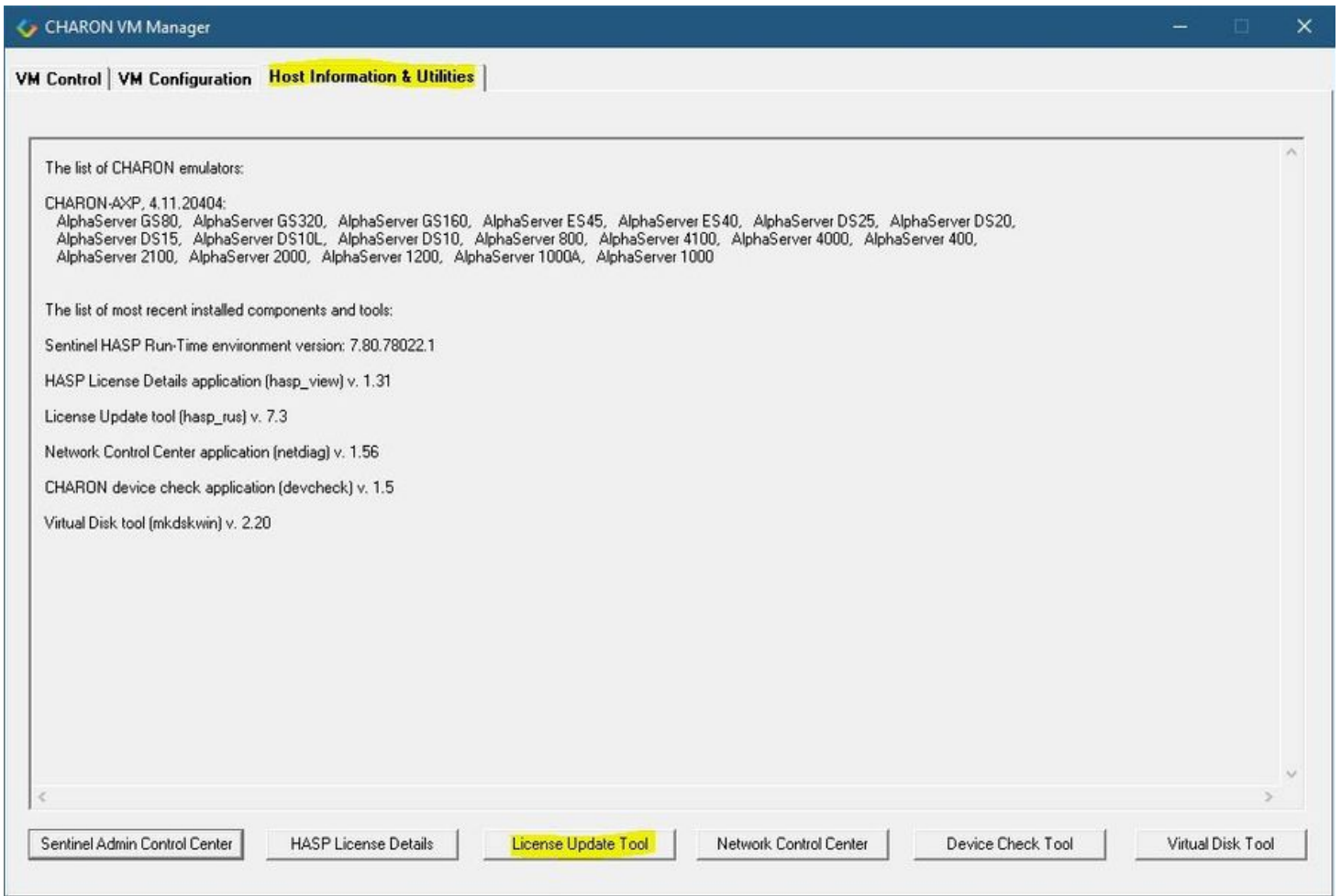
## Table of Contents

- General Description
- Collecting the host fingerprint and information on an existing license
- Installation and update of a CHARON license

## General Description

The "License Update Service" utility is used to manage CHARON licenses, collect the host system fingerprint.

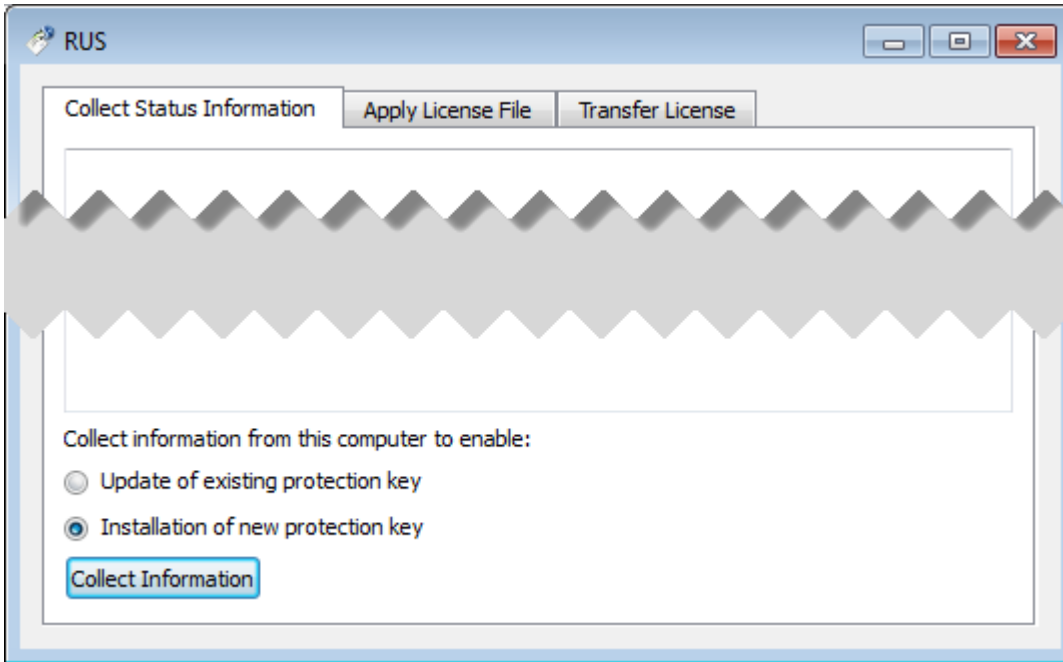
Open the CHARON Virtual Machine Manager, switch to the "Host Information & Utilities" tab and press the "License Update Tool" button:



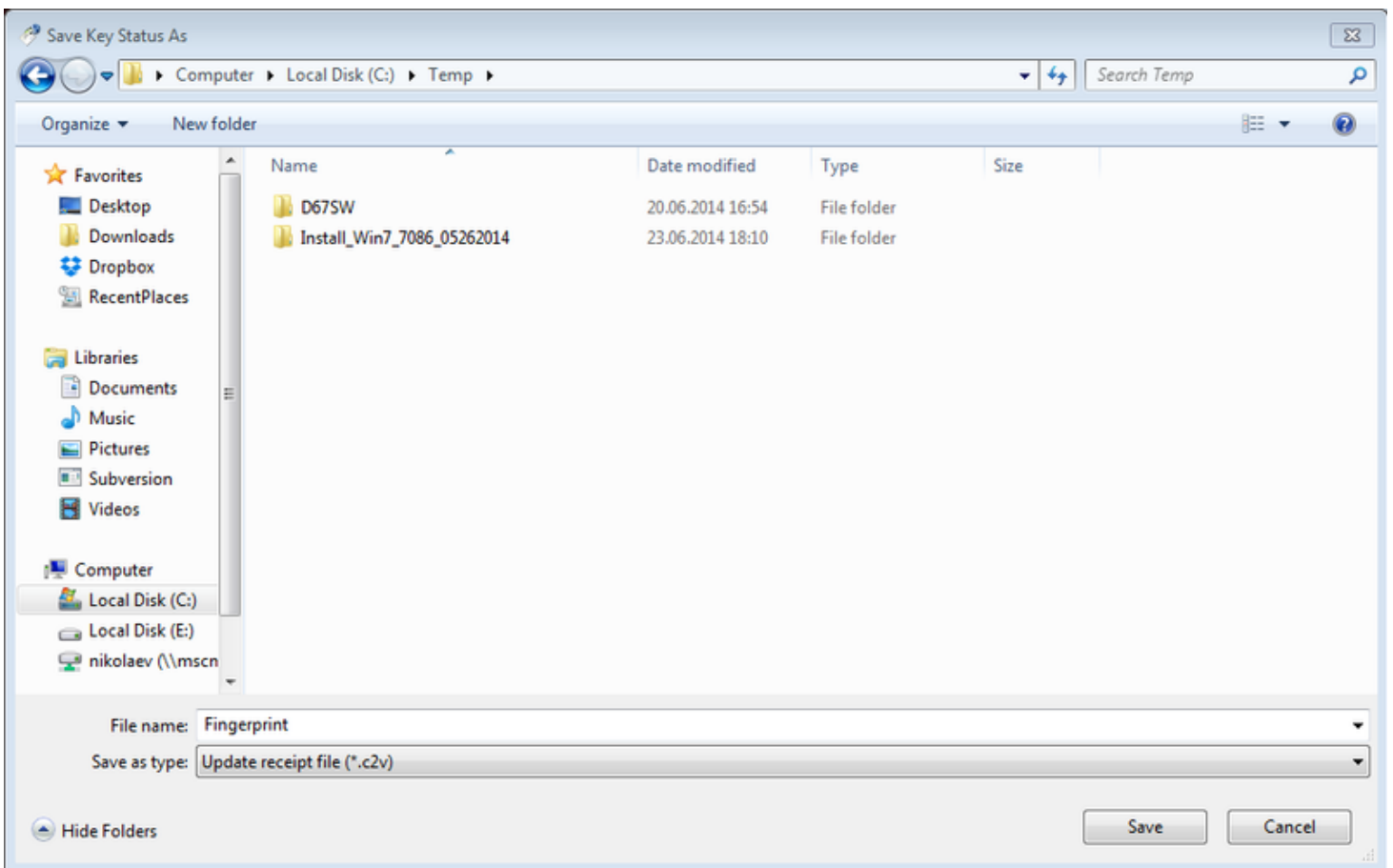
## Collecting the host fingerprint and information on an existing license

Open the "Collect Status Information" tab.

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



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

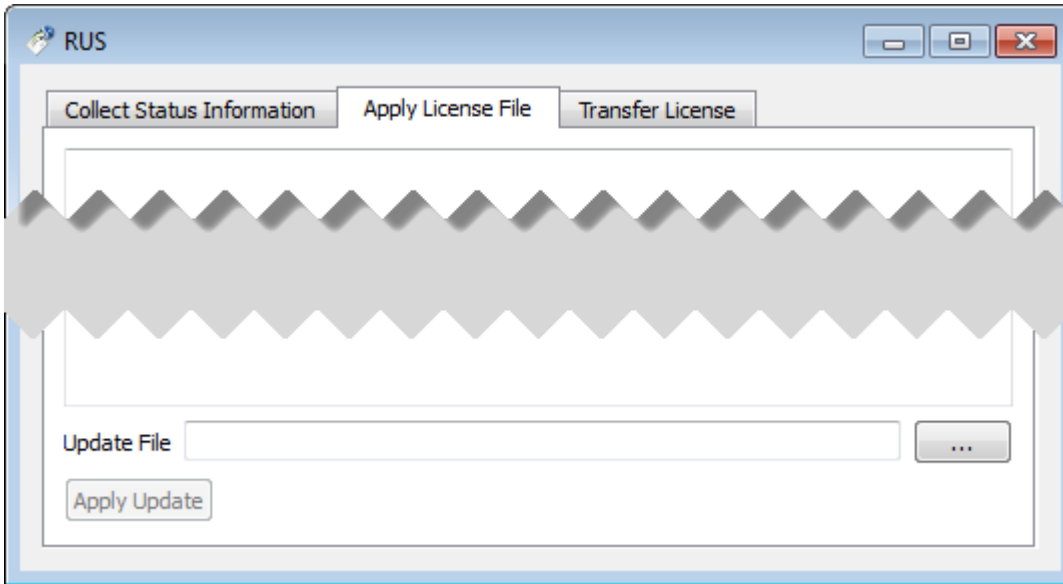


Press the "Save" button to create the "\*.c2v" file and send this file to STROMASYS to receive an update of the current license or a new license.

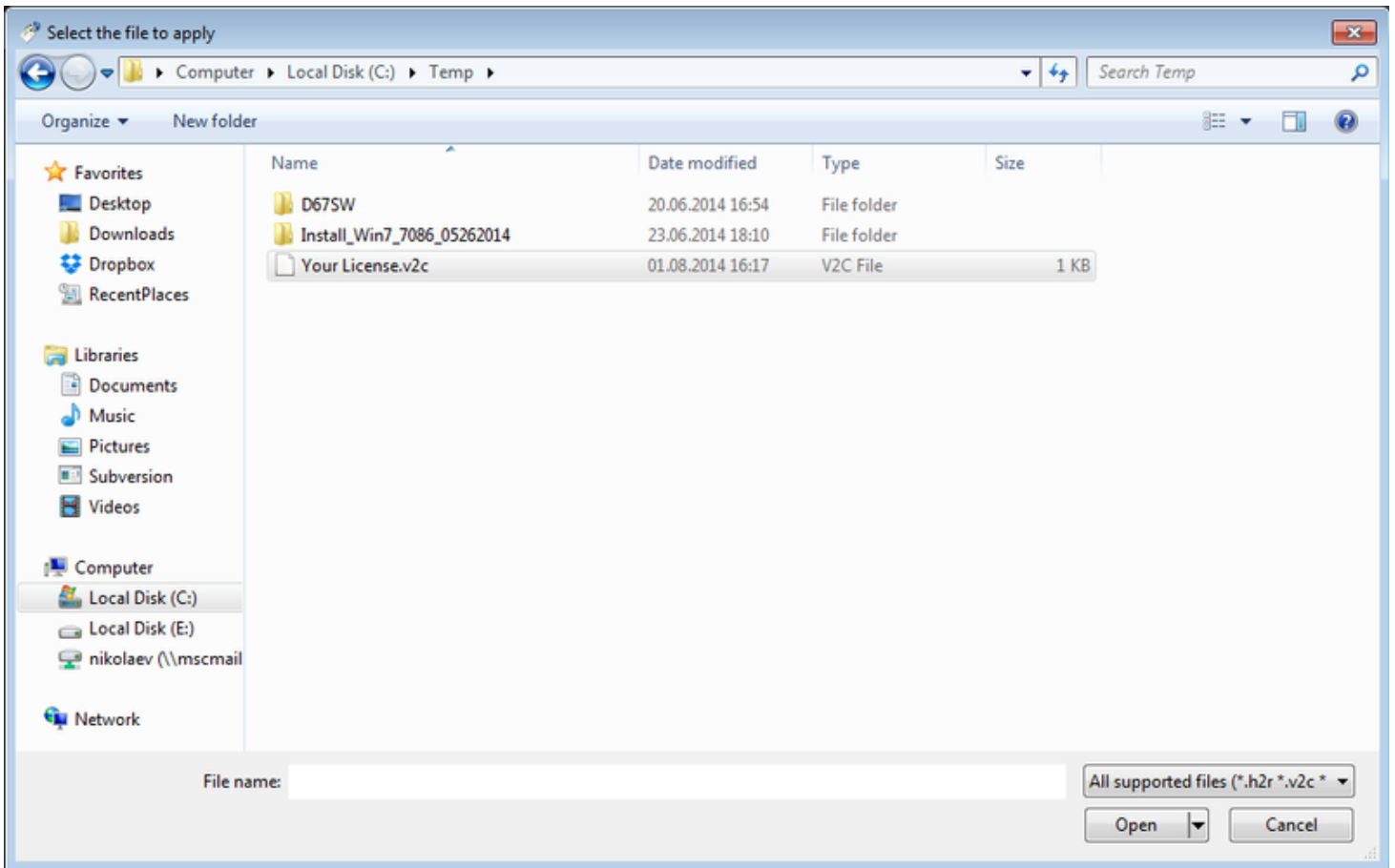


## Installation and update of a CHARON license

Open the "Apply License File" tab and press the "..." button:



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



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

**i** In case you receive 2 files to update the license, install the "\_fmt" (format) one first then the second one.

# License expiration check

## Introduction

Once setup, this utility will execute once a day at 09:00 AM and will send Windows Application Events 15 days before expiration. No other action will be taken unless you customize the `expiredlicense.bat` script. The utility can be found in the Charon installation folder under the "Redistributables\Stromasys" child folder.

**i** The license expiration check utility is not provided with the CHARON-VAX/PDP V4.12 kit. If you need it, please ask us for the download link.

The installation process is described below. It is recommended not to update the example files given with the kit as they could be overwritten by a patch installation or product upgrade. It is preferable to copy them in a dedicated folder.

**Version:** 1.18 - **Published:** 17 Jan 2023

## Table of contents

- [Description](#)
- [Step-by-step guide](#)
  - [Installation](#)
  - [Interactive check](#)
  - [Scheduled task creation](#)
    - [Scheduled task creation using the Graphical User Interface](#)
    - [Scheduled task creation using the Character User Interface](#)
    - [Configuration file example](#)
    - [Hints](#)
- [Sending alerts via email - example](#)
  - [Email account password encryption](#)
  - [Powershell script used to send emails](#)
  - [expiredlicense.bat file settings](#)
  - [Scheduled task settings update](#)
- [Related articles](#)

## Description

The goal of this Powershell script is to read the license content and send alerts 15 days (by default) before expiration. This document is related to the kit versions 1.9 to 1.17.

**This utility is replaced by [CHARON on Windows - Charon License Utility](#)** that provides a graphical interface providing the Windows server has PowerShell 5.1 and .NET Framework 4.6 minimum installed (4.8 recommended) installed.

### Alerts consist in:

- Adding a Windows Event entry in the Application log
- Executing a customized script named "`expiredlicense.bat`" that can execute actions at your convenience: send an email, send an event to a monitoring software, etc...

### Alert levels:

Days before expiration (date limited license)	Hours before expiration (time limited license)	Alert level
Between 7 and number of days defined (15 by default)	Between 73 and 96 hours	Informational
Between 3 and 7	Between 49 and 72 hours	Warning
Less than 3 and expired	Less than 49 and expired	Error

### Related products:



Charon-AXP and Charon-VAX/PDP V4.5 and above

### Operating systems:

Windows 7, Windows 8.1, Windows 10 Professional, Windows Server 2008 R2, Windows Server 2012 (R2), Windows Server 2016, Windows Server 2019

## Step-by-step guide

### Installation

- Create a folder dedicated to store the scripts, for example: "C:\Charon"
- Download the kit from our SFTP server (please ask us connection credentials if you have no access) and extract all files in this folder.
  -  The kit contains some files, their usage is described further in this document.
- Optionally create the "expiredlicense.bat" file in the same folder and customize it at your own (to send emails for example).
  -  If the script does not exist, a warning event will be posted but it will not prevent the script from running. In this case only Windows Application events will be sent.

Below is a simple example to append entries in a log file. You will find further in this article an example on [how to send an email using powershell](#).

```
echo %date% %time% %1 %2 >> C:\Charon\expiredlicense.log
```

#### Notes:

- "%1" corresponds to the severity level and can be "informational", "warning" or "error"
- "%2" corresponds to the alert message containing the license number, the product name and the expiration status


### Interactive check

- Open a command line window
- Move to the folder you created, example: C:\Charon
- Execute the powershell script. The following parameters can be passed to the script:
  - "-folder" is used to specify where the scripts (.ps1 and .bat) are stored. If not specified, default is "C:\Charon"
  - "-nbdays" is used to specify the number of days before expiration to start sending alerts, default is 15
  - "-nbhours" is used to specify the number of hours before expiration of a backup/spare license (time limited) to start sending alerts, default is 96. Must not be lower than 72.
  - "-usevanilla" is used in case the `hasp_view.exe` program is not able to provide the expiration date in the output file (bug found in builds 18304 and 18305 with `hasp_view.exe` version 1.26). In this case, the script will use the `hasp_view.exe` program located in the "xxxxx\_ORIGINALS\_xxx\_PATCH\_xxx" folder (example: B18302\_ORIGINALS\_AXP\_PATCH\_X64). The bug is solved with `hasp_view.exe` version 1.27 and later.
  - "-haspview <hasp\_view.exe location>" is used to specify the location of the `hasp_view.exe` file. If this parameter is not specified then the script searches for the location itself depending on Charon installation folder. It is used in case a license server performs the checks instead of the server hosting the Charon virtual machines. The "-usevanilla" parameter is ignored if this parameter is defined.
  - "-ini <file>" is used to specify a file where the `-folder`, `-nbdays`, `-nbhours`, `-usevanilla`, `-haspview` and `-tick` parameters can be stored. Priority goes to the parameters passed in the command line then those in the specified file. The file is used to facilitate updates in the parameters passed in the tasks scheduler (update the file instead of the task).
  - "-tick" parameter is used to execute external script (`expiredlicense.bat`) even if no alert is detected (counter alert for monitoring software). If used in the .ini file, "1", "y", "yes" or "true" value activates it, other values will invalidate (**Example:** tick=true)

To run PowerShell scripts (files that end with .ps1), you must first set the execution policy to Unrestricted (This operation has to be done once).

To do so, open a command line window (cmd.exe) as an Administrator and use the following command:

```
c:\Charon>powershell -command "Set-ExecutionPolicy Unrestricted"
```


 The ExecutionPolicy can also be set to "RemoteSigned". In this case the .ps1 script files will have to be unblocked as described below.

If you are still prompted to allow for execution of the script, please run the following command to unblock the downloaded `charon_expchk.ps1` file:

```
c:\Charon>powershell -command "Unblock-File -path c:\charon\charon_expchk.ps1"
```

#### Example:

```
c:\Charon>powershell -file charon_expchk.ps1 -folder c:\charon -nbdays 20
```

 If the license is not readable (dongle not connected for example), the script will end with an "Unable to generate the license view from `hasp_view.exe`!" error, an entry will be added to the Windows Event Application log and the `expiredlicense.bat` script will be executed.


If no script error is detected, it is now possible to create a new scheduled task.

## Scheduled task creation

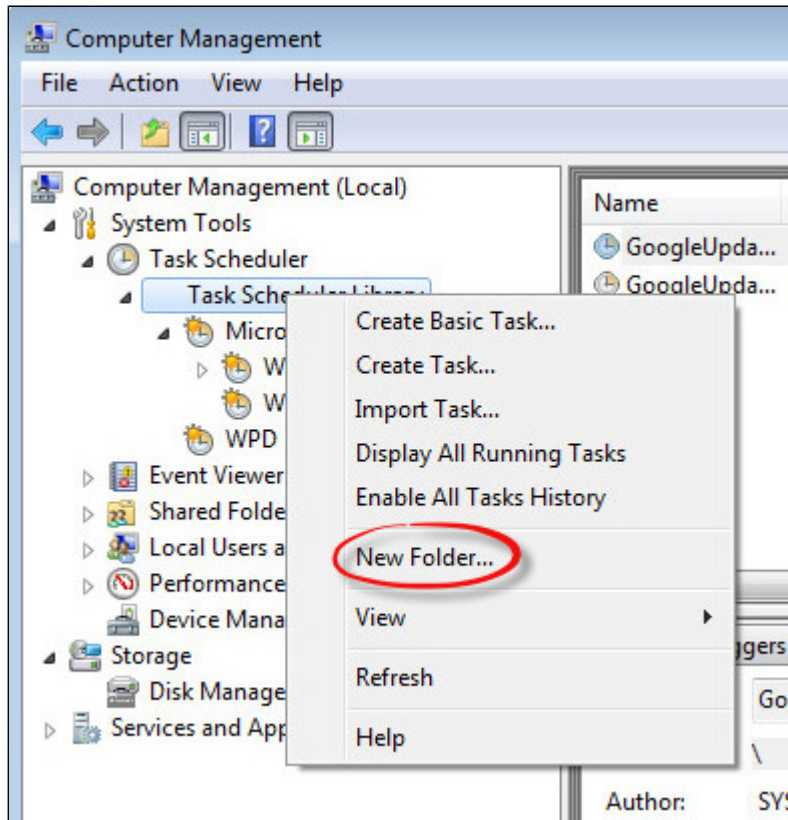
### Scheduled task creation using the Graphical User Interface

To set this,


- either open the Administrative tools under Control Panel and select "**Task Scheduler**" or
- execute `taskschd.msc` from the Windows run command (press Windows key + R)

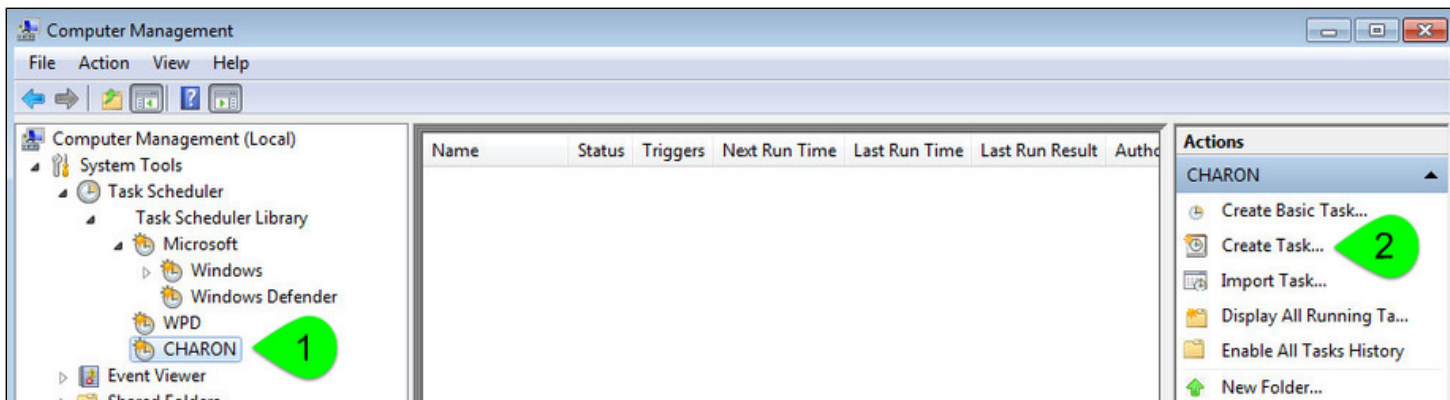
 Examples are given based on a Windows 2008 R2 server

Right click on **Task Scheduler Library**, select New folder then enter **CHARON**.

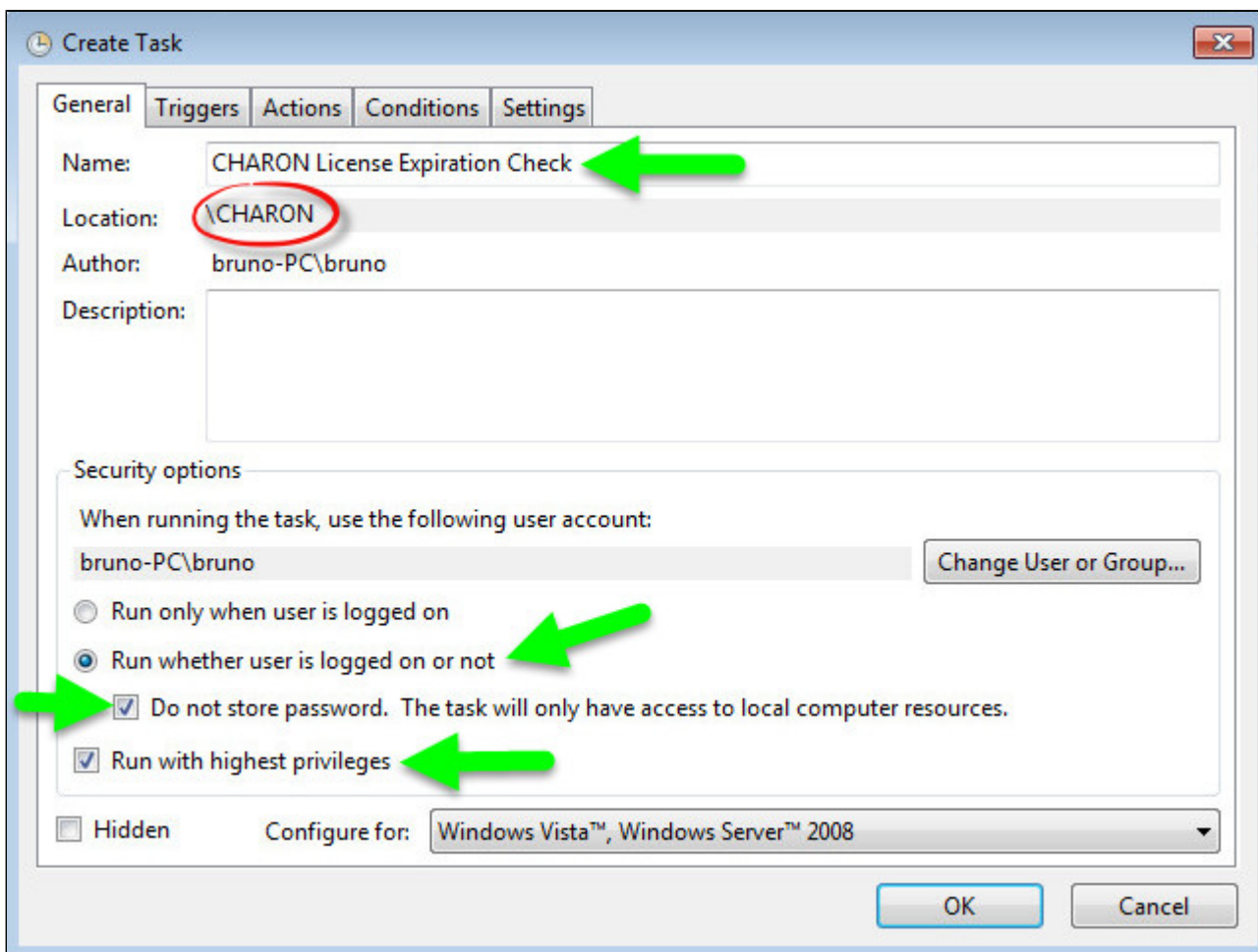


Select the newly created folder (1) and on the right pane, select "**Create Task...**" (2).

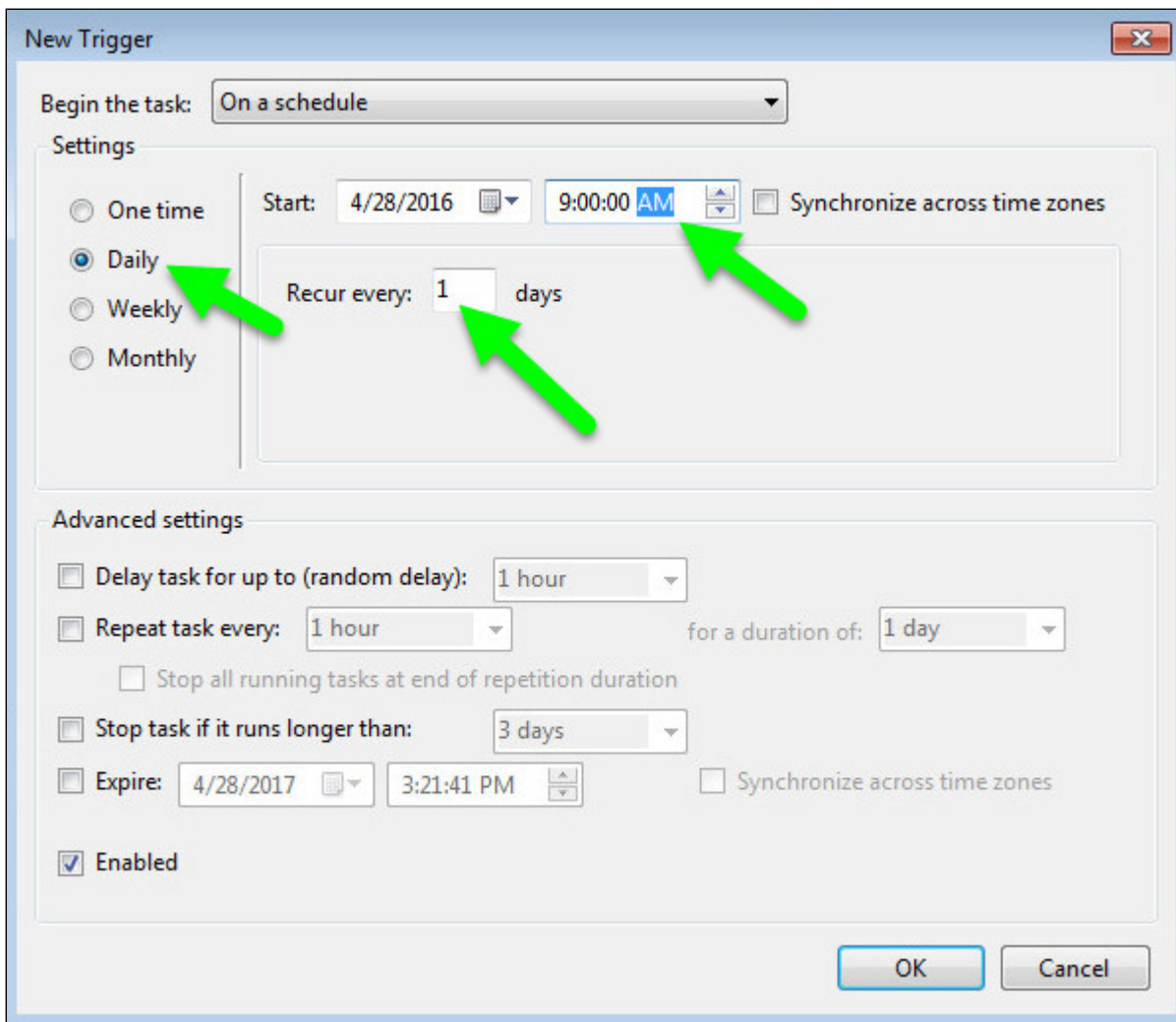
 Do not use "Create Basic Task..." some of the options required aren't available in that wizard.



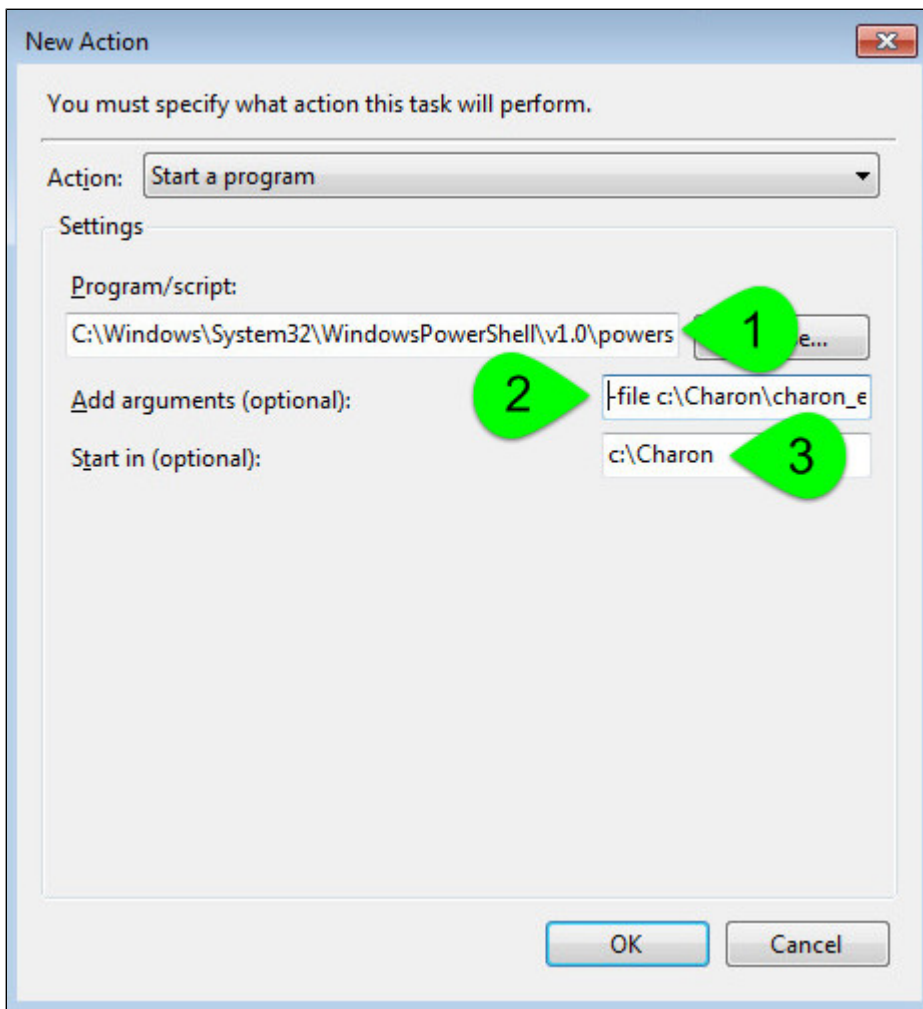
Fill the **General** tab as below:



Select the **Triggers** tab and set the check interval, for example here: everyday at 09:00 AM



Select the **Actions** tab and configure powershell to be executed with the following parameters:



(1) : Powershell location here is "C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe"

(2) : Define the powershell arguments to be added: `-file c:\Charon\charon_expchk.ps1 -folder c:\Charon` (the folder parameter is optional here as it is the same as the default folder)

(3) : Set the folder to be the one where the scripts reside

**i** Once the task is created, it is recommended to right click on the task then "Run". To verify the script ran correctly, please check the existence of the "charon\_expchk.txt" file in the installation folder (it should contain the license view, if the license is present) and check for events in the Windows Event Application log if any.

## Scheduled task creation using the Character User Interface

Open a command line window (cmd.exe) and execute the following command:

```
c:\Charon>schtasks /create /SC DAILY /ST 09:00 /NP /RL HIGHEST /TN "CHARON\CHARON License Expiration Check" ^
More? /TR "C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe -file c:\Charon\charon_expchk.ps1 -folder c:\Charon"
```

**💡** Update the schedule type (/SC) and time (/ST) to your desired settings

Start the task as shown below:

```
c:\Charon>schtasks /run /TN "CHARON\CHARON License Expiration Check"
```

To verify the script ran correctly, please check the existence of the "charon\_expchk.txt" file in the installation folder (it should contain the license view, if the license is present) and check for events in the Windows Event Application log if any.

## Configuration file example



In the following example, the parameters/values are stored in the `c:\charon\charon_expchk.ini` file.

**Example:**

```
#Folder where the scripts are located
folder=C:\Charon

# Number of days before expiration alert
nbdays=21

# Location of the hasp_view.exe file if Charon is not installed
haspview=C:\Charon

# Send an alert even if no license is about to or has expired
tick=true
```


To execute the script with the .ini file, use the following command:

```
c:\Charon>powershell -file c:\charon\charon_expchk.ps1 -ini c:\charon\charon_expchk.ini
```

To create the scheduled task, open a command line window (cmd.exe) and execute the following command:

```
c:\Charon>schtasks /create /SC DAILY /ST 09:00 /NP /RL HIGHEST /TN "CHARON\CHARON License Expiration Check" ^
More? /TR "C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe -file c:\Charon\charon_expchk.ps1 -ini c:
\Charon\charon_expchk.ini"
```

## Hints


 To verify the script is correctly working and to force the creation of a Windows event, and eventually the execution of the "expiredlicense.bat" file, you can:

- either play with the removal of the dongle (if there is, that is no Software License)
- or with the number of days (if the license is not unlimited)
- or you can specify a folder that does not exist.

Windows Event Application log example - Information alert:

Event 314, CHARON

General Details

CHARON License Expiration Check: CHARON-VAX-66x: all models license is approaching termination (13 days remaining).  
License number: 1000.800 

Log Name: Application  
 Source: CHARON  
 Event ID: 314  
 Level: **Information**  
 User: N/A  
 OpCode:


Logged: 6/3/2016 9:01:42 AM  
 Task Category: (1)  
 Keywords: Classic  
 Computer: bruno-PC

More Information: [Event Log Online Help](#)

Windows Event Application log example - License dongle not connected:

Event 314, CHARON

General Details

CHARON License Expiration Check: Unable to generate the license view from hasp\_view.exe! 

Log Name: Application  
 Source: CHARON  
 Event ID: 314  
 Level: **Error**  
 User: N/A  
 OpCode:


Logged: 4/28/2016 3:46:17 PM  
 Task Category: (1)  
 Keywords: Classic  
 Computer: bruno-PC

More Information: [Event Log Online Help](#)

## Sending alerts via email - example


Using Powershell we can send alerts via email. You will find below an example with configuration steps.


### Configuration details:


- Windows 2008 R2 server
- Powershell V4.0 installed.
  -  **This version is required** to have the "-port" option available from the "send-mailmessage" powershell command.
  - To determine which version of Powershell is installed and upgrade if necessary, see [Powershell version, upgrade, enabling scripts execution, tips and tricks](#)
- Email account hosted on Office365

## Email account password encryption

Due to the email account settings, to be able to send an email we need to generate a CliXML file that will store the encrypted password as described below.


 This file will only be valid for the current user and machine. The following operations must be performed using the same user as the author of the scheduled task.

 The "<monitoring-account>@<somewhere>" value must be adapted to your configuration

 When you will execute the "get-credential" command, you will be prompted to store the password of the corresponding email account, that is the "sender".


```
C:\Charon>powershell
Windows PowerShell
Copyright (C) 2014 Microsoft Corporation. All rights reserved.

PS C:\Charon> $From="<monitoring-account>@<somewhere>"
PS C:\Charon> $Creds=(get-credential -credential $From)
PS C:\Charon> $Creds | Export-CliXML C:\Charon\creds.clixml
PS C:\Charon> exit
C:\Charon>exit
```

 If the sender's email address or its password has to be changed, the .clixml file will have to be recreated.

## Powershell script used to send emails

Create a Powershell script named "sendmail.ps1" in the "c:\Charon" folder for example.


 The sender (\$From) and recipient (\$To) values must be adapted to your configuration

```
Param(
    [string]$lvl = "Information",
    [string]$msg = "no message"
)

$Hostname = (Get-Item env:\COMPUTERNAME).value
$Creds = Import-clipxml C:\Charon\creds.clipxml

#--- Customize the values below: "From" and "To" -----
$From = "<monitoring-account>@<somewhere>"
$To = "<someone>@<somewhere>"
#-----

send-mailmessage -to $To -from $From -subject "CHARON License Expiration Check for $Hostname ($lvl)" -body "$msg" -credential
$Creds -smtpserver smtp.office365.com -usessl -port 587 -delivery none
```

 If you have to specify multiple recipients, an array has to be used as described below.

**Example1:**


```
$To = @( "James T. Kirk <kirk@uss-enterprise.fed>", "Spock <spock@uss-enterprise.fed>" )
```

**Example2:**

```
$To = @( "kirk@uss-enterprise.fed", "spock@uss-enterprise.fed", "uhura@uss-enterprise.fed" )
```

 The example shown above and another one given with embedded html code and colored email body are part of the kit downloaded previously.

Office365 email account is given here as an example but you can use "gmail" also and create a free dedicated account to send alerts. In this case the last line of the script(s) above has to be updated: the "-smtpserver" parameter has its "smtp.office365.com" value to be replaced by "smtp.gmail.com" (no other value to update)


 Google may block sign-in attempts when using powershell and send-mailmessage. In this case the sender will receive a "Sign in attempt prevented" alert email. To allow emails to be sent:

1. Create a dedicated gmail account
2. Allow less secure apps to access your account. See this article: <https://support.google.com/accounts/answer/6010255?hl=en>

## expiredlicense.bat file settings

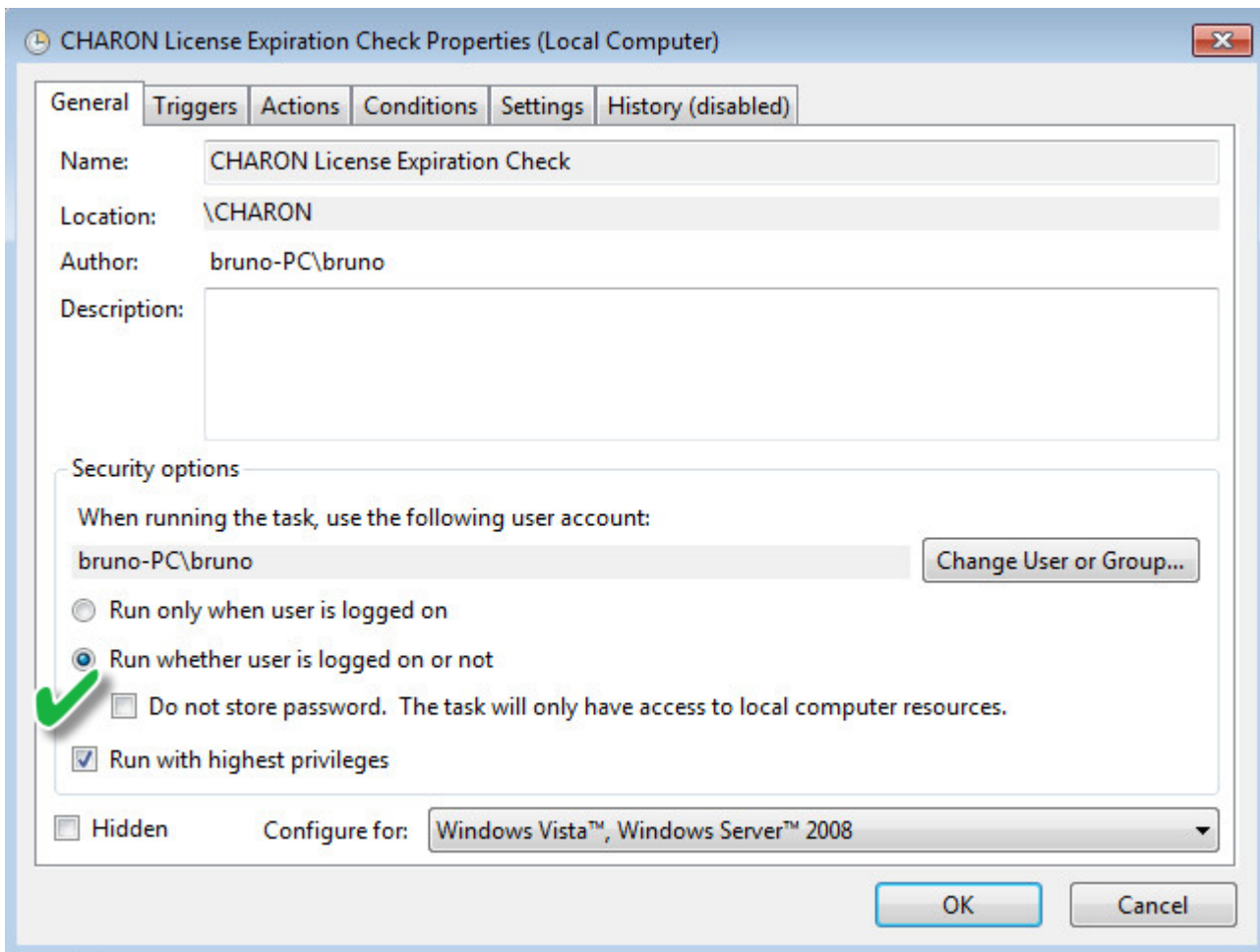
Add the following line in the "expiredlicense.bat" file:

```
powershell -NonInteractive -file c:\charon\sendmail.ps1 -lvl %1 -msg %2
```

 This .bat file can be found in the kit downloaded previously.

## Scheduled task settings update

This time, the **General** tab of the Scheduled task has to be updated in order to store the password and to be able to send the email:



You will be prompted to store the password of the author.

This can also be done using the command line:

```
c:\Charon>schtasks /change /TN "CHARON\CHARON License Expiration Check" /rp <password>
```

## Related articles

- [CHARON on Windows - Automated License Expiration Check](#)
- [How-to install Charon license expiration check on a Linux server](#)
- [CHARON on Windows - Charon License Utility](#)
- [CHARON on Windows - Charon License Utility - Release Notes](#)
- [How to read a HASP key over RDP on Windows](#)

# mtd

## Table of Contents

- [General Description](#)
- [Usage](#)
- [Tape container formats transfer](#)

## 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.
- Convert tape images formats.
- Test a tape container *integrity*.

To start the utility open "cmd.exe" in the Start menu and switch to the CHARON installation folder then the "\Build\_XXX\XXX" child folder where the utility is located.

## Usage

The following are examples of the usage syntax:

Dump tape content to file:

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

Restore dump to tape:

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

Convert formats:

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

Examine tape dump and check its integrity:

```
...> mtd [options] <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")

Parameter(cont.)	Description
-s	Write tape image in Charon-AXP/SMA (named SMA further) format
-g	Gather statistics and print upon completion
-a	Do not print logo

**Example 1:**

```
...> mtd -l tape1.txt -r 10 \\.\Tape0 "C:\Charon\Tapes\tape1.vtape"
```

**Example 2:**

```
...> mtd "C:\Charon\Tapes\tape1.vtape" \\.\Tape0
```

## Tape container formats transfer

Use the following syntax to transfer the CHARON-AXP/SMA tape container format to the CHARON-AXP/VAX/PDP one:

```
...> mtd <SMA tape container name> <AXP/VAX/PDP tape container name>
```

**Example:**

```
...> mtd C:\charon\tapes\sma_tape.vtape C:\charon\tapes\axp_tape.vtape
```

Use the following syntax to transfer the CHARON-AXP/VAX/PDP tape container format to the CHARON-AXP/SMA one:

```
...> mtd -s <AXP/VAX/PDP tape container name> <SMA tape container name>
```

**Example:**

```
...> mtd -s C:\charon\tapes\axp_tape.vtape C:\charon\tapes\sma_tape.vtape
```

# 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](#)
- [Log Monitor and Dispatcher best practice](#)


## General Description

The Log Monitor & Dispatcher is a special program which monitors a guest LOG file produced by CHARON and executes a custom 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. It detects the following messages and executes the associated scripts (created manually):

Error code	Description	Script executed
00000424	Detected removal of a license.	nolicense.bat
0000040B	License has changed. License detected and online.	license_changed.bat

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.

 The tool requires the specific files "nolicense.bat" and "license\_changed.bat" containing specific instructions to be taken in situation of license absence or change. It is recommended you create these files 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.  
Start the Log Monitor & Dispatcher using the following command line as an example (for Windows x64; for Windows x86 use the following path: "C:\Program Files\CHARON\Utilities\_X.X.XXXXXX\x86"):
 

```


```

For single log file:

```
...> "C:\Program Files\CHARON\Build_XXX\x86\logmond" my_charon.log
```

For rotating log file:

```
...> "C:\Program Files\CHARON\Build_XXX\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\Build_XXX\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.

## 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\Build_XXX\x86\logmond" -r
...> "C:\Program Files\CHARON\Build_XXX\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.

- ⚠ There will have one logmond father process that will scan the virtual machines services and one logmond process per log monitored.
- The EmulatorLogMonitor service running the logmond processes will discover already installed CHARON virtual machines services. If a new virtual machine service is added, it's log file will be automatically discovered and monitored.
- The "nolicense.bat" and "license\_changed.bat" files have to be placed in the virtual machine "Home directory". This directory corresponds to the configuration file folder when the service is installed, however it will not change if the configuration file is relocated and the service is updated. To find this "Home directory", open the "CHARON Service Manager" utility, right click on the corresponding service and select "View configuration"

## 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\Build_XXX\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. But these cases can still be monitored with "logmond" invoked from command line prompt with name of the log file as an argument (see above).

## 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. Stop the Log Monitor & Dispatcher service using the following command line as an example

```
...> "C:\Program Files\CHARON\Build_XXX\x86\logmond" -d
```

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

## 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. Uninstall the Log Monitor & Dispatcher service using the following command line as an example

```
...> "C:\Program Files\CHARON\Build_XXX\x86\logmond" -r
```

## Log Monitor and Dispatcher best practice

Refer to [this article](#) in the Appendixes for proper configuration of the "nolicense.bat" file to establish needed actions on the dongle removal.

# HOSTPrint

## Table of Contents

- [General Description](#)
- [Usage](#)

## General Description

The HOSTPrint utility is used to print CHARON output to Windows printers. It receives data from an emulated LPV11 printer 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.

Operation modes supported by the HOSTprint utility:

Mode	Description
Line-printer (Digital LA75) emulation	In this mode the utility emulates ESC-sequences of the selected line printer (see the LA75 manual for details). Once a page is completely filled with data it is sent to the printer output (to emulate line-printer functionality).  To use this mode, properly configure the OpenVMS printer queue and use the OpenVMS "PRINT" command.
Working in an old version of the utility's compatible printing mode	Access the popup menu and set the Flush buffer delay time. Recommended value is 5 seconds.  The VAX/VMS command:  \$ COPY MY_PRINTED_FILE LPA0:


Both modes support the print preview of the last page in the printing buffer.

## Usage

HOSTPrint calling is specified in the CHARON configuration file. The call implements 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.  <b>Example 1:</b> -printer=[\\.\Microsoft Office Document Image Writer]  <b>Example 2:</b> -printer=[\\print_server\MSCCLPS]
-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 from a CHARON configuration file for the HOSTPrint usage (for 2 LPV11 printers):**

```
load chapi lpv1
set lpv1 dll=lpv11.dll
set lpv1 port=10015 application="hostprint.exe -port=10015 -printer=[\\print_server\MSCPS2] -font=\Courier New\ -fontsize=10"
load chapi lpv2 address=017764004
set lpv2 dll=lpv11.dll
set lpv2 port=10016 application="hostprint.exe -port=10016 -printer=[\\print_server\MSCLPS] -font=\Courier New\ -fontsize=12"
```

After initialization, HOSTprint creates an icon in the Windows tray. The icon can have two colors:

Color	Description
GREEN	IDLE (or Ready) state
YELLOW	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. Then preview the last page content, change the utility mode via the flush buffer delay or change the default font setting.

# VT-30TV

## Table of Contents

- [General Description](#)
- [Usage](#)

## General Description

The VT-30TV is a VT30-TV virtual color television monitor emulation. This emulation is a separate executable file "vt30-tv.exe" usually located in "C:\Program Files\CHARON\Build\_XXX\x86".

The VT30-TV virtual color television monitor should be connected to a VAX/PDP11 via the VT30-H emulation. It must be started with a desktop color quality equal to 8-bit, 16-bit or 32-bit.

## Usage

The VT30-TV can be started by means of user defined configuration options as follows:

```
> vt30-tv.exe --address=<host(vt30-h) IP address> --port=<host(vt30-h) IP port> --log=<log file name> --trace_level=<level>
```

### Options:

Parameter	Description
--address	IP address for connection to VT30-H.
--port	IP port for connection to VT30-H.
--log	Log file name
--trace_level	Tracing level

### Example:

```
> vt30tv.exe --address=127.0.0.1 --port=10020 --log="chapi_vt30tv.log" --trace_level=0
```

# CHARON Guest Utilities for OpenVMS

## Table of Contents

- General Description
- Installation
- Performance optimization
- Adjusting emulator's speed
- Emulator's shutdown control
- Virtual tapes management
- Defining keys
- Displaying version

## General Description

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

This set of utilities is located in the "charoncp013.vdisk" disk file in the "C:\Program Files\CHARON\Virtual Disk Images\Tape utilities package".

**Supported OpenVMS versions:** OpenVMS 6.1 and above.

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

## Installation

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

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

Issue the following commands to install the package:

```
$ @SYS$UPDATE:VMSINSTAL
* Are you satisfied with the backup of your system disk [YES]? YES
* Where will the distribution volumes be mounted: <device name>:[000000.CHARONCP013.KIT]

Enter the products to be processed from the first distribution volume set.
* Products: CHARONCP013
```

Select all the components included to the package:

```
Component Selection

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

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

4. Exit

* Your choice [4]: 4
```

Proceed with installation using all the default options.

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

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

After that the package will be loaded automatically on startup.

## Performance optimization

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

To load the OpenVMS idle loop detection software, use:

```
$ CHARONCP SET IDLE /ENABLE
```

This allows CHARON to detect when the emulated CPU(s) are idle and use the host power saving instructions to reduce power usage.

To unload the OpenVMS idle loop detection software, use:

```
$ CHARONCP SET IDLE /DISABLE
```

## Adjusting emulator's speed

CHARON speed can be adjusted using these commands.

To increase the emulator speed by the specified number of steps, use:

```
$ CHARONCP SET SPEED /UP=<number of steps>
```

To decrease the emulator speed by the specified number of steps, use:

```
$ CHARONCP SET SPEED /DOWN=<number of steps>
```

To remove all speed stepping restraints, use:

```
$ CHARONCP SET SPEED /RESET
```

## Emulator's shutdown control

It is possible to schedule CHARON to shutdown directly from OpenVMS.

To specify the number of seconds before the CHARON emulator is to shut down, use:

```
$ CHARONCP SET SHUTDOWN /IN=<seconds, 60 is default>
```

The value can be between 0 and 65535 (approx. 18 hours).

To clear a scheduled shutdown, use:

```
$ CHARONCP SET SHUTDOWN /RESET
```

## Virtual tapes management

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

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



- It is mandatory to set the "removable" parameter to "true"
- The container name must be ".vtape", no name and path must be specified, only extension.

### Example:

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

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

To create the specified host-file (if it does not already exist) and attach it to the specified virtual tape device, use:

```
$ CHARONCP SET MAGTAPE <device> /LOAD="<filename>.vtape"
```



The container name specified in the /LOAD parameter must not be more than 255 characters

### Example:

```
$ CHARONCP SET MAGTAPE MKA600: /LOAD="E:\Charon\Tapes\backup_01.vtape"
```

To detach any file currently attached to the specified virtual tape device, use:

```
$ CHARONCP SET MAGTAPE <device> /UNLOAD
```

### Example:

```
$ CHARONCP SET MAGTAPE MKA600: /UNLOAD
```



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

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

## Defining keys

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

To define an equivalence string and a set of attributes with a key on the terminal keyboard, use:

```
$ CHARONCP
CHARONCP> DEFINE /KEY <key-name> <equivalence-string>
```

You can have a set of keys defined automatically for use with the CHARONCP utility by placing DEFINE/KEY commands in the file `SYSS$LOGIN:CHARONCP_KEYDEFS.INI`

### Example:

```
$ CHARONCP
CHARONCP> DEFINE /KEY F1 "SET MAGTAPE MKA600: /UNLOAD"
```

To display key definitions created with the DEFINE/KEY command, use:

```
$ CHARONCP
CHARONCP> SHOW KEY <key-name>
```

Refer to the DCL help entry for SHOW KEY for further information.

### Example:

```
$ CHARONCP
CHARONCP> SHOW KEY F1
DEFAULT key state definitions:
F1 = "set magtape mka600: /unload"
CHARONCP>
```

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

## Displaying version

To display the CHARONCP package version number and architecture, use:

```
$ CHARONCP SHOW VERSION
```

This can be useful for customers reporting issues with the CHARONCP software.

**Example:**

```
$ CHARONCP SHOW VERSION  
CHARONCP version id is: V1.3
```

# Charon Report Utility

## Description

The Charon Report Utility creates an HTML report file with useful information for the Stromasys support team, and a zip file that contains this HTML file and the configuration and log files of the virtual machines defined on the Windows server.

## Installation

The Charon Report Utility is provided as a package in the Charon product kit and as an executable in case installing a package requires authorizations. It is not installed by default. The package is located in the "CharonReportUtility" folder of the kit. It is also possible that Stromasys will provide a download link to get the latest version from our file server.


Install the package (.msi file), this will create a shortcut on the desktop and an entry in the Windows menu.

## Prerequisites

- PowerShell 4.0, PowerShell 5.1 recommended.
  - Powershell 4.0 can work but with limitations, some functions will not be operational
  - If you installed PowerShell 7, please ensure PowerShell 5.1 is installed (not 4.0)
  - To upgrade, please see: <https://docs.microsoft.com/en-us/powershell/scripting/windows-powershell/install/installing-windows-powershell>
- .NET Framework 4.6 minimum, 4.8 recommended.
  - Installed by default on Windows 10, 11, 2016, 2019 and 2022.
  - For older versions of Windows, this has to be installed manually. For more, see: <https://docs.microsoft.com/en-us/dotnet/framework/migration-guide/versions-and-dependencies>

## Usage

Open the executable file and click the "Generate report" button (the operation can take some time). Once completed, the resulting .zip file has to be sent to our support team.

 The utility must be executed "as administrator"

## Example

The screenshot shows the Charon Report Utility V1.36 application window. The title bar reads "Charon Report Utility V1.36". The interface includes a top navigation bar with icons for Home, File Explorer, Mail, and Help, and the Stromasys logo. On the left side, there are three buttons: "Generate report" (with a refresh icon), "Open HTML report" (with a document icon), and "Open ZIP report" (with a folder icon). At the bottom left, there is an "Exit" button with a right-pointing arrow icon. The main content area displays system information under the heading "General settings" and "Charon-AXP/VAX/PDP information".

**General settings**

- Utility version : V1.36 - 02-Apr-22
- Mode : Administrator (Console)
- Computer name : WIN2019BM
- Server uptime : 0d 00:01:30
- Username : Administrator
- Windows Version : Microsoft Windows Server 2019 Standard  
No service pack
- Windows Release Id : 1809
- Secure Boot : Enabled

**Charon-AXP/VAX/PDP information**

- Installation folder : C:\Program Files\CHARON
- Installed versions : CHARON-AXP, 4.11.20410  
CHARON-VAX, 4.11.20410  
CHARON-AXP, 4.11.20414  
CHARON-VAX, 4.11.20414

Click the 'Generate Report' button to create the zipped file and its HTML report file.

# CHARON-VAX and CHARON-PDP for Windows configuration details

## Introduction


This chapter describes, in detail, the configuration parameters of the devices emulated by CHARON-VAX / CHARON-PDP for Windows, with corresponding examples and parameters.

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

### Example:

```
load RQDX3/RQDX3 DUA
set DUA container[0]="C:\Charon\Disks\BootDisk.vdisk"
```

In this example, an instance of an RQDX3 controller is loaded with the name "DUA". 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-VAX / CHARON-PDP 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-VAX / CHARON-PDP configuration

- [General Settings](#)
- [Core Devices](#)
- [Serial lines](#)
- [Remote Management Console \(RMC\)](#)
- [Disks and tapes](#)
  - [MSCP and TMSCP Controllers](#)
  - [SCSI Controllers](#)
  - [DSSI Subsystem](#)
  - [CI Subsystem](#)
- [Networking](#)
- [Specific Controllers](#)
- [Sample configuration files](#)
  - [PDP-11/93 configuration file](#)
  - [VAX 4000 Model 108 configuration file](#)
  - [VAX 6310 configuration file](#)
  - [VAX 6610 configuration file](#)
  - ["configuration\\_name.icfg" configuration file](#)

# General Settings

## Table of Contents

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

## Session

General settings that control the execution of Charon-VAX / Charon-PDP belong to an object called the "session". It is a preloaded object; therefore, only "set" commands are necessary.

### 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, case sensitive
<b>Value</b>	<p>Virtual VAX/PDP11 system hardware model to be emulated.</p> <p>Use a default configuration template for each particular model as a starting point for a custom configuration. This would ensure that the parameter is set correctly.</p> <p><b>Example:</b></p> <pre>set session hw_model="VAX_6610"</pre> <p><b>Available models are:</b></p> <ul style="list-style-type: none"> <li>• PDP_1193</li> <li>• PDP_1194</li> <li>• MicroVAX_3100_Model_96</li> <li>• MicroVAX_3100_Model_98</li> <li>• MicroVAX_3600</li> <li>• MicroVAX_3900</li> <li>• MicroVAX_II</li> <li>• VAXserver_3600</li> <li>• VAXserver_3600</li> <li>• VAXserver_3600_128</li> <li>• VAXserver_3600_512</li> <li>• VAXserver_3900</li> <li>• VAXserver_3900_128</li> <li>• VAXserver_3900_512</li> <li>• VAX_4000_Model_106</li> <li>• VAX_4000_Model_108</li> <li>• VAX_4000_Model_700</li> <li>• VAX_4000_Model_705</li> <li>• VAX_6000_Model_310</li> <li>• VAXstation_4000_Model_90</li> <li>• VAX_6610</li> <li>• VAX_6620</li> <li>• VAX_6630</li> <li>• VAX_6640</li> <li>• VAX_6650</li> <li>• VAX_6660</li> </ul>

## configuration\_name

<b>Parameter</b>	configuration_name
<b>Type</b>	Text string
<b>Value</b>	<p>Name of the Charon-VAX / Charon-PDP instance (it must be unique):</p> <pre>set session configuration_name="MSCDV1"</pre> <p> In provided templates this parameter is specified in the included configuration file "configuration_name.icfg".</p> <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-VAX / Charon-PDP 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-VAX / Charon-PDP execution session is stored.</p> <p style="text-align: center;"><b>Log specified as a file name</b></p> <p>It is possible to overwrite the existing log file or to extend it using the "General Settings#log_method" parameter.</p> <p> The "log_method" parameter is effective only when a single log file is specified, not a directory.</p> <p><b>Example:</b></p> <pre>set session log="C:\Charon\vax4106prod.log"</pre> <p style="text-align: center;"><b>Log specified as a directory</b></p> <p>Charon-VAX / Charon-PDP automatically creates individual log files for each Charon-VAX / Charon-PDP execution session. If the log parameter is omitted, Charon-VAX / Charon-PDP creates a log file for each Charon-VAX / Charon-PDP execution session in the directory where the emulator was started. In these two cases, the log rotation mode is enabled, meaning a new log file is created each time the virtual machine is started and when the log file size exceeds the one specified (see <a href="#">General Settings#log_file_size</a>) and/or when the log file is older than a specified number of days (see <a href="#">General Settings#log_rotation_period</a>).</p> <p> A shortcut located in the same directory will be created, pointing to the active log file. Its name is based on the hw_model parameter or the configuration_name parameter if specified.</p> <p>If the "configuration_name" parameter of the session is specified, the log file name is composed as follows:</p> <pre>&lt;configuration_name&gt;-YYYY-MM-DD-hh-mm-ss-xxxxxxxxx.log</pre> <p>If the "configuration_name" parameter is omitted, the log file name will have the following format:</p> <pre>&lt;hw_model&gt;-YYYY-MM-DD-hh-mm-ss-xxxxxxxxx.log</pre> <p>where "xxxxxxxxx" is an increasing decimal integer, starting from 00000000 to separate log files with the same time of creation (in case the log is being created faster than one log file per second).</p> <p> Only existing directory can be specified. If the directory specified does not exist, this will be considered as a flat file. No trailing backslash character is allowed.</p> <p><b>Example:</b></p> <pre>set session configuration_name="vax4106prod"  set session log="C:\Charon\Logs"</pre> <p>The execution of the virtual machine will create a log file, named C:\Charon\Logs\vax4106prod-2016-10-13-10-00-00-000000000.log (for example) and a shortcut named C:\Charon\Logs\vax4106prod.log pointing to this file. The shortcut will be updated when the log rotation will occur.</p> <p> Starting with version 4.11 build 204-11, log file numbering is implemented when log rotating is used: when starting the Charon emulator the first log file name will be like &lt;configuration_name&gt;-YYYY-MM-DD-hh-mm-ss-xxxxxxxxx.log where xxxxxxxxxxx will be 000000000, this number will increase with each log rotation to make it easier to find a log file set (1st log file at start + all its rotated files)</p>


## log\_method

<b>Parameter</b>	log_method
<b>Type</b>	Text string
<b>Value</b>	<ul style="list-style-type: none"> <li>• "append" (default)</li> <li>• "overwrite"</li> </ul> <p>Determines if the previous log information is maintained or overwritten.</p> <p>This parameter can only be specified in addition to the "log" parameter on the same line.</p> <p>This parameter is applicable only if the Charon-VAX / Charon-PDP log is stored to a file that is specified explicitly with the "log" parameter.</p> <p><b>Example:</b></p> <pre>set session log="log.txt" log_method="append"</pre>

## 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><b>Example:</b></p> <pre>set session log_show_messages="error, warning"</pre>


## log\_file\_size

<b>Parameter</b>	log_file_size
<b>Type</b>	Text string
<b>Value</b>	<p>If log rotation is enabled, the log_file_size parameter determines the log file size threshold at which the log is automatically rotated. Rotating log file size is multiple of 64K</p> <ul style="list-style-type: none"> <li>• "unlimited" or "0" (default) - the feature is disabled</li> <li>• "default" - default size is used (4Mb)</li> <li>• &lt;size&gt;[KMG] - size of the current log file in bytes with additional multipliers: <ul style="list-style-type: none"> <li>• K - Kilobyte - multiply by 1024</li> <li>• M - Megabyte - multiply by 1024*1024</li> <li>• G - Gigabyte - multiply by 1024*1024*1024</li> </ul> </li> </ul> <p><b>Example 1:</b></p> <pre>set session log_file_size="default"</pre> <p><b>Example 2:</b></p> <pre>set session log_file_size=10M</pre> <p> Minimum LOG File size is 64K, maximum is 1G. Setting a size less than 64K effectively makes the log file size unlimited.</p>

## log\_rotation\_period

<b>Parameter</b>	log_rotation_period
<b>Type</b>	Text string
<b>Value</b>	<ul style="list-style-type: none"> <li>• "default" - default value, 7 days. This value is used even if the "log_rotation_period" is not specified.</li> <li>• "daily" or "1"</li> <li>• "weekly" or "7"</li> <li>• "never"</li> <li>• &lt;N&gt; - in N days where N is greater than 0</li> </ul> <p>If the rotation log mode is enabled, this parameter controls switching to the next log file based on a period of time. If enabled, the switching to the next log file occurs at midnight.</p> <p><b>Example 1:</b></p> <pre>set session log_rotation_period="weekly"</pre> <p><b>Example 2:</b></p> <pre>set session log_rotation_period=14</pre>

## 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" and "off" (default).</p> <p>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 entry "The previous message has been repeated <i>N</i> times." and the counter is cleared.</p> <p> Stromasys highly recommends to let this value to "off" otherwise important information will not be reported in the log file.</p> <p><b>Example:</b></p> <pre>set session log_repeat_filter="off"</pre>

## log\_flush\_period

<b>Parameter</b>	log_flush_period
<b>Type</b>	Numeric
<b>Value</b>	<ul style="list-style-type: none"> <li>• &lt;period-in-seconds&gt;</li> </ul> <p>Defines a period of flushing log to disk.</p> <p>Default period is 60 seconds (it means that every minute log file is flushed to disk)</p> <p><b>Examples:</b></p> <pre>set session log_flush_period=30</pre>

## license\_key\_id

<b>Parameter</b>	license_key_id
<b>Type</b>	Text string
<b>Value</b>	<p>A set of Sentinel Key IDs that specifies the license keys to be used by Charon. It is also possible to use a keyword "<b>any</b>" to force Charon to look for a suitable license in all available keys if the license is not found in the specified keys.</p> <p><b>Example:</b></p> <pre>set session license_key_id = "1877752571,354850588,any"</pre> <p>Based on the presence of this parameter in the configuration file, Charon behaves as follows:</p> <ol style="list-style-type: none"> <li><b>1. No keys are specified (the parameter is absent)</b> Charon performs an unqualified search for any suitable key in unspecified order. If no key is found, Charon exits.</li> <li><b>2. One or many keys are specified</b> Charon performs a qualified search for a regular license key in the specified order. If it is not found, Charon exits (if the keyword "any" is not set).</li> </ol> <p>If the keyword "any" is specified then if no valid license has been found in the keys with specified ID's all other available keys are examined for valid license as well.</p> <p><b>Please note</b> that the order in which keys are specified is very important. If a valid license was found in the key which ID was not the first one specified in configuration file, then available keys are periodically re-scanned and if the key with the ID earlier in the list than the current one is found Charon tries to find a valid license there and in case of success switches to that key.</p>

## license\_id

<b>Parameter</b>	license_id
<b>Type</b>	Text string
<b>Value</b>	<p>A set of license identifiers that specifies the licenses to be used by Charon. This parameter is applicable only to licenses on which Stromasys placed restrictions on what products can be combined on a single license key. Please contact your Stromasys representative or VAR for more information.</p> <p><b>Example:</b></p> <pre>set session license_id = "2718281828,314159265"</pre> <p>If this parameter is set, Charon considers for validation only the available licenses with license ID parameter set and equal to one of the license ID's specified in the configuration. This prioritized list corresponds to the "Product License Number" line in the Product section of the license.</p>

## affinity

<b>Parameter</b>	affinity
<b>Type</b>	Text string
<b>Value</b>	<p>Overrides any initial process affinity mask provided by the host operating system. Once specified it binds the running instance of the emulator to particular host CPUs.</p> <p>Used for soft partitioning of the host CPU resources and/or for isolating host CPUs for other applications.</p> <p>By default the Charon-VAX / Charon-PDP 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 VAX CPUs and for the Charon VM itself (including the CPUs to be used for I/O - they are controlled by the "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\_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-VAX / Charon-PDP must use for I/O handling.</p> <p>By default the Charon VM reserves one third of all available host CPUs for I/O processing (round down, at least one). The "n_of_io_cpus" parameter overrides that by specifying the number of Charon I/O CPUs explicitly.</p> <p><b>Example:</b></p> <pre>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-VAX / Charon-PDP license connection is not present when the guest starts up, this parameter specifies how many times the Charon VM will try to establish the connection and, optionally, a period of time between retries.</p> <p><b>Syntax:</b></p> <pre>set session license_key_lookup_retry = "N [, T]"</pre> <p><b>Options:</b></p> <ul style="list-style-type: none"> <li>• N - Number of retries to look for license keys.</li> <li>• T - Time between retries in seconds. If not specified, 60 seconds are used</li> </ul> <p><b>Example 1:</b></p> <pre>set session license_key_lookup_retry = 1</pre> <p>If the license key is not found during initial scan, only one more attempt after 60 seconds is performed.</p> <p><b>Example 2:</b></p> <pre>set session license_key_lookup_retry = "1,30"</pre> <p>Same as above but retry in 30 seconds.</p> <p><b>Example 3:</b></p> <pre>set session license_key_lookup_retry = "3,10"</pre> <p>If the license key is not found during initial scan, do 3 more attempts waiting 10 seconds between them.</p> <p><b>Example 4:</b></p> <pre>set session license_key_lookup_retry = "5"</pre> <p>If the license key is not found during the initial scan, do 5 more attempts waiting 60 seconds between them.</p>

## File inclusion


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

### Format:

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

### Example:

```
include "c:\Charon\commonpart.icfg"
```

 The configuration template files are defined to use the "`configuration_name.icfg`" [configuration file](#) which is used by the Charon Virtual Machines Manager. If you use an existing configuration file, including this file is not necessary.

# Core Devices

## Table of Contents

- CPU
  - ace\_mode
- RAM
  - size
- TOY
  - container
- KW11-L and KW11-P Timers (PDP-11)
  - kw11 frequency
- ROM
  - container
- PDP-11 Boot ROM
  - ext\_rom
  - ext\_rom\_address
- EEPROM
  - container
- Auto Boot
  - MicroVAX3100, VAXstation 4000, VAX6310 and VAX 4000
  - MicroVAX II, MicroVAX 3600/3900 and VAXserver 3600/3900
  - PDP11/93 and PDP11/94
  - VAX66x0

## CPU

### ace\_mode

A CHARON-VAX emulated CPU is configured with the "ace\_mode" parameter.

Two VAX CPU implementations are available: the standard VAX instruction decoder and the optional high performance Advanced CPU Emulation mode ("ACE"). The ACE option optimizes the VAX instruction interpretation and significantly improves performance. It also requires approximately twice the amount of host memory to store the optimized code.

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 VAX instruction decoding; CHARON-VAX remains fully VAX hardware compatible and completely transparent to the VAX operating systems and applications.

Both CPU implementations passed the HP VAX Architecture (AXE) tests, the standard qualification for VAX instruction execution correctness.

The default VAX/PDP11 CPU mode is determined by the specific CHARON-VAX product license.

<b>Parameter</b>	ace_mode
<b>Type</b>	Boolean
<b>Value</b>	true or false

This statement enables the ACE mode if the CHARON-VAX license permits it. If this statement is omitted from the CHARON-VAX configuration file and the license permits it, "true" is the default, otherwise "false" is the default. For test purposes the ACE mechanism can be disabled with:

```
set cpu ace_mode=false
```

#### The command

```
"set cpu ace_mode=true"
```

**is ignored when the license does not permit ACE operation.**



The CHARON-VAX log file displays the status of the ACE option.

The ACE mode is disabled when the host system does not meet the minimum physical requirements for this operation. If the emulator appears to not 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.

## RAM

The CHARON-VAX / CHARON-PDP 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 = 512
```

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-VAX / CHARON-PDP 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 VAX/PDP11 systems:


Hardware Model	RAM size (in MB)			
	Min	Max	Default	Increment
PDP1193	2	4	2	2
PDP1194	2	4	2	2
MicroVAX_II	1	16	16	1,8,16
MicroVAX_3600	16	64	16	16
MicroVAX_3900	16	64	16	16
VAXserver_3600	16	64	16	16
VAXserver_3900	16	64	16	16
VAXserver_3600_128	32	128	32	32
VAXserver_3900_128	32	128	32	32
MicroVAX_3100_Model_96	16	128	16	16
VAXstation_4000_Model_90	16	128	16	16
VAX_4000_Model_106	16	128	16	16
VAX_6000_Model_310	32	512	32	32
VAXserver_3600_512	32	512	32	32
VAXserver_3900_512	32	512	32	32
MicroVAX_3100_Model_98	16	512	16	16
VAX_4000_Model_108	16	512	16	16
VAX_4000_Model_700	64	512	64	64
VAX_4000_Model_705	64	512	64	64
VAX_6610	128	3584	128	128

Hardware Model (cont.)	RAM size (in MB)			
	Min	Max	Default	Increment
VAX_6620	128	3584	128	128
VAX_6630	128	3584	128	128
VAX_6640	128	3584	128	128
VAX_6650	128	3584	128	128
VAX_6660	128	3584	128	128

## TOY

CHARON-VAX / CHARON-PDP 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-VAX / CHARON-PDP 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>

### Example:

```
set toy container="C:\Charon\my_virtual_system.dat "
```

The CHARON-VAX / CHARON-PDP 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.

## KW11-L and KW11-P Timers (PDP-11)

The KW11-L timer is used in PDP-11 system emulation. It supports a 50Hz, 60Hz and 70Hz frequency.

## kw11 frequency

A time correction mechanism handles any time slips between the emulated and the host system time. The KW11-L frequency can be set in the configuration file as follows:

```
set kw11 frequency=50
```


Characteristics:

- For PDP-11 configurations this device is loaded automatically.
- The KW11-P is a programmable timer for PDP-11 systems. The hardware KW11-P timer runs on a 100KHz clock and can be programmed to interrupt at any frequency up to 100KHz.
- The software KW11-P implementation is limited to 1KHz due to the host operating system limitations; it can be programmed to interrupt at any frequency up to 1KHz.
- The current implementation does not contain a way to handle a time slip recovery from the host system time. This can lead to time skew between the emulated and host system time when the programmed frequencies are not divisors of 1000.
- The KW11-P is loaded automatically. It does not have any configuration file parameters because it is set by the PDP-11 software.

## 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-VAX 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>

**Example:**

```
set rom container="C:\Charon\my_virtual_system.rom"
```

## PDP-11 Boot ROM

PDP-11 Boot ROM has two parameters: 'ext\_rom' and 'ext\_rom\_address'. Specify the external ROM file and offset within the file to load the ROM from. This feature allows changing the used boot ROM to a custom one.

### ext\_rom

<b>Parameter</b>	ext_rom
<b>Type</b>	Text string
<b>Value</b>	Specifies a file containing external boot code

## ext\_rom\_address

<b>Parameter</b>	ext_rom_address
<b>Type</b>	Numeric
<b>Value</b>	Specifies an offset to be used for the file specified by the ext_rom parameter


### Example:

```
set rom ext_rom="custom_rom.bin"
set rom ext_rom_address=0
```

The ROM file must contain a binary dump.

## EEPROM

### container

<b>Parameter</b>	container
<b>Type</b>	Text string
<b>Value</b>	<p>A string specifying a file name to store the content of the EEPROM for VAX 6000 series.</p> <p><b>Example:</b></p> <pre>set eeprom container="vx6k610.rom"</pre> <p>This command enables EEPROM parameters (e.g., default boot drive) to be automatically saved to a specified file. The EEPROM file is created in the directory in which CHARON-VAX starts. The EEPROM file is created or overwritten each time any parameter relevant to the EEPROM content is changed.</p> <p> it is recommended to specify the full path to this file</p>

### Example:

```
set eeprom container="C:\Charon\my_virtual_system.rom"
```

## Auto Boot

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

### MicroVAX3100, VAXstation 4000, VAX6310 and VAX 4000

Those models boot automatically if the correct boot flags are set (and saved in the VAX console files) using the following command:

<b>Console Parameter</b>	halt
<b>Description</b>	<p>Determines whether the MicroVAX3100, VAXstation 4000, VAX6310 and VAX 4000 boot automatically if the correct boot flags are set (and saved in the VAX console files).</p> <p>The value is "reboot" .</p> <p><b>Example:</b></p> <pre>&gt;&gt;&gt;set halt reboot</pre>

**Important:** the "toy container" and "rom container" parameters must be specified in the configuration file to store the boot flags.

### MicroVAX II, MicroVAX 3600/3900 and VAXserver 3600/3900

The ROM of the MicroVAX II, MicroVAX 3600, MicroVAX 3900, VAXserver 3600 and VAXserver 3900 servers does not allow the VAX console to accept the command setting "auto-boot". Instead, an automatic boot on startup can be specified in the CHARON-VAX configuration file as follows:

<b>Parameter</b>	bdr boot
<b>Type</b>	Text string
<b>Value</b>	<p>Determines whether the MicroVAX II, MicroVAX 3600, MicroVAX 3900, VAXserver 3600 and VAXserver 3900 servers boot automatically if the correct boot flags are set (and saved in the VAX console files).</p> <p>The value is "auto" .</p> <p><b>Example:</b></p> <pre>set bdr boot=auto</pre>

**Important:** the "toy container" and "rom container" parameters must be specified in the configuration file to store the boot flags.

## PDP11/93 and PDP11/94

The ROM of the PDP11/93 and PDP11/94 does not allow the PDP11 console to accept the command setting "auto-boot". Instead, automatic boot on startup can be specified in the CHARON-PDP configuration file as follows:

<b>Parameters</b>	cpu_0 auto_boot
<b>Type</b>	Text string
<b>Value</b>	<p>Determines whether the PDP11/93 or PDP11/94 server boots automatically from the specified disk.</p> <p>The value is the disk drive to boot from.</p> <p><b>Example:</b></p> <pre>set cpu_0 auto_boot = "DU0"</pre>

**Important:** the "toy container" and "rom container" parameters must be specified in the configuration file to store the boot flags.

## VAX66x0

<b>Parameter</b>	xmi boot
<b>Type</b>	Text string
<b>Value</b>	<p>Determines whether the CHARON VAX66x0 startup procedure stops at the "&gt;&gt;&gt;" prompt after self-tests.</p> <p>The value can be "auto" or "manual" (default).</p> <p><b>Example:</b></p> <pre>set xmi boot = "auto"</pre> <p>The value "auto" enables automatic boot from a <u>default</u> boot specification, previously configured in the VAX console. The value "manual" disables the automatic boot once the self tests are passed.</p>

**Important:** the "toy container" and "rom container" parameters must be specified in the configuration file to store the boot flags.

# Serial lines

## Table of Contents

- General Description
  - Before build 204-14
  - Starting with build 204-14
- Console
  - alias
  - rts
  - dsr
  - communication
  - line
- Serial line controllers
  - address
  - vector
  - line
  - communication
  - rts
  - dsr
  - tx\_q\_max\_depth
- Mapping Serial line controllers to system resources
  - Types of serial line mapping
  - physical\_serial\_line
    - line
    - baud
    - break\_on
    - stop\_on
    - log
    - log\_file\_size
  - virtual\_serial\_line
    - host
    - port
    - application
    - break\_on
    - stop\_on
    - log
    - log\_file\_size
    - log\_flush\_period
    - connection\_override
    - access\_control
  - Example - mapping a console line to an onboard serial line
- Linking serial controller port to host connection
- OPA0 console configuration examples
  - Using legacy syntax (not recommended)
  - Using new syntax (recommended)

## General Description

### Before build 204-14

Configuration of the CHARON-VAX / CHARON-PDP serial lines is performed in 3 steps:

1. Loading the virtual serial lines' controller, for example:

```
load DHV11/DHV11 TXA
```

In this example, an instance of a "DHV11" serial line controller is loaded and named "TXA".

Note that the VAX/PDP11 console adapters ("UART", "QUART") do not need to be loaded; they are preloaded.

2. Mapping an object type to host resources. For example for the virtual serial lines (see below):

```
load virtual_serial_line/chserial TTA1
set TTA1 port = 10003 application = "putty -load CHTERM-VT100 -title TTA1@XYZ -P 10003"
```

In this example the object "virtual\_serial\_line" is loaded, named "TTA1", mapped to the "putty" terminal emulator and connected to this CHARON instance using the port 10003 (this setting is also specified via the parameter "-P". "CHTERM-VT100" parameter stands for the Putty terminal emulator specific settings. "-title" parameter is used for naming the terminal emulator window.

3. Connect the loaded virtual line controller and the mapped object:

```
set TXA line[5]=TTA1
```

In this example, the 6th line of the DHV11 controller, "TXA", loaded in step 1, is connected to the mapping object, "TTA1", loaded in step 2.

**i** This syntax is still supported after build 204-14 but using it will not provide the new features provided by this build.

### Starting with build 204-14

Configuration of the CHARON-VAX / CHARON-PDP serial lines is performed as detailed below:

#### Example for MicroVAX II, 3600, 3900, VAX 4000-700/705:

```
set UART alias=OPA0
set OPA0 port=10003 application="putty -load CHTERM-VT100 -title OPA0@myvax -P 10003"
```

**i** The first line above is optional as Charon-VAX/PDP presets UART's alias to OPA0

#### Example for MicroVAX 3100-96/98, VAX 4000-106/108, and VAXstation 4000-90:

```
set QUART alias[3] = OPA0
set OPA0 port=10003 application="putty -load CHTERM-VT100 -title OPA0@myvax -P 10003"
```

**i** The first line above is optional as Charon-VAX/PDP presets UART's alias to OPA0

Refer to [Mapping Serial line controllers to system resources](#) for details of mapping and examples provided in the configuration file templates located in the Charon installation folder.

Please note using the old method with "load virtual\_serial\_line OPA0" will invalidate some features detailed further.

## Console

CHARON-VAX / CHARON-PDP offers a one- or four-port serial console, depending on the specified VAX/PDP11 model. The one port serial line controller is identified in CHARON-VAX / CHARON-PDP with the name UART. The four port serial lines controller is identified in CHARON-VAX with the name QUART.

UART is used in Qbus systems only (e.g. the MicroVAX/VAXserver 3600/3900, PDP11/93 and PDP11/94).


QUART is used in SCSI (e.g. MicroVAX 3100 model 96/98, VAXstation 4000 model 90) and SCSI/Qbus systems (e.g. VAX4000 model 106/108). The last QUART line (line[3]) is the console port (known in VAX/VMS as OPA0).




CHARON-VAX / CHARON-PDP console ports can be configured to connect to an external terminal via the host system COM port or can be connected via TCP/IP.

## alias


 This applies to new syntax only, not if "load ..." is used.

<b>Parameter</b>	alias
<b>Type</b>	Identifier
<b>Value</b>	<p>The main purpose of this parameter is migration from old CHARON systems (which do not have the described implementation of consoles) to the current design, since it allows retaining the original name used for parametrization, since the rest of the parameters stay the same in both implementations.</p> <p><b>Example:</b></p> <pre>set UART alias=OPA0</pre> <p> This parameter is optional as Charon-VAX/PDP presets UART's alias to OPA0</p>

## rts

<b>Parameter</b>	rts[<line>]
<b>Type</b>	Text string
<b>Value</b>	<ul style="list-style-type: none"> <li>"On" - assert the RTS (Request To Send) signal</li> <li>"Off" - clear the RTS signal (default)</li> <li>"DTR" - assert the RTS signal as soon as the DTR signal is asserted (  Applicable only for QUART )</li> </ul>

## dsr

<b>Parameter</b>	dsr[<line>]
<b>Type</b>	Text string
<b>Value</b>	<ul style="list-style-type: none"> <li>"On" - always reports DSR signal assertion</li> <li>"Off" - always reports DSR signal deassertion</li> <li>"DSR" - use the DSR signal of the physical serial line (if configured)</li> <li>"CD", "DCD", "RLSD" - use the CD (carrier detect) signal of the physical serial line (if configured)</li> </ul> <p></p>

## communication

<b>Parameter</b>	communication[<line>]
<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 serial lines carrying binary (packet) protocols, which are used mainly for communicating with PLCs</li> </ul>

## line

<b>Parameter</b>	line[<line>]
<b>Type</b>	Identifier
<b>Value</b>	This parameter is used to connect a particular serial line interface to the controller. See below for details.

Note that the "line" parameter in the table above is applicable only in the case of QUART.

All the values in this table are case insensitive.

**Example:**

```
set QUART rts[2]="DTR"
set QUART dsr[2]="On"
set QUART communication[2]="binary"
```

**Line 2 of the QUART is the only one which can be used for connecting modems.** Therefore, the "DSR" parameter for that line (i.e. "dsr[2]") is internally set to the appropriate value ("CD") but can be changed from the configuration file. Values for the "rts" and "dsr" parameters for the lines other than 2 are not visible for any applications running on CHARON-VAX / CHARON-PDP.

## Serial line controllers

Asynchronous serial line multiplexers are capable of serving up to 8 asynchronous serial lines (the DHW42-BA supports up to 16 lines).

The following asynchronous serial line multiplexers are supported:

VAX model	Asynchronous serial line emulation
PDP-11/93, PDP-11/94, MicroVAX II, MicroVAX 3600, MicroVAX 3900, VAXserver 3600, VAXserver 3900 (QBUS systems)	CXA16, CXB16, CXY08, DHQ11, DHV11, DZV11, DZQ11, DL11, DLV11, DZ11
MicroVAX 3100 - 96, MicroVAX 3100 - 98 (SCSI systems)	DHW42-AA, DHW42-BA, DHW42-CA
VAX4000 - 106, VAX4000 - 108, VAX4000 - 700, VAX4000 - 705 (QBUS/SCSI systems)	CXA16, CXB16, CXY08, DHQ11, DHV11, DZV11, DZQ11, DLV11, DHW42-AA, DHW42-BA, DHW42-CA
VAX6310, VAXstation 4090	N/A

The following names are used for the multiplexers:

Device name	Module name
DHV11	DHV11
DHQ11	DHV11
CXY08	DHV11
CXA16	DHV11
CXB16	DHV11
DHW42AA	DHV11
DHW42BA	DHV11
DHW42CA	DHV11
DZV11, DZ11	DZ11
DZQ11	DZ11
DL11, DLV11	DL11

The following example loads an instance of an asynchronous serial line multiplexer:

```
load DHQ11/DHV11 TXA
```

Only **one instance of DHW42** can be loaded. There is no restriction on the number of the other multiplexers.

The multiplexers offer the following configuration parameters, specified with the "set" command:

## address

<b>Parameter</b>	address
<b>Type</b>	Numeric
<b>Value</b>	<p>Specifies CSR address. The address must be a valid QBUS 22-bit wide address in I/O space.</p> <p>Default values are 017760440 for the DHV11-family controllers and 017760100 for the DZV11/DZQ11 family controllers, which are the factory settings for asynchronous serial line multiplexers.</p> <p><b>This parameter is not applicable to DHW42-xx serial line controllers.</b></p>

## vector

<b>Parameter</b>	vector
<b>Type</b>	Numeric
<b>Value</b>	<p>Specifies the interrupt vector.</p> <p>The default value is 0300, which is the factory setting for asynchronous serial line multiplexers.</p> <p><b>This parameter is not applicable to DHW42-xx serial line controllers.</b></p>

## line

<b>Parameter</b>	line[N] N=0...3(7,15)
<b>Type</b>	Identifier
<b>Value</b>	Specifies the name of the serial line interface object in the configuration to which the N-th line of the multiplexer is connected. See below for details.

## communication

<b>Parameter</b>	communication[N] N=0...4(7,15)
<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 serial lines carrying binary (packet) protocols, which are used mainly for communicating with PLCs</li> </ul>

## rts

<b>Parameter</b>	rts[N] N=0...3(7,15)
<b>Type</b>	Text String
<b>Value</b>	<p>Controls the RTS signal of the Nth line of the multiplexer.</p> <ul style="list-style-type: none"> <li>"On" - assert the RTS (Request To Send) signal</li> <li>"Off" - clear the RTS signal (default)</li> <li>"DTR" - assert the RTS signal as soon as DTR signal is asserted</li> </ul> <p>When left blank (initial state), the level of the RTS signal is as requested by the VAX/PDP11 software.</p>

## dsr

<b>Parameter</b>	dsr[N] N=0...3(7,15)
<b>Type</b>	Text String
<b>Value</b>	<ul style="list-style-type: none"> <li>"On" - always reports the DSR signal assertion</li> <li>"Off" - always reports the DSR signal deassertion</li> <li>"DSR" - use the DSR signal of the physical serial line (if configured)</li> <li>"CD", "DCD", "RLSD" - use the CD (carrier detect) signal of the physical serial line (if configured)</li> </ul> <p><b>This parameter is applicable only to the DZV11 and DZQ11 serial lines controllers.</b></p>

## tx\_q\_max\_depth

<b>Parameter</b>	tx_q_max_depth[N] N=0...3(7,15)
<b>Type</b>	Numeric
<b>Value</b>	Specifies the depth of the TX FIFO for the N-th line of the multiplexer. Possible values are 1...1000, initially set to 1, to properly represent the hardware limitation of certain multiplexers. Values greater than 1 improve transmission rate of the corresponding lines, but break correspondence to the original hardware.  <b>This parameter is applicable only to the DHV11 serial lines controllers.</b>

Use the "address" and "vector" parameters to load several instances of the Qbus multiplexers. Both "address" and "vector" parameter values must be unique for every instance of a QBUS multiplexer.

Read the VAX/PDP11 hardware documentation and the VM system management documentation to understand how to correctly assign the "address" and "vector" parameters.

**Example:** loading 2 instances of DHV11

```
load DHV11/DHV11 TXA address=017760440 vector=0300
load DHV11/DHV11 TXB address=017760460 vector=0310
```

**Example:** loading DHW42CA

```
load DHW42CA/DHV11 TXA
```

## Mapping Serial line controllers to system resources

### Types of serial line mapping

Type	Function
physical_serial_line	This type of mapping associates a COM port on a host system with an emulated VAX/PDP11 serial line controller virtual "line". The COM port can be a physical hardware port or a logical COM port.
virtual_serial_line	This type of mapping associates a network connection on the host system with an emulated VAX/PDP11 serial line controller virtual "line"

**Example:**

```
load physical_serial_line/chserial OPA0
```

## physical\_serial\_line


### line

<b>Parameter</b>	line
<b>Type</b>	Text string
<b>Description</b>	A defined COM port on a host system in the form of "\\.\COMn"  <b>Example:</b> <pre>set OPA0 line="\\.\COM1"</pre>


### baud

<b>Parameter</b>	baud
<b>Type</b>	Numeric
<b>Description</b>	Forces the baud rate of the corresponding COM port to be a specified value.  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.  <b>Example:</b> <pre>set OPA0 baud=38400</pre>

### break\_on

<b>Parameter</b>	break_on
<b>Type</b>	Text string
<b>Description</b>	Specifies what byte sequences received over the physical serial line will trigger a HALT command.   This parameter works only for the console line (for the one UART line and "line[3]" of QUART).  Specify the following values: "Ctrl-P", "Break" or "none" ("none" disables triggering a HALT condition).  <b>Example:</b> <pre>set OPA0 break_on="Ctrl-P"</pre>  The default value is "Break" for line 3 of QUART and "none" for other lines.


## stop\_on

<b>Parameter</b>	stop_on
<b>Type</b>	Text string
<b>Description</b>	<p>Specifies what byte sequences received over the physical serial line will trigger a STOP condition. The STOP condition causes CHARON-VAX / CHARON-PDP to exit.</p> <p> This parameter works only for the console line (for the one UART line and "line[3]" of QUART).</p> <p>Specify the one of the following values: "F6" or "none" ("none" disables triggering a STOP condition).</p> <p><b>Example:</b></p> <pre>set OPA0 stop_on="F6"</pre> <p>The default value is "none".</p> <p>Setting "F6" triggers the STOP condition upon receipt of the "&lt;ESC&gt;[17~" sequence. Terminals usually send these sequences by pressing the <b>F6</b> button</p>

## log

<b>Parameter</b>	log
<b>Type</b>	Text string
<b>Description</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-VAX / CHARON-PDP 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-VAX / CHARON-PDP does not create a console log.</p> <p><b>Example 1:</b></p> <pre>set OPA0 log="log.txt"</pre> <p><b>Example 2:</b></p> <pre>set OPA0 log="C:\Charon\Logs"</pre> <p><b>Notes:</b></p> <ul style="list-style-type: none"> <li>Specify only an existing directory.</li> <li>If you want to store the OPA0 log file in the Charon virtual machine <a href="#">home directory</a> (default for virtual machine log file), use: <code>set OPA0 log=".\"</code></li> </ul>

## log\_file\_size

<b>Parameter</b>	log_file_size
<b>Type</b>	Text string
<b>Description</b>	<p>If log rotation is enabled, the log_file_size parameter determines the log file size threshold at which the log is automatically rotated.</p> <ul style="list-style-type: none"> <li>• "unlimited" or "0" (default) - the feature is disabled</li> <li>• "default" - default size is used (4Mb)</li> <li>• &lt;size&gt;[KMG] - size of the current log file in bytes with additional multipliers: <ul style="list-style-type: none"> <li>• K - Kilobyte - multiply by 1024</li> <li>• M - Megabyte - multiply by 1024*1024</li> <li>• G - Gigabyte - multiply by 1024*1024*1024</li> </ul> </li> </ul> <p><b>Example 1:</b></p> <pre>set OPA0 log_file_size="default"</pre> <p><b>Example 2:</b></p> <pre>set OPA0 log_file_size=10M</pre> <p> Minimum log file size is 64K, maximum is 1G. Setting size less than 64K effectively makes the log file unlimited.</p>

**Example:** mapping a console line to a host physical serial line

```
load physical_serial_line/chserial OPA0
set OPA0 line="\\.COM1"
```

## virtual\_serial\_line

### host

<b>Parameter</b>	host
<b>Type</b>	Text string
<b>Description</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>Description</b>	The TCP/IP port number for the virtual serial line. A virtual serial line always listens on this port for incoming connection requests.


## application

<b>Parameter</b>	application
<b>Type</b>	Text string
<b>Description</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 OPA0 application = "putty -load CHTERM-VT100 -title OPA0@XYZ -P 10003"</pre> <p>In this example the terminal emulator application "putty" is started with the parameters:</p> <ul style="list-style-type: none"> <li>• "-load CHTERM-VT100" telling it to load a specific saved session named "CHTERM-VT100" (created separately), from the host registry</li> <li>• "-title" specifies the name of the given instance of putty terminal emulator</li> <li>• "-P" stands for communication port used</li> </ul> <p>The "application" parameter is often combined with a "port" parameter:</p> <pre>set TTA0 port = 10004 application = "putty -load CHTERM-VT100 -title TTA0@XYZ -P 10004"</pre> <p>The CHTERM-VT100 settings in form of ".reg" file can be found under Charon installation location in the subfolder "Build_xxxx\x86"</p>

## break\_on

<b>Parameter</b>	break_on
<b>Type</b>	Text string
<b>Description</b>	<p>Specifies what byte sequences received over a virtual serial line triggers a HALT command.</p> <p> This parameter works only on the console line (for CHARON-VAX / CHARON-PDP it is the only line of UART and "line[3]" of QUART).</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><b>Example:</b></p> <pre>set OPA0 break_on="Ctrl-P"</pre> <p>The default value is "Break" for line 3 of QUART and "none" for other lines.</p>


## stop\_on

<b>Parameter</b>	stop_on
<b>Type</b>	Text string
<b>Description</b>	<p>Specifies what byte sequences received over the virtual serial line will trigger a STOP condition. The STOP condition causes CHARON-VAX / CHARON-PDP to exit.</p> <p> This parameter works only on the console line (for CHARON-VAX / CHARON-PDP it is the only line of UART and "line[3]" of QUART).</p> <p>Specify one of the following values: "F6" or "none" ("none" disables triggering a STOP condition). The commands are case insensitive.</p> <p><b>Example:</b></p> <pre>set OPA0 stop_on="F6"</pre> <p>The default value is "none".</p> <p>Setting "F6" triggers the STOP condition upon receipt of the "&lt;ESC&gt;[17~" sequence.</p>

## log

<b>Parameter</b>	log
<b>Type</b>	Text string
<b>Description</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-VAX / CHARON-PDP 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-VAX / CHARON-PDP does not create any console log.</p> <p><b>Example 1:</b></p> <pre>set OPA0 log="log.txt"</pre> <p><b>Example 2:</b></p> <pre>set OPA0 log="C:\Charon\Logs"</pre> <p><b>Notes:</b></p> <ul style="list-style-type: none"> <li>Specify only an existing directory</li> <li>If you want to store the OPA0 log file in the Charon virtual machine <a href="#">home directory</a> (default for virtual machine log file), use: <pre>set OPA0 log=".\"</pre></li> </ul>


## log\_file\_size

<b>Parameter</b>	log_file_size
<b>Type</b>	Text string
<b>Description</b>	<p>If log rotation is enabled, the log_file_size parameter determines the log file size threshold at which the log is automatically rotated.</p> <ul style="list-style-type: none"> <li>• "unlimited" or "0" (default) - the feature is disabled</li> <li>• "default" - default size is used (4Mb)</li> <li>• &lt;size&gt;[KMG] - size of the current log file in bytes with additional multipliers: <ul style="list-style-type: none"> <li>• K - Kilobyte - multiply by 1024</li> <li>• M - Megabyte - multiply by 1024*1024</li> <li>• G - Gigabyte - multiply by 1024*1024*1024</li> </ul> </li> </ul> <p><b>Example 1:</b></p> <pre>set OPA0 log_file_size="default"</pre> <p><b>Example 2:</b></p> <pre>set OPA0 log_file_size=10M</pre> <p> Minimum log file size is 64K, maximum is 1G. Setting size less than 64K effectively makes the log file unlimited.</p>


## log\_flush\_period


<b>Parameter</b>	log_flush_period
<b>Type</b>	Numeric
<b>Value</b>	<ul style="list-style-type: none"> <li>• &lt;period-in-seconds&gt;</li> </ul> <p>Defines a period of flushing log to disk.</p> <p>Default period is 60 seconds (it means that every minute log file is flushed to disk)</p> <p><b>Example:</b></p> <pre>set OPA0 log_flush_period=30</pre>

## connection\_override

<b>Parameter</b>	connection_override
<b>Type</b>	text string
<b>Value</b>	<p>"enable"</p> <p>Allows new connection to override existing connection, if any. Enabled connection override on OPA0 allows to intercept virtual serial console.</p> <p>When emulator detects new connection request on the port (10003 for the below example), it closes old connection, if any, and switches to the new one.</p> <p><b>Example:</b></p> <pre>set OPA0 port = 10003 connection_override = enable</pre> <p> This is implemented only for serial lines using the new syntax, not for lines using the legacy syntax (load virtual_serial_line ...).</p>

## access\_control

 Available only since build 204-14

<b>Parameter</b>	access_control
<b>Type</b>	text string
<b>Value</b>	<p>"disable"</p> <p>Since build 204-14, Incoming connection requests are by default filtered for virtual serial lines and then allowed only for the localhost. This is to avoid security scanners that can block the port.</p> <p><b>Example:</b></p> <pre>set OPA0 port = 10003 set OPA0 access_control = disable</pre> <p> This is implemented only for serial lines using the new syntax, not for lines using the legacy syntax (load virtual_serial_line ...).</p>

## Example - mapping a console line to an onboard serial line

```
load virtual_serial_line/chserial OPA0
set OPA0 port=10003 stop_on="F6"
```

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/chserial TTA0 host="192.168.1.1" port=10000
```

In this example CHARON-VAX / CHARON-PDP 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/chserial TTA0 host="192.168.1.1:20000" port=10000
```

In this example CHARON-VAX / CHARON-PDP 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-VAX / CHARON-PDP to an application that communicates to CHARON-VAX / CHARON-PDP as described below.

**Example:** two CHARON systems connected to each other

On host "A":

```
load virtual_serial_line/chserial TXA0 port=5500 host="B"
```

On host "B":

```
load virtual_serial_line/chserial TXA0 port=5500 host="A"
```

On these two hosts, executing CHARON-VAX, the two TXA0 lines connect to each other, thus creating a "serial" cable between the two emulated VAXes. The sequential order in which the instances of CHARON-VAX / CHARON-PDP are started makes no difference.

## Linking serial controller port to host connection

The final step in the CHARON-VAX / CHARON-PDP serial line configuration is the association of each loaded serial port with a CHARON-VAX / CHARON-PDP host connection instance as follows:

```
set <serial controller instance name> line[<line number>]=<serial line instance name>
```

**Example:**

```
set quart line[0]=TTA0
```

This command connects the first serial line ("line[0]") of a "QUART" serial line controller to a CHARON-VAX / CHARON-PDP connection instance named "TTA0". As explained earlier, TTA0 may be a virtual serial line connected to a port or a physical serial line connected to a host serial port or a virtual terminal.

In the example below, the command connects the sixth serial line of a previously loaded controller (named "TXA") to "TTA1". "TTA1" could be defined, for example, as a physical serial line connected to COM/COM port:

```
set TXA line[5]=TTA1
```

## OPA0 console configuration examples

### Using legacy syntax (not recommended)

This example, using a MicroVAX 4000-106, maps OPA0 to port 10003, enable F6 key (emulator stop) and logs the console input/output to a rotating log file in its home directory:

```
load virtual_serial_line OPA0
set OPA0 port=10003
set quart line[3]=OPA0
set OPA0 stop_on="F6"
set OPA0 log=".\"
```

### Using new syntax (recommended)

This example, using a MicroVAX 4000-106, maps OPA0 to port 10003, enable F6 key (emulator stop), logs the console input/output to a rotating log file in its home directory and enables the connection\_override feature:

```
set QUART alias[3]=OPA0 (optional starting with build 204-13)
set OPA0 port=10003
set OPA0 stop_on="F6"
set OPA0 log=".\"
set OPA0 connection_override=enable
```

# Remote Management Console (RMC)

## Table of Contents

- General Description
  - Parameter
  - Examples
- Connection
- Available commands

## General Description

The purpose of the Remote Management Console is to let the emulator trigger actions on the Charon host to properly unload and save a vtape container while the guest operating system is running.

Originally, the tape **offline** state in Charon-VAX and Charon-AXP was not persistent. An automatic, periodic loading function will bring the tape device **online** again using the same container. That means that subsequent operations might overwrite the tape container.

To improve this behavior, an initially persistent **offline** state was introduced that could be configured by providing an "empty tape drive" using a name of **.vtape**. With this improvement, a virtual tape device will remain **offline**, until a valid name is provided during the runtime of the emulator. This name can be provided in two ways:

- CHARONCP: SCSI devices on OpenVMS only (see "Charon Guest Utilities for OpenVMS" chapter)
- Remote Management Console (this section)

However, this first improvement did not allow to return to an "empty tape drive" once a valid name had been provided. Hence, the automatic loading process would once again create the problem of possibly overwriting an existing container. This problem triggered an improvement that will allow a **persistent offline state**. The persistent offline state is configured on platforms where it is supported by adding a new keyword to the removable parameter as shown in the example below from a configuration file:

```
set PKA container[0] = ".vtape" removable = "noauto"
```

With this configuration, a tape device in Charon behaves as follows. When the tape device is put into status **offline**, it clears the runtime value of the associated container turning it into an "empty tape drive". Any further operations on this tape device are only possible after changing value of the container to a meaningful value and loading the tape. This behavior makes the runtime **offline** state persistent.

To load the Remote Management Console use the following syntax:

```
load remote_management_console RMC
```

 Only one Remote Management Console can be set per Charon instance.

## Parameter

<b>Parameter</b>	port
<b>Type</b>	Numeric
<b>Value</b>	The TCP/IP port number for the remote management console. If multiple Charon instances are running on a server, ensure the port number is unique across the platform.

## Examples

### Example 1:

```
load remote_management_console RMC port=13000
```

### Example 2:

```
load remote_management_console RMC
set RMC port=13000
```

## Connection

The RMC is reachable via the configured TCP port using (for example) a **telnet** client.

### Notes:

- Line mode has to be used (default) and not character mode during telnet connection.
- If you use 'putty' to access the RMC, use "Raw" connection type and not "Telnet" and ensure the "Implicit CR in every LF" option in the "Terminal" settings is checked.
- If you access from a remote location to the RMC port, please define appropriate firewall rules if necessary.

### Example:


```
Remote Management Console - port 13000
1, Ok
4, Unknown
help
hello
help
set <name> {<parameter> = <value>}*
show removable
acquire <name> <unit-no>
release <name> <unit-no>
bye
1, Ok
bye
```



## Available commands


Short description of the available commands:

Command	Description
<b>help</b>	Usage information as shown in the example above
<b>hello</b>	Currently no user-relevant function
<b>set</b> <name> {<parameter> = <value>}	The set command is used to set a meaningful name for a vtape container. <b>Example:</b> <pre>set PUA container[11] = mytape1.vtape</pre>
<b>show removable</b>	Lists the removable devices configured for the specific emulator instance.
<b>acquire</b> <name> <unit-no>	Brings a vtape <b>online</b> . <b>Example:</b> <pre>acquire pua 11</pre>
<b>release</b> <name> <unit-no>	Brings a vtape <b>offline</b> . If <b>noauto</b> is configured, an empty container name is set. <b>Example:</b> <pre>release pua 11</pre>
<b>show virtual</b> <serial line>	List configured serial lines. If run without parameters, all serial lines will be listed.
<b>disconnect</b> <name> <unit-no>	Disconnect virtual serial line. <b>Example:</b> <pre>disconnect com2 0</pre>
<b>bye</b>	Disconnect from RMC

 Commands cannot be shortened

**Example:**

```
1, Ok
show removable
PUA, 11, offline, ".vtape"
PUA, 12, offline, ".vtape"
1, Ok
set PUA container[11] = xxx.vtape
1, Ok
show removable
PUA, 11, offline, "xxx.vtape"
PUA, 12, offline, ".vtape"
1, Ok
acquire pua 11
1, Ok
show removable
PUA, 11, available, "xxx.vtape"
PUA, 12, offline, ".vtape"
1, Ok
release pua 11
1, Ok
show removable
PUA, 11, offline, "xxx.vtape"
PUA, 12, offline, ".vtape"
1, Ok
bye
```

-  Future versions may contain additional features to support the automation of the functionality that the Remote Management Console provides. Currently, if you need support for an automation project, please contact your Stromasys representative or your Stromasys VAR.

# Disks and tapes


## Contents

- [MSCP and TMSCP Controllers](#)
- [SCSI Controllers](#)
- [DSSI Subsystem](#)
- [CI Subsystem](#)

## Note on virtual tapes

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

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

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

### To perform backup:

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

### To restore from a backup:

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

CHARON does not support multi-volume backup for tape images. If some multi-volume set (in form of tape images) has to be restored it is recommended to configure several tape drives in CHARON VM 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:...
```

# MSCP and TMSCP Controllers


## Table of Contents

- Introduction
- RQDX3 Controller
  - address
  - max\_n\_of\_units
  - container
  - media\_type
  - geometry
  - use\_io\_file\_buffering
- TQK50 and TUK50 Controllers
  - address
  - container
  - media\_type
  - geometry
- KDM70 Controller
  - xmi\_node\_id
  - container
  - media\_type
  - geometry
  - use\_io\_file\_buffering
- KDB50 Storage Controller
  - vax\_bi\_node\_id
  - container
  - media\_type
  - geometry
  - use\_io\_file\_buffering

## Introduction

CHARON-VAX / CHARON-PDP provides MSCP controllers for hardware disks (including floppy and CD/DVD) and disks images. TMSCP controllers provide support for hardware tapes and tape images.

MSCP and TMSCP controllers are added to the configuration using the "load" command. The individual units are defined by using the container parameter.

-  MSCP devices appear in VMS as DUA for the first controller and DUB for the second controller, etc.  
TMSCP devices appear in VMS as MUA, MUB, etc.

When adding multiple MSCP or TMSCP controllers, follow the QBUS addressing conventions.

When a tape or disk image, connected to an emulated TMSCP or MSCP controller, is dismounted, in VAX/VMS, it is disconnected from CHARON-VAX / CHARON-PDP 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-VAX / CHARON-PDP disk images while CHARON-VAX / CHARON-PDP is running, please take care to minimize the risk of overloading a heavily loaded CHARON-VAX / CHARON-PDP host system. For example, using a sequential series of simple ftp binary copies is less resource intensive and thus less disruptive than multiple, simultaneous copies.

Empty disk images are created with the "mkdiskcmd" utility. Tape images (\*.vtape) will be created automatically if they don't exist (no utility needed).

CHARON-VAX / CHARON-PDP is able to boot from disk images of any VMS/RSX11/RT11 version (for VAX/VMS starting with 4.5 or higher for MicroVAX II or VAX 3600 and VAX/VMS 5.5-2 or higher for the VAX4000).

## RQDX3 Controller

The CHARON-VAX / CHARON-PDP QBUS system provides support for RQDX3 disk controllers. The original RQDX3 disk controller is capable of serving up to 4 disk units. CHARON-VAX / CHARON-PDP extends this limit so that the RQDX3 disk controller can be configured with up to 256 disk units. Normally all 256 disks can be connected to one MSCP disk controller, but if an application does intensive simultaneous I/O to more than 16 disks on one MSCP controller, it is recommended to configure additional RQDX3 controllers.

Use the following command to load an instance of an RQDX3 disk controller:

```
load RQDX3/RQDX3 <logical name>
```

### Example:

```
load RQDX3/RQDX3 DUA
```

The RQDX3 offers the following configuration parameters, which are specified with the "set" command:


### address

<b>Parameter</b>	address
<b>Type</b>	Numeric
<b>Value</b>	<p>Specifies the CSR address. This address must be a valid QBUS 22-bit address in IO space.</p> <p>Initially its value is 017772150, which is the factory setting for the RQDX3 disk controller.</p> <p>Use the "address" parameter for loading several instances of RQDX3. The "address" parameter value must be unique for every instance of the controller.</p>

### max\_n\_of\_units

<b>Parameter</b>	max_n_of_units
<b>Type</b>	Numeric
<b>Value</b>	<p>Specifies the maximum number of units supported by the controller. Possible values are 4...9999.</p> <p>The default value is 9999.</p>

## container

<b>Parameter</b>	container[N] N=0...9999
<b>Type</b>	Text String
<b>Value</b>	<p>Specifies the location of the disk container. It can be either the name of a ".vdisk" file or the name of a physical disk:</p> <ul style="list-style-type: none"> <li>• <b>Local fixed disks (IDE, SCSI, SATA)</b> <ul style="list-style-type: none"> <li>• "\\.\PhysicalDrive&lt;N&gt;"</li> <li>• "\\.\PhysicalDrive(DevID =XXX-XXXX-XXXX-XXXX-XXXX-XXXX-XXXX-XXXX)"</li> </ul> <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><b>Example:</b></p> <pre>set DUA container[3]="\\.\PhysicalDrive(DevID= 6008-05F3-0005-2950-BF8E-0B86-A0C7-0001) "</pre> <p>The WWID values can be obtained from "<a href="#">Host Device Check</a>" utility ("All drives" section).</p> <p> If the "Host Device Check" utility does not display any WWID, this means the target disk does not have one. Use the "\\.\PhysicalDrive&lt;N&gt;" mapping in this case.</p> </li> <li>• <b>iSCSI disks</b> <ul style="list-style-type: none"> <li>• "\\.\PhysicalDrive(iScsiTarget = &lt;iSCSI target&gt;, LUN = &lt;LUN number&gt;)"</li> </ul> <p>iScsiTarget addresses the disk by its iSCSI target name. LUN specifies LUN on connected iSCSI disk.</p> <p><b>Example:</b></p> <pre>set DUA container[0]="\\.\PhysicalDrive(iScsiTarget= iqn.2008-04:iscsi.charon-target-test1, LUN= 1) "</pre> </li> <li>• <b>Floppy drives</b> <ul style="list-style-type: none"> <li>• "\\.\A:"</li> <li>• "\\.\B:"</li> </ul> </li> <li>• <b>CD-ROM, DVD drives (IDE, SCSI, ...)</b> <ul style="list-style-type: none"> <li>• "\\.\CdRom&lt;N&gt;"</li> </ul> </li> </ul>


## media\_type

<b>Parameter</b>	media_type[N] N=0...9999
<b>Type</b>	Text String
<b>Value</b>	<p>Overrides the default (automatically determined) MSCP media type of a device.</p> <p><b>Syntax:</b></p> <pre>"&lt;device-name&gt;[ , &lt;device-type&gt;]"</pre> <p>where:</p> <ul style="list-style-type: none"> <li>• &lt;device-name&gt; is one of "DU", "DK", "SCSI", "DI", "DSSI", "DJ"</li> <li>• &lt;device-type&gt; is of the form "LLD" or "LLLD", where "L" is a letter from A through Z, and "D" is a decimal number from 0 through 99</li> </ul> <p>If not specified, the device name is set to "DU" and the device type is selected based on disk size.</p> <p>Initially not specified.</p>

## geometry

<b>Parameter</b>	geometry[N] N=0...9999
<b>Type</b>	Text String
<b>Value</b>	<p>This formatted string value specifies the explicit geometry of the disk storage element with a DSSI node id: N and an MSCP unit number: N. This parameter is not applicable to tape storage elements.</p> <p>The string format is &lt;X&gt;"/"&lt;Y&gt;["/"&lt;Z&gt;] where:</p> <ol style="list-style-type: none"> <li>1. X is the number of sectors per track;</li> <li>2. Y is the number of tracks per cylinder;</li> <li>3. Z (optional) is 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;</li> </ol> <p>If this parameter is not set, CHARON-VAX / CHARON-PDP will configure the geometry based on the most probable disk type.</p>

## use\_io\_file\_buffering

<b>Parameter</b>	use_io_file_buffering[N] N=0...9999
<b>Type</b>	Text String
<b>Value</b>	<p>Instructs CHARON-VAX to enable host operating system I/O cache on reading/writing operations.</p> <p>The following values are possible:</p> <ul style="list-style-type: none"> <li>• "true" or "write-back" - the "write back" caching mode is used</li> <li>• "write-through" - the "write through" caching mode is used</li> <li>• "false" - caching is switched off</li> </ul> <p> Note that this caching has a significant effect only in case of mapping to disk and tape containers, not physical drives.</p> <p>When enabled, host operating system I/O cache may significantly improve I/O performance of the virtual system. At the same time maintaining I/O cache requires additional host resources (CPU and memory) which may negatively affect overall performance of the virtual system.</p> <p>Initially is set to "false".</p> <p><b>Example:</b></p> <pre>set DUA use_io_file_buffering[2]=true</pre>

### Example 1:

```
load RQDX3/RQDX3 DUA address=017772150 max_n_of_units=4
set DUA container[0] = "C:\Charon\Disks\rx23.vdisk"
set DUA container[1] = "\\.\PhysicalDrive1"

load RQDX3/RQDX3 DUB address=017760334
set DUB container[5] = "\\.\CdRom0"
```

In the above example the "rx23.vdisk" will be seen in VMS as DUA0, "\\.\PhysicalDrive1" as DUA1 and "\\.\CdRom0" as DUB5.

### Example 2:

```
load RQDX3/RQDX3 DIA address=017772150 max_n_of_units=4
set DIA container[0] = "C:\Charon\Disks\rx23.vdisk"
set DIA media_type[0] = "dssi"

set DIA container[1] = "\\.\PhysicalDrive1"
set DIA media_type[1] = "dssi"
```

In the above example the "rx23.vdisk" will be seen in VMS as DIA0 and "\\.\PhysicalDrive1" as DIA1.



## TQK50 and TUK50 Controllers

The CHARON-VAX / CHARON-PDP QBUS system provides support for the TQK50 tape controller. UNIBUS systems support the TUK50 tape controller.

The original TQK50/TUK50 tape controllers were capable of serving only 1 tape unit. CHARON-VAX / CHARON-PDP extends the limit to 10000 tape units.

Use the following commands to load an instance of a TQK50/TUK50 tape controller:

```
load TQK50/TQK50 <logical name 1>
load TUK50/TUK50 <logical name 2>
```

### Example:

```
load TQK50/TQK50 MUA1
load TUK50/TUK50 MUA2
```

The TQK50/TUK50 controllers have the following configuration parameters, which are specified with the "set" command:

### address

<b>Parameter</b>	address
<b>Type</b>	Numeric
<b>Value</b>	<p>Specifies the CSR address. This address must be a valid QBUS 22-bit address in IO space for a TQK50 and a valid UniBUS 18-bit address in I/O space for a TUK50.</p> <p>The initial values are 017774500 (TQK50) and 0774500 (TUK50), which is the factory setting for these tape controllers.</p> <p>Use the "address" parameter to load several instances of TQK50's/TUK50's. The "address" parameter value must be unique for each instance of a TQK50/TUK50.</p>

### container

<b>Parameter</b>	container[N] N=0...9999
<b>Type</b>	Text String
<b>Value</b>	<p>Specifies the location of the tape container. It can be either the name of a ".vtape" (".mtd") file or the name of a physical tape drive:</p> <ul style="list-style-type: none"> <li>• <b>Tape images</b> <ul style="list-style-type: none"> <li>• &lt;file name&gt;.vtape</li> </ul> </li> <li>• <b>Local physical tape drives recognized by Windows</b> <ul style="list-style-type: none"> <li>• "\\Tape&lt;N&gt;"</li> </ul> </li> </ul>

## media\_type

<b>Parameter</b>	media_type[N] N=0...9999
<b>Type</b>	Text String
<b>Value</b>	<p>Overrides the default (automatically determined) TMSCP media type of a device.</p> <p><b>Syntax:</b></p> <pre>"&lt;device-name&gt;[ , &lt;device-type&gt;]"</pre> <p>where:</p> <ul style="list-style-type: none"> <li>• &lt;device-name&gt; is either: "MU", "MK", "SCSI", "MI", "DSSI", or "MJ"</li> <li>• &lt;device-type&gt; is of the form "LLD" or "LLLD", where "L" is letter a from A through Z, and "D" is a decimal number from 0 through 99</li> </ul> <p>If not specified, the device name is set to "MU" and the device type is set to "TK50"</p> <p>Initially not specified.</p>

## geometry

<b>Parameter</b>	geometry[N] N=0...9999
<b>Type</b>	Text String
<b>Value</b>	<p>Specifies the size of a tape image and (optionally) the size of an "early-warning" area at the end of a tape image.</p> <p><b>Syntax:</b></p> <pre>"&lt;image-size&gt;[ , &lt;early-warning-zone-size&gt;]"</pre> <p>where:</p> <ul style="list-style-type: none"> <li>• &lt;image-size&gt; is the tape size in MB</li> <li>• &lt;early-warning-zone-size&gt; is the size (in KB) of the space left on the tape that, when reached, issues a warning to the OS. If omitted, 64K is assumed.</li> </ul> <p><b>Example:</b></p> <pre>load TQK50 MUA set MUA geometry[0] = 90</pre>

### Example:

```
load TQK50 MUA address=017774500
set MUA container[0] = "\\.\Tape0"
set MUA container[1] = "C:\Charon\Tapes\tape1.vtape"
```

Multi-volume tape images are handled as follows:

```
set MUA container[0] = "..."  
set MUA container[1] = "..."  
set MUA container[2] = "..."  
set MUA container[3] = "..."
```

Once this configuration is established, the following VMS command (for example) can be used:

```
$ BACKUP MUA0:BACKUP.BCK,MUA1,MUA2,MUA3/SAVE_SET DUA0:...
```

## KDM70 Controller

KDM70 is an MSCP/TMSCP disk and tape storage controller for a VAX 6000.

The CHARON-VAX virtual KDM70 storage controller supports 9999 disks and tapes instead of the 8 disk limitation of the original hardware. This design modification has the advantage of using only one XMI slot for up to 9999 disk and tape devices.

The I/O behavior of the virtual KDM70 storage controller is as follows:

- Up to 16 connected disks operate in parallel without any I/O performance degradation.
- For systems with more than 16 heavily used disks, configure two controllers and distribute the heavily loaded disks evenly.
- As in the hardware KDM70 storage controller, VMS can be booted only from the first 10 devices (DU0 - DU9) on the KDM70 storage controller.
- Hardware KDM70 storage controllers do not support tape drives. The virtual KDM70 storage controllers support a transparent extension for data tapes (boot from tape is not supported).

The line below loads an emulated KDM70 storage controller:


```
load KDM70/KDM70 PUA
```

The KDM70 storage controller emulation has the following configuration parameters:

### xmi\_node\_id

<b>Parameter</b>	xmi_node_id
<b>Type</b>	Numeric
<b>Value</b>	Specifies the XMI slot in which the virtual KDM70 storage controller is placed. For CHARON-VAX/66X0 a free slot between 10 (A) and 14 (E) must be chosen.

## container

<b>Parameter</b>	container[N] N=0...9999
<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 drives</b> <ul style="list-style-type: none"> <li>"\\.\PhysicalDrive&lt;X&gt;", where X is 0, 1, ...</li> <li>"\\.\PhysicalDrive(DevID =XXXX-XXXX-XXXX-XXXX-XXXX-XXXX-XXXX-XXXX)"</li> </ul> <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><b>Example:</b></p> <pre>set PUA container[3]="\\.\PhysicalDrive(DevID= 6008-05F3-0005-2950-BF8E-0B86-A0C7-0001) "</pre> <p>The WWID values can be obtained from "Host Device Check" utility ("All drives" section).</p> <p> If the "Host Device Check" utility does not display any WWID, this means the target disk does not have one. Use the "\\.\PhysicalDrive&lt;N&gt;" mapping in this case.</p> </li> <li> <b>iSCSI disks</b> <ul style="list-style-type: none"> <li>"\\.\PhysicalDrive(iScsiTarget = &lt;iSCSI target&gt;, LUN = &lt;LUN number&gt;)"</li> </ul> <p>iScsiTarget addresses the disk by its iSCSI target name. LUN specifies LUN on connected iSCSI disk.</p> <p><b>Example:</b></p> <pre>set PUA container[0]="\\.\PhysicalDrive(iScsiTarget= iqn.2008-04:iscsi.charon-target-test1, LUN= 1) "</pre> </li> <li> <b>Physical tape drives recognized by Windows</b> <ul style="list-style-type: none"> <li>"\\.\Tape&lt;X&gt;", where X is 0, 1, ...</li> </ul> </li> <li> <b>Floppy drives</b> <ul style="list-style-type: none"> <li>"\\.\A:"</li> <li>"\\.\B:"</li> </ul> </li> <li> <b>CD-ROM drives (read-only)</b> <ul style="list-style-type: none"> <li>"\\.\CdRom&lt;X&gt;", where X is 0, 1, ...</li> </ul> </li> <li> <b>CHARON-VAX disk images</b> <ul style="list-style-type: none"> <li>[&lt;drive&gt;:\&lt;path-name&gt;\&lt;file-name&gt;[".vdisk"]</li> </ul> </li> <li> <b>CHARON-VAX tape images</b> <ul style="list-style-type: none"> <li>[&lt;drive&gt;:\&lt;path-name&gt;\&lt;file-name&gt;".vtape"</li> </ul> </li> </ul>


## media\_type

<b>Parameter</b>	media_type[N] N=0...9999
<b>Type</b>	Text String
<b>Value</b>	<p>Overrides the default (automatically determined) MSCP media type of the device.</p> <p><b>Syntax:</b></p> <pre>"&lt;device-name&gt; , &lt;device-type&gt; "</pre> <p>where:</p> <ul style="list-style-type: none"> <li>• &lt;device-name&gt; is either: "DU", "MU", "DK", "MK", "SCSI", "DI", "MI", "DSSI", "DJ", "MJ"</li> <li>• &lt;device-type&gt; is in the form: "LLD" or "LLLD", where "L" is a letter from A through Z, and "D" is a decimal number from 0 through 99</li> </ul> <p>If not specified, the device name is set to "DI", and the device type is selected based on disk size for the disk storage elements. For tape storage elements, the device name and type are set to "MI" and "TF86", respectively.</p> <p>Initially not specified.</p>

## geometry

<b>Parameter</b>	geometry[N] N=0...9999
<b>Type</b>	Text String
<b>Value</b>	<p>This formatted string value specifies the explicit geometry of the disk storage element with DSSI node id: N and MSCP unit number: N. This parameter is not applicable to tape storage elements.</p> <p>The string format is &lt;X&gt;"/"&lt;Y&gt;["/"&lt;Z&gt;] where:</p> <ul style="list-style-type: none"> <li>• X is the number of sectors per track</li> <li>• Y is the number of tracks per cylinder</li> <li>• Z (optional) is 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</li> </ul> <p>If this parameter is not set, CHARON-VAX will configure the geometry based on the most probable disk type.</p> <p>Initially not set.</p> <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><b>Syntax:</b></p> <pre>"&lt;image-size&gt;[ , &lt;early-warning-zone-size&gt; ]"</pre> <p>where:</p> <ul style="list-style-type: none"> <li>• &lt;image-size&gt; is the tape size in MB</li> <li>• &lt;early-warning-zone-size&gt; is the size (in KB) of the space left on the tape when a warning to the OS is issued. If omitted, 64K is assumed.</li> </ul>

## use\_io\_file\_buffering

<b>Parameter</b>	use_io_file_buffering[N] N=0...9999
<b>Type</b>	Text String
<b>Value</b>	<p>Instructs CHARON-VAX to enable host operating system I/O cache on reading/writing operations.</p> <p>The following values are possible:</p> <ul style="list-style-type: none"> <li>"true" or "write-back" - the "write back" caching mode is used</li> <li>"write-through" - the "write through" caching mode is used</li> <li>"false" - caching is switched off</li> </ul> <p> Note that this caching has a significant effect only in case of mapping to disk and tape containers, not physical drives.</p> <p>When enabled, host operating system I/O cache may significantly improve I/O performance of the virtual system. At the same time maintaining I/O cache requires additional host resources (CPU and memory) which may negatively affect overall performance of the virtual system.</p> <p>Initially is set to "false".</p> <p><b>Example:</b></p> <pre>set PUA use_io_file_buffering[0]=true</pre>

### Example:

Create a KDM70 storage controller: (T)MSCP controller in XMI slot 10:

```
load KDM70/KDM70 PUA xmi_node_id=10
```

Configure, on this controller, a system disk to be DUA0: in VMS:

```
set PUA container[0]="C:\Charon\Disks\vms72-66X0.vdisk"
```

Configure a user disk to be DUA1: in VMS:

```
set PUA container[1]="C:\Charon\Disks\usertest.vdisk"
```

Configure the first SCSI tape drive connected to a host to be MUA4: in VMS:

```
set PUA container[4]="\\.\Tape0"
```

The file my\_tape.vtape in the default directory is used by VMS as MUA5:


```
set PUA container[5]="C:\Charon\Tapes\my_tape.vtape"
```

The first host system CD-ROM is used to read VMS CDs and be DUA9:

```
set PUA container[9]="\\.\CdRom0"
```

The host system floppy drive "A:" is used in VMS as DUA10:

```
set PUA container[10]="\\.\A:"
```

-  The virtual KDM70 storage controller examines the file extension (`vdiskor 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 VAX/VMS 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.

## KDB50 Storage Controller

KDB50 is an MSCP storage controller for a VAX 6000.

The CHARON-VAX virtual KDB50 storage controller supports up to 9999 disks instead of the 4 disk limitation of the original hardware. This design modification has the advantage of using only one *VAXB/* slot for up to 9999 disk and tape devices. The I/O behavior of the virtual KDB50 storage controller is as follows:

- Up to 16 connected disks operate in parallel without any I/O performance degradation.
- For systems with more than 16 heavily used disks, configure two storage controllers and distribute the heavily loaded disks evenly.
- VMS can boot only from the first 10 devices (DU0 - DU9) on the KDB50 storage controller, same as for the original hardware.

The line below loads an emulated KDB50 storage controller:


```
load KDB50 PUA
```

The KDB50 storage controller emulation has the following configuration parameters:

### vax\_bi\_node\_id

<b>Parameter</b>	vax_bi_node_id
<b>Type</b>	Numeric
<b>Value</b>	Specifies the VAXB/ slot in which the virtual KDB50 storage controller is placed. For CHARON-VAX / CHARON-PDP a free slot between 1 (1) and 15 (F) must be chosen.  Initially set to 14.

## container

<b>Parameter</b>	container[N] N=0...9999
<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 drives</b> <ul style="list-style-type: none"> <li>• "\\.\PhysicalDrive&lt;X&gt;", where X is 0, 1, ...</li> <li>• "\\.\PhysicalDrive(DevID =XXXX-XXXX-XXXX-XXXX-XXXX-XXXX-XXXX-XXXX)"</li> </ul> <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><b>Example:</b></p> <pre>set PUA container[3]="\\.\PhysicalDrive(DevID= 6008-05F3-0005-2950-BF8E-0B86-A0C7-0001)"</pre> <p>The WWID values can be obtained from "<a href="#">Host Device Check</a>" utility ("All drives" section).</p> <p> If the "Host Device Check" utility does not display any WWID, this means the target disk does not have one. Use the "\\.\PhysicalDrive&lt;N&gt;" mapping in this case.</p> </li> <li>• <b>iSCSI disks</b> <ul style="list-style-type: none"> <li>• "\\.\PhysicalDrive(iScsiTarget = &lt;iSCSI target&gt;, LUN = &lt;LUN number&gt;)"</li> </ul> <p>iScsiTarget addresses the disk by its iSCSI target name. LUN specifies LUN on connected iSCSI disk.</p> <p><b>Example:</b></p> <pre>set PUA container[0]="\\.\PhysicalDrive(iScsiTarget= iqn.2008-04:iscsi.charon-target-test1, LUN= 1)"</pre> </li> <li>• <b>Floppy drives</b> <ul style="list-style-type: none"> <li>• "\\.\A:"</li> <li>• "\\.\B:"</li> </ul> </li> <li>• <b>CD-ROM drives (read-only)</b> <ul style="list-style-type: none"> <li>• "\\.\CdRom&lt;X&gt;", where X is 0, 1, ...</li> </ul> </li> <li>• <b>CHARON-VAX disk images</b> <ul style="list-style-type: none"> <li>• [&lt;drive&gt;:\&lt;path-name&gt;\&lt;file-name&gt;[".vdisk"]</li> </ul> </li> </ul> <p>This parameter is initially not set, thus creating NO storage elements on the storage controller</p>




## media\_type

<b>Parameter</b>	media_type[N] N=0...9999
<b>Type</b>	Text String
<b>Value</b>	<p>Overrides the default (automatically determined) MSCP media type of the device.</p> <p><b>Syntax:</b></p> <p>"&lt;device-name&gt; , &lt;device-type&gt; "</p> <p>where:</p> <ul style="list-style-type: none"> <li>• &lt;device-name&gt; is either: "DU", "DK", "SCSI", "DI", "DSSI" or "DJ"</li> <li>• &lt;device-type&gt; is in the form of: "LLD" or "LLLD", where "L" is a letter from A through Z, and "D" is a decimal number from 0 through 99</li> </ul> <p>If not specified, the device name is set to "DU" and the device type is selected, based on the disk size for the disk storage elements. For tape storage elements, the device name and type are set to "MI" and "TF86", respectively.</p> <p>Initially not specified.</p>

## geometry

<b>Parameter</b>	geometry[N] N=0...9999
<b>Type</b>	Text String
<b>Value</b>	<p>This formatted string value specifies the explicit geometry of the disk storage element with MSCP unit number: N. This parameter is not applicable to tape storage elements.</p> <p>The string format is &lt;X&gt;"/"&lt;Y&gt;["/"&lt;Z&gt;] where:</p> <ul style="list-style-type: none"> <li>• X is the number of sectors per track</li> <li>• Y is the number of tracks per cylinder</li> <li>• Z (optional) is 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</li> </ul> <p>If this parameter is not set, CHARON-VAX / CHARON-PDP will configure the geometry based on the most probable disk type.</p> <p>Initially not set.</p>

## use\_io\_file\_buffering

<b>Parameter</b>	use_io_file_buffering[N] N=0...9999
<b>Type</b>	Text String
<b>Value</b>	<p>Instructs CHARON-VAX to enable host operating system I/O cache on reading/writing operations.</p> <p>The following values are possible:</p> <ul style="list-style-type: none"> <li>• "true" or "write-back" - the "write back" caching mode is used</li> <li>• "write-through" - the "write through" caching mode is used</li> <li>• "false" - caching is switched off</li> </ul> <p> Note that this caching has a significant effect only in case of mapping to disk and tape containers, not physical drives.</p> <p>When enabled, host operating system I/O cache may significantly improve I/O performance of the virtual system. At the same time maintaining I/O cache requires additional host resources (CPU and memory) which may negatively affect overall performance of the virtual system.</p> <p>Initially is set to "false".</p> <p><b>Example:</b></p> <pre>set PUA use_io_file_buffering[0]=true</pre>

### Example:

Create a KDB50 MSCP controller in BI slot 1 of a VAX/PDP11:

```
load KDB50/KDB50 PUA vax_bi_node_id=1
```

Configure on this controller a system disk to be DUA0: in VMS:

```
set PUA container[0]="C:\Charon\Disks\vms72-66X0.vdisk"
```

Configure a user disk to be DUA1: in VMS:


```
set PUA container[1]="C:\Charon\Disks\usertest.vdisk"
```

The first host system CD-ROM can be used to read VMS CDs, named DUA9:

```
set PUA container[9]="\\.\CdRom0"
```

The host system floppy drive `A:` can be used in VMS, named DUA10:

```
set PUA container[10]="\\.\A:"
```

 Configured physical devices or tape/disk images that do not exist on the host system will, in general, cause VAX/VMS 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.

# SCSI Controllers

## Table of Contents

- Introduction
- Mapping to host resources
  - [scsi\\_bus](#)
  - [scsi\\_id](#)
  - [virtual\\_scsi\\_disk](#)
    - [container](#)
    - [media\\_type](#)
    - [geometry](#)
    - [use\\_io\\_file\\_buffering](#)
    - [removable](#)
  - [floppy\\_scsi\\_device](#)
    - [container](#)
  - [virtual\\_scsi\\_tape](#)
    - [container](#)
    - [media\\_type](#)
    - [geometry](#)
  - [virtual\\_scsi\\_cdrom](#)
    - [container](#)
    - [media\\_type](#)
    - [geometry](#)
    - [use\\_io\\_file\\_buffering](#)
  - [physical\\_scsi\\_device](#)
    - [container](#)
    - [media\\_type](#)
    - [geometry](#)
    - [removable](#)
    - [disconnect\\_timeout](#)

## Introduction

CHARON-VAX provides two SCSI controllers for the SCSI and SCSI/QBUS models of VAX.

Hardware disks, disk images, hardware tapes, tape images, floppy devices and CD-ROM devices can be connected to these SCSI controllers. Each device has to be configured to connect to a specific SCSI address in CHARON-VAX.

Use the following emulated device types to map real peripherals to the emulated SCSI devices:

Type of mapping	Description
<a href="#">virtual_scsi_disk</a>	For disk image containers and physical disks
<a href="#">floppy_scsi_device</a>	For floppy drives
<a href="#">virtual_scsi_tape</a>	For tape image containers
<a href="#">virtual_scsi_cdrom</a>	For host CD-ROM and *.iso images
<a href="#">physical_scsi_device</a>	For physical tapes and other SCSI devices connected to a host

CHARON-VAX disk/tape devices can be SCSI disk/tape devices connected to a host system or disk/tape containers that are presented to the operating system environment as files.

Two SCSI controllers are provided ("PKA" and "PKB") in CHARON-VAX, with 7 addresses each.

Beyond the capabilities of the hardware, VAX 3100/9x and 4000/10x, CHARON-VAX/XX implements extended SCSI addressing. Each of the seven device addresses, of a SCSI controller, supports up to eight disk/tape images. Thus the number of disks supported becomes 2x Controllers\*7 addresses\*8 Disks/Tapes, a total of 112 disks/tapes.

SCSI devices with the same ID but different LUNs (logical units) appear in the VAX console with different names. The naming convention is as follows:


Each SCSI device has the name in the form of "xKct0n:", where:

- "x" stands for the device type (D means disks, M means magnetic tapes, G is reserved by VAX/VMS for special purposes)
- "c" stands for the controller letter (A - the first controller, B - the second controller, ...)
- "t" stands for the SCSI device ID (usually 0 through 6, and 7 is allocated by the controller itself)
- "n" stands for a particular logical unit number, LUN.


Most of the 'normal' SCSI devices have only one logical unit - 0. Therefore, under normal conditions, disks in VAX/VMS appear as DKA0 (which is really DKA000), DKA100, DKA200, ..., tapes as MKA0 (which is really MKA000), MKA100, MKA200, ...

As soon as there is a disk/tape device with LUNs 0 and 1, VMS identifies them as, for example, DKA300 and DKA301 (MKA300 and MKA301) respectively.

The boot ROM of CHARON-VAX detects SCSI devices with multiple LUNs and builds proper device names for them.

 To display a list of devices on the VAX console (SRM), enter ">>> show device".

This list is passed to VAX/VMS at boot time.

 VAX/VMS creates devices only for logical unit 0 for each device detected in the boot ROM. To add additional logical units, use the following SYSGEN command:

```
$ MCR SYSGEN CONNECT DKxxx/NOADAPTER
```

where DKxxx (or MKxxx) stands for the correct VAX/VMS name of the logical unit to be connected. You can find its name from the SRM console using the ">>> show scsi" command.

This command is not boot persistent, so it must be included in the VAX/VMS "SYSTARTUP\_VMS.COM" file to ensure it is executed with each startup.

Also note: that the following rules are applied for logical units.

1. Each SCSI device must implement logical unit 0.
2. A SCSI device must implement all logical unit numbers between the highest and the lowest numbers implemented.

Empty disk images can be created with the "MkDisk" utility.

CHARON-VAX is able to boot from disk images of any VAX/VMS version, starting with 4.5 or higher for MicroVAX II or VAX 3600 and VMS 5.5-2 or higher for the VAX4000.

## Mapping to host resources

Load a mapping device with the "load" command. Specify the name of the device instance, the emulated SCSI bus to connect the device to and the SCSI identifier of the CHARON-VAX device.

### scsi\_bus

<b>Parameter</b>	scsi_bus
<b>Type</b>	Identifier
<b>Value</b>	Name of the emulated SCSI disk controller: "pka" or "pkb"

## scsi\_id

<b>Parameter</b>	scsi_id
<b>Type</b>	Numeric
<b>Value</b>	A value between 0 and 7. This is the ID number of the emulated SCSI device. The SCSI adapter is preloaded with address 7. If required, set it to another value in the range of 0-7 from the VAX console.



### Syntax:

```
load <instance type>/<module name> <instance name> scsi_bus=<bus name> scsi_id=<number>
```

### Example:

```
load virtual_scsi_disk/chscsi pka_0 scsi_bus=pka scsi_id=0
```

CHARON-VAX/XX has only one preloaded SCSI adapter, named: PKA. If a second adapter (PKB) is required then add the following line to the configuration file before loading and configuring any device on the second SCSI adapter PKB:

```
include kzdda.icfg
```

"kzdda.icfg" loads the second SCSI adapter.



OpenVMS version 5.5-2H4, or above, is required to use the "pkb" controller.

## virtual\_scsi\_disk

Use the "virtual\_scsi\_disk" mapping for disk containers and physical disks. This is the most convenient way of connecting disks to SCSI adapters in CHARON-VAX

The "virtual\_scsi\_disk" mapping has the following parameters:

## container

<b>Parameter</b>	container[N] N=0...7
<b>Type</b>	Text String
<b>Value</b>	<p>A string containing the device name to map to the emulator.</p> <ul style="list-style-type: none"> <li> <b>Disk images</b> <ul style="list-style-type: none"> <li> <code>[&lt;path-name&gt;"\"]&lt;file-name&gt;[".vdisk"]</code> <p>Format: A string containing the full path to a disk container.</p> <p>If only the name of the disk container is specified, CHARON-VAX will look for the container in the folder where the emulator executable resides, typically</p> <p>"C:\Program Files (x86)\CHARON\Build_XXXXX\x86" (Windows x64; for Windows x86 the path is "C:\Program Files\CHARON\Build_XXXXX\x86").</p> <p><b>Using compressed folders to store virtual disks and tapes is not supported.</b></p> </li> </ul> </li> <li> <b>Local fixed disks (IDE, SATA)</b> <ul style="list-style-type: none"> <li> <code>"\\.\PhysicalDrive&lt;X&gt;"</code>, where X is 0, 1, ... </li> <li> <code>"\\.\PhysicalDrive(DevID =XXXX-XXXX-XXXX-XXXX-XXXX-XXXX-XXXX-XXXX)"</code> <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><b>Example:</b></p> <pre>set pka_1 container="\\.\PhysicalDrive(DevID= 6008-05F3-0005-2950-BF8E-0B86-A0C7-0001) "</pre> <p>The WWID values can be obtained from "<a href="#">Host Device Check</a>" utility ("All drives" section).</p> <p>If the "Host Device Check" utility does not display any WWID, this means the target disk does not have one. Use the "\\.\PhysicalDrive&lt;N&gt;" mapping in this case.</p> </li> </ul> </li> <li> <b>iSCSI disks</b> <ul style="list-style-type: none"> <li> <code>"\\.\PhysicalDrive(iScsiTarget = &lt;iSCSI target&gt;, LUN = &lt;LUN number&gt;)"</code> <p>iScsiTarget addresses the disk by its iSCSI target name. LUN specifies LUN on connected iSCSI disk.</p> <p><b>Example:</b></p> <pre>set pka_2 container="\\.\PhysicalDrive(iScsiTarget= iqn.2008-04:iscsi.charon-target-test1, LUN= 1) "</pre> </li> </ul> </li> </ul> <p>This parameter is initially not set, thus creating NO storage elements on the controller</p>


## media\_type

<b>Parameter</b>	media_type[N] N=0...7
<b>Type</b>	Text String
<b>Value</b>	Overrides the PRODUCT ID in the default SCSI INQUIRY data.  Valid values may contain uppercase letters, integers and spaces. The length of the string cannot exceed 16 characters.  If not specified, synthetic SCSI INQUIRY data is returned containing a PRODUCT ID selected based on the disk size.  Initially left unspecified.

## geometry


<b>Parameter</b>	geometry[N] N=0...7
<b>Type</b>	Text String
<b>Value</b>	This formatted string value specifies the explicit geometry of the disk storage element  The string format is <X>"/"<Y>["/"<Z>] where: <ul style="list-style-type: none"> <li>• "X" is the number of sectors per track;</li> <li>• "Y" is the number of tracks per cylinder;</li> <li>• "Z" (optional) is 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;</li> </ul> <p>If this parameter is not set, CHARON-VAX will configure the geometry based on the most probable disk type.</p>

## use\_io\_file\_buffering

<b>Parameter</b>	use_io_file_buffering[N] N=0...7
<b>Type</b>	Text String
<b>Value</b>	Instructs CHARON-VAX to enable host operating system I/O cache on reading/writing operations.  The following values are possible: <ul style="list-style-type: none"> <li>• "true" or "write-back" - the "write back" caching mode is used</li> <li>• "write-through" - the "write through" caching mode is used</li> <li>• "false" - caching is switched off</li> </ul> <p> Note that this caching has a significant effect only in case of mapping to disk containers, not physical drives.</p> <p>When enabled, host operating system I/O cache may significantly improve I/O performance of the virtual system. At the same time maintaining I/O cache requires additional host resources (CPU and memory) which may negatively affect overall performance of the virtual system.</p> <p>Initially is set to "false".</p> <p><b>Example:</b></p> <pre>set pka_0 use_io_file_buffering[0]=true</pre>

## removable

<b>Parameter</b>	removable[N] N=0...7
<b>Type</b>	boolean
<b>Value</b>	Enables the logical unit to appear as a removable SCSI disk drive. Initially set to "false" (fixed, non-removable).

 In the table above N stands for logical unit number. The first unit must be 0 with no gaps in subsequent numbering.

**Example:**

```
load virtual_scsi_disk/chscsi_pka_0 scsi_bus=pka scsi_id=0
set pka_0 container[0] = "C:\Charon\Disks\disk1.vdisk"
set pka_0 container[1] = "\\.\PhysicalDrive1"
```

If only one LUN is configured, the LUN number can be omitted:

```
set pka_0 container = "C:\Charon\Disks\disk1.vdisk"
set pka_0 media_type = "RZ1ED"
```

When a virtual SCSI disk image is dismounted in VMS, it is no longer open by CHARON and may be copied. This capability can be useful when designing back-up and restore procedures. If copying CHARON-VAX disk images while CHARON-VAX is running, take care to minimize the risk of overloading the host system.

Unlike MSCP controlled disk images, **a disk image connected to a SCSI controller as a virtual SCSI disk CANNOT be replaced by another disk image** unless "removable" parameter is set for this particular disk image.

**Example:**

```
set pka_0 container[2] = "C:\Charon\Disks\my_removable_disk.vdisk"
set pka_0 removable[2] = true
```


## floppy\_scsi\_device

Use "floppy\_scsi\_device" for host floppy drives.

"floppy\_scsi\_device" has the following parameter:

## container

<b>Parameter</b>	container[N] N=0...7
<b>Type</b>	Text String
<b>Value</b>	"\\.\A:" or "\\.\B:"

 In the table above N stands for logical unit number. The first unit must be 0 with no gaps in subsequent numbering. If only one LUN is configured, the LUN number can be omitted:



**Example:**

```
load floppy_scsi_device/chscsi pka_0 scsi_bus=pka scsi_id=0
set pka_0 container= "\\.\A:"
```

## virtual\_scsi\_tape

Use "virtual\_scsi\_tape" for tape containers. This is the most convenient way of connecting tapes to SCSI adapters in CHARON-VAX.

"virtual\_scsi\_tape" has the following parameters:

### container

<b>Parameter</b>	container[N] N=0...7
<b>Type</b>	Text String
<b>Value</b>	<p>A string containing the full path to a tape container. If the specified tape image does not exist, CHARON-VAX creates it.</p> <p>If only the name of the tape container is specified, CHARON-VAX will look for the container in the folder where the emulator executable resides, typically "C:\Program Files (x86)\CHARON\Build_XXXXX\x86" (Windows x64; for Windows x86 the path is "C:\Program Files\CHARON\Build_XXXXX\x86").</p> <p>If the "<a href="#">CHARON Guest Utilities for OpenVMS</a>" (CHARONCP) package is used, the syntax is different. Please read the corresponding chapter.</p> <p><b>Using compressed folders to store virtual disks and tapes is not supported.</b></p>

### media\_type

<b>Parameter</b>	media_type[N] N=0...7
<b>Type</b>	Text String
<b>Value</b>	<p>Overrides PRODUCT ID in the default SCSI INQUIRY data.</p> <p>Valid values may contain uppercase letters, integers and spaces. Length of this string cannot exceed 16 characters.</p> <p>By default the PRODUCT ID returned is "TLZ04"</p>

## geometry

<b>Parameter</b>	geometry[N] N=0...7
<b>Type</b>	Text String
<b>Value</b>	<p>Specifies the size of the tape image and (optionally) the size of the "early-warning" area at the end of the tape image.</p> <p><b>Syntax:</b></p> <pre>"&lt;image-size&gt;[ , &lt;early-warning-zone-size&gt;]"</pre> <p>where:</p> <ul style="list-style-type: none"> <li>• &lt;image-size&gt; is the tape size in MB</li> <li>• &lt;early-warning-zone-size&gt; is the size (in KB) of the space remaining on the tape when a warning to the OS is issued. If omitted, 64K is assumed.</li> </ul> <p><b>Example:</b></p> <pre>load virtual_scsi_tape/chscsi pka_0 set pka_0 geometry[0] = 90</pre>

⚠ In the table above N stands for logical unit number. The first unit must be 0 with no gaps in subsequent numbering.

### Example:

```
load virtual_scsi_tape/chscsi pka_0 scsi_bus=pka scsi_id=0
set pka_0 container[0] = "C:\Charon\Tapes\tape1.vtape"
set pka_0 container[1] = "C:\Charon\Tapes\tape2.vtape"
```

If only one LUN is configured, the LUN number can be omitted:

```
set pka_0 container = "C:\Charon\Tapes\tape1.vtape"
set pka_0 media_type = "TLZ08"
```

## virtual\_scsi\_cdrom

Use "virtual\_scsi\_cdrom" to connect IDE or SATA host CD-ROM or \*.iso file to CHARON-VAX.

"virtual\_scsi\_cdrom" has the following parameters:

## container

<b>Parameter</b>	container[N] N=0...7
<b>Type</b>	Text String
<b>Value</b>	<p>A string containing the device name to map to the emulator.</p> <ul style="list-style-type: none"> <li>• <b>ISO images</b> <ul style="list-style-type: none"> <li>• [<code>&lt;path-name&gt;"\"&lt;file-name&gt;".iso"</code>]</li> </ul> <p>Format: A string containing the full path to an *.iso image.</p> <p>If only the name of the *.iso image is specified, CHARON-VAX will look for the image in the folder where the emulator executable resides, typically</p> <p>"C:\Program Files (x86)\CHARON\Build_XXXXX\x86" (Windows x64; for Windows x86 the path is "C:\Program Files\CHARON\Build_XXXXX\x86").</p> </li> <li>• <b>CD-ROM and DVD drives (IDE and SATA)</b> <ul style="list-style-type: none"> <li>• "\\.\CdRom&lt;X&gt;", where X is 0, 1, ...</li> </ul> </li> </ul> <p>This parameter is initially not set, thus creating NO storage elements on the controller</p>


## media\_type


<b>Parameter</b>	media_type[N] N=0...7
<b>Type</b>	Text String
<b>Value</b>	<p>Overrides the PRODUCT ID in the default SCSI INQUIRY data.</p> <p>Valid values may contain uppercase letters, integers and spaces. The length of the string cannot exceed 16 characters.</p> <p>If not specified, synthetic SCSI INQUIRY data is returned containing a PRODUCT ID selected based on the disk size.</p> <p>Initially left unspecified.</p>

## geometry

<b>Parameter</b>	geometry[N] N=0...7
<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 <code>&lt;X&gt;"\"&lt;Y&gt;["\"&lt;Z&gt;]</code> where:</p> <ul style="list-style-type: none"> <li>• "X" is the number of sectors per track;</li> <li>• "Y" is the number of tracks per cylinder;</li> <li>• "Z" (optional) is 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;</li> </ul> <p>If this parameter is not set, CHARON-VAX will configure the geometry based on the most probable disk type.</p>

## use\_io\_file\_buffering

<b>Parameter</b>	use_io_file_buffering[N] N=0...7
<b>Type</b>	Text String
<b>Value</b>	<p>Instructs CHARON-VAX to enable host operating system I/O cache on reading/writing operations.</p> <p>The following values are possible:</p> <ul style="list-style-type: none"> <li>• "true" or "write-back" - the "write back" caching mode is used</li> <li>• "write-through" - the "write through" caching mode is used</li> <li>• "false" - caching is switched off</li> </ul> <p> Note that this caching has a significant effect only in case of mapping to ISO images, not physical drives.</p> <p>When enabled, host operating system I/O cache may significantly improve I/O performance of the virtual system. At the same time maintaining I/O cache requires additional host resources (CPU and memory) which may negatively affect overall performance of the virtual system.</p> <p>Initially is set to "false".</p> <p><b>Example:</b></p> <pre>set pka_0 use_io_file_buffering[0]=true</pre>

 In the table above N stands for logical unit number. The first unit must be 0 with no gaps in subsequent numbering. If only one LUN is configured, the LUN number can be omitted

**Example:**

```
load virtual_scsi_cdrom/chscsi pka_0 scsi_bus=pka scsi_id=0
set pka_0 container="\\.\CdRom0"
```

This example associates an unallocated CD-ROM drive "\\.\CdRom0" with virtual\_scsi\_cdrom "pka\_0".

## physical\_scsi\_device

Use "physical\_scsi\_device" to connect any host "raw" SCSI device to CHARON-VAX. This mapping type is suitable only for tape drives and other specific SCSI devices connected to CHARON host.

"physical\_scsi\_device" has the following parameters:

### container

<b>Parameter</b>	container[N] N=0...7
<b>Type</b>	Text String
<b>Value</b>	<p>A string containing the device name to map to the emulator.</p> <ul style="list-style-type: none"> <li>• <b>Physical tape drives recognized by Windows</b> <ul style="list-style-type: none"> <li>• "\\.\Tape&lt;X&gt;", where X is 0, 1, ...</li> </ul> </li> <li>• <b>Tape changer</b> <ul style="list-style-type: none"> <li>• "\\.\Changer&lt;X&gt;", where X is 0, 1, ...</li> </ul> </li> </ul> <p>This parameter is initially not set, thus creating NO storage elements on the controller</p>

### media\_type

<b>Parameter</b>	media_type[N] N=0...7
<b>Type</b>	Text String
<b>Value</b>	<p>Overrides the PRODUCT ID in the default SCSI INQUIRY data.</p> <p>Valid values may contain uppercase letters, integers and spaces. The length of the string cannot exceed 16 characters.</p> <p>If not specified, synthetic SCSI INQUIRY data is returned containing a PRODUCT ID selected based on the disk size.</p> <p>Initially left unspecified.</p>

### geometry


<b>Parameter</b>	geometry[N] N=0...7
<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;] where:</p> <ul style="list-style-type: none"> <li>• "X" is the number of sectors per track;</li> <li>• "Y" is the number of tracks per cylinder;</li> <li>• "Z" (optional) is 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;</li> </ul> <p>If this parameter is not set, CHARON-VAX will configure the geometry based on the most probable disk type.</p>

## removable

<b>Parameter</b>	removable[N] N=0...7
<b>Type</b>	boolean
<b>Value</b>	Enables the logical unit to appear as a removable SCSI disk drive. Initially set to "false" (fixed, non-removable).

## disconnect\_timeout

<b>Parameter</b>	disconnect_timeout[N] N=0...7
<b>Type</b>	Numeric
<b>Value</b>	Sets logical unit disconnect timeout. This parameter helps if a connected SCSI device performs a given SCSI command for a very long time.  The default value depends on the type of the SCSI device attached. For example for a disk it is 10 seconds, for a tape - an hour. If the type of the device is not known the timeout is 1 hour.  <b>Example:</b> 48 days timeout  <code>set pka_0 disconnect_timeout[1] = 0x400000</code>

 In the table above N stands for logical unit number. The first unit must be 0 with no gaps in subsequent numbering. If only one LUN is configured, the LUN number can be omitted.

**Example:**

```
load physical_scsi_device/chscsi pka_0 scsi_bus=pka scsi_id=0
set pka_0 container="\\.\Tape0"
```

This example associates a host tape device "\\.\Tape0" with physical\_scsi\_device "pka\_0".

# DSSI Subsystem

## Table of Contents

- Introduction
- SHAC host adapter
  - port
  - host
  - scs\_node\_name
  - scs\_system\_id
  - mscp\_allocation\_class
  - container
  - media\_type
  - geometry
  - use\_io\_file\_buffering
- HSD50 storage controller
  - dssi\_host
  - dssi\_node\_id
  - scs\_node\_name
  - scs\_system\_id
  - mscp\_allocation\_class
  - container
  - media\_type
  - geometry
  - use\_io\_file\_buffering

## Introduction

The DSSI storage subsystem, for the VAX 4000 Models: 106, 108, 700 and 705 emulators, is based on the emulation of SHAC host adapters and the ability to route SCS data packets between the emulated SHAC host adapters, of multiple nodes, via separate TCP/IP links.

The DSSI storage subsystem is functionally emulated, but the emulation is incompatible with the physical DSSI and operates at a much higher throughput than the original hardware. Connection to the physical DSSI hardware is neither possible nor planned for future releases.


This version of the DSSI emulation for CHARON-VAX supports up to 3 VAX nodes in a virtual DSSI cluster and handles a maximum cluster size of 8 nodes. A single virtual DSSI network supports up to 256 storage elements.


To use a **single** CHARON-VAX system with DSSI emulation, either one or both of the following two elements must be configured:

1. A DSSI storage element (disk or tape).
2. A DSSI storage controller. Currently an emulated HSD50 storage controller is provided. The emulated HSD50 supports physical host drives, CD-ROM drives, physical tapes, removable disks, virtual disks and virtual tapes.

To create a **cluster** of DSSI interconnected CHARON-VAX systems, the DSSI hardware topology is emulated by establishing TCP/IP channels between the emulated SHAC host adapters on each CHARON-VAX system (The use of TCP/IP for the interconnects makes the cluster in principle routable in a WAN). Virtual HSD50 storage controllers are then connected to every SHAC host adapter in the virtual DSSI network.

Cluster operation requires (virtual) disks that are simultaneously accessible by all CHARON-VAX nodes involved. This can be implemented, for instance, by using a properly configured iSCSI initiator / target structure or a fiber channel storage back-end.

 When a tape or disk image connected to an virtual HSD50 storage controller is dismounted in VAX/VMS, it is disconnected from CHARON-VAX and can be manipulated. It may even 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.

 The emulated DSSI subsystem has many configurable parameters when multiple nodes on a single DSSI bus are to be connected. Incorrect configuration, in particular non-identical specification of emulated HSD50 disks in the DSSI nodes, is likely to cause data corruption. It is advisable to start any field test by implementing a single node.

## SHAC host adapter

To connect an emulated VAX 4000 model 106, 108, 700 and 705 node to a virtual DSSI network, the CHARON-VAX configuration must load at least one emulated SHAC host adapter.

Emulated VAX 4000 models 106, 108, 700 and 705 have two pre-loaded SHAC host adapters named "PAA" and "PAB". There is no need to load any extra instances of SHAC in the configuration file.

**Note:** VAX/VMS running on an emulated VAX 4000 model 106 or 108 node enumerates the emulated SHAC host adapters and assigns them the VMS internal names "PAA" and "PAB". It is recommended, for clarity, to keep the same naming scheme in the CHARON-VAX configuration file for these emulated SHAC host adapters.

A virtual SHAC has the following configuration parameters:

### port

<b>Parameter</b>	port[N] N=0...7
<b>Type</b>	Numeric
<b>Value</b>	An integer value that specifies the TCP/IP port number on which the emulated SHAC host adapter listens for connections from another emulated SHAC host adapter with DSSI node id: N.  Possible values are from 1024 through 32767.  Initially not set.

### host

<b>Parameter</b>	host[N] N=0...7
<b>Type</b>	Text String
<b>Value</b>	A string value that specifies the TCP/IP hostname (and optionally the TCP/IP port number) to connect to another emulated SHAC host adapter with DSSI node id: N.  The syntax for the string is "host-name[:port-no]", with possible values for port-no in the range of 1024 through 32767.  Initially not set.

### scs\_node\_name

<b>Parameter</b>	scs_node_name[N] N=0...7
<b>Type</b>	Text String
<b>Value</b>	A string value that specifies the SCSNODENAME of the emulated storage element. This string can contain up to 6 characters. Possible characters are uppercase letters A through Z and integers 0 through 9.  Initially set to an arbitrary value that is guaranteed to be unique within the running emulated VAX 4000 model 106, 108, 700 or 705 node.




## scs\_system\_id

<b>Parameter</b>	scs_system_id[N] N=0...7
<b>Type</b>	Numeric
<b>Value</b>	An integer value that specifies the SCSSYSTEMID of the emulated storage element. Initially set to an arbitrary value that is guaranteed to be unique within the running emulated VAX 4000 model 106, 108, 700 or 705 node.

## mscp\_allocation\_class

<b>Parameter</b>	mscp_allocation_class[N] N=0...7
<b>Type</b>	Numeric
<b>Value</b>	An integer value that specifies the ALLOCLASS of the emulated storage element. Possible values are from 0 through 255. Initially set to 0 which means no allocation class assigned.

## container

<b>Parameter</b>	container[N] N=0...7
<b>Type</b>	Text String
<b>Value</b>	<p>A string value that specifies the container of the storage element with DSSI node id: N and MSCP unit number: N. This storage element might be either a (virtual) disk or tape. In VMS, running on an emulated VAX 4000 model 106 or 108 node, these storage elements appear as DSSI disks (DIAN:) or DSSI (TF86) tapes (MIAN:).</p> <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 drives</b> <ul style="list-style-type: none"> <li>"\\.\PhysicalDrive&lt;X&gt;", where X is 0, 1, ...</li> <li>"\\.\PhysicalDrive(DevID = XXXX-XXXX-XXXX-XXXX-XXXX-XXXX-XXXX-XXXX)"</li> </ul> <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><b>Example:</b></p> <pre>set PAA container[2]="\\.\PhysicalDrive(DevID= 6008-05F3-0005-2950-BF8E-0B86-A0C7-0001) "</pre> <p>The WWID values can be obtained from "<a href="#">Host Device Check</a>" utility ("All drives" section).</p> <p> If the "Host Device Check" utility does not display any WWID, this means the target disk does not have one. Use the "\\.\PhysicalDrive&lt;N&gt;" mapping in this case.</p> </li> <li> <b>iSCSI disks</b> <ul style="list-style-type: none"> <li>"\\.\PhysicalDrive(iScsiTarget = &lt;iSCSI target&gt;, LUN = &lt;LUN number&gt;)"</li> </ul> <p>iScsiTarget addresses the disk by its iSCSI target name. LUN specifies LUN on connected iSCSI disk.</p> <p><b>Example:</b></p> <pre>set PAA container[3]="\\.\PhysicalDrive(iScsiTarget= ign.2008-04:iscsi.charon-target-test1, LUN= 1) "</pre> </li> <li> <b>Physical tape drives recognized by Windows</b> <ul style="list-style-type: none"> <li>"\\.\Tape&lt;X&gt;", where X is 0, 1, ...</li> </ul> </li> <li> <b>Floppy drives</b> <ul style="list-style-type: none"> <li>"\\.\A:" or</li> <li>"\\.\B:"</li> </ul> </li> <li> <b>CD-ROM drives (read-only)</b> <ul style="list-style-type: none"> <li>"\\.\CdRom&lt;X&gt;", where X is 0, 1, ...</li> </ul> </li> <li> <b>CHARON-VAX disk images</b> <ul style="list-style-type: none"> <li>[&lt;drive&gt;:\&lt;path-name&gt;\&lt;file-name&gt;[" .vdisk"]</li> </ul> </li> <li> <b>CHARON-VAX tape images</b> <ul style="list-style-type: none"> <li>[&lt;drive&gt;:\&lt;path-name&gt;\&lt;file-name&gt; .vtape</li> </ul> </li> </ul> <p>This parameter is initially not set, thus creating NO storage elements on the bus with corresponding DSSI node id.</p>

## media\_type

<b>Parameter</b>	media_type[N] N=0...7
<b>Type</b>	Text String
<b>Value</b>	<p>Overrides default (automatically determined) MSCP media type of the device.</p> <p><b>Syntax:</b></p> <pre>"&lt;device-name&gt; , &lt;device-type&gt; "</pre> <p>where:</p> <ul style="list-style-type: none"> <li>• &lt;device-name&gt; is either: "DU", "MU", "DK", "MK", "SCSI", "DI", "MI", "DSSI", "DJ" or "MJ"</li> <li>• &lt;device-type&gt; is the form of: "LLD" or "LLLD", where "L" is a letter from A through Z, and "D" is a decimal number from 0 through 99</li> </ul> <p>If not specified, the device name is set to "DI", and the device type is selected based on disk size for the disk storage elements. For tape storage elements, the device name and type are set to "MI" and "TF86", respectively.</p> <p>Initially not specified.</p>

## geometry

<b>Parameter</b>	geometry[N] N=0...7
<b>Type</b>	Text String
<b>Value</b>	<p>This formatted string value specifies the explicit geometry of the disk storage element with DSSI node ID: N and MSCP unit number: N. This parameter is not applicable to tape storage elements.</p> <p>The string format is &lt;X&gt;"/"/&lt;Y&gt;["/"&lt;Z&gt;] where:</p> <ol style="list-style-type: none"> <li>1. X is the number of sectors per track</li> <li>2. Y is the number of tracks per cylinder</li> <li>3. Z (optional) is 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</li> </ol> <p>If this parameter is not set, CHARON-VAX will configure the geometry based on the most probable disk type.</p> <p>Initially not set.</p> <p>The syntax above is applicable only to disk storage elements.</p> <p>If the container is a tape image, the following format is used:</p> <p><b>Syntax:</b></p> <pre>"&lt;image-size&gt;[ , &lt;early-warning-zone-size&gt;]"</pre> <p>where:</p> <ul style="list-style-type: none"> <li>• &lt;image-size&gt; is the tape size in MB</li> <li>• &lt;early-warning-zone-size&gt; is the size (in KB) of the space left on the tape when a warning to the OS is issued. If omitted, 64K is assumed.</li> </ul>

## use\_io\_file\_buffering

<b>Parameter</b>	use_io_file_buffering[N] N=0...7
<b>Type</b>	Text String
<b>Value</b>	<p>Instructs CHARON-VAX to enable host operating system I/O cache on reading/writing operations.</p> <p>The following values are possible:</p> <ul style="list-style-type: none"> <li>• "true" or "write-back" - the "write back" caching mode is used</li> <li>• "write-through" - the "write through" caching mode is used</li> <li>• "false" - caching is switched off</li> </ul> <p><b>i</b> Note that this caching has a significant effect only in case of mapping to disk and tape containers, not physical drives.</p> <p>When enabled, host operating system I/O cache may significantly improve I/O performance of the virtual system. At the same time maintaining I/O cache requires additional host resources (CPU and memory) which may negatively affect overall performance of the virtual system.</p> <p>Initially is set to "false".</p> <p><b>Example:</b></p> <pre>set PAA use_io_file_buffering[0]=true</pre>

These parameters are configured with the "set" command.

**Example:** Standalone VAX system with 2 virtual DSSI disks on a PAA SHAC controller

```
set session hw_model="VAX_4000_Model_108"
load virtual_serial_line/chserial OPA0 port=10003 application="putty.exe -load OPA0"
set quart line[3]=OPA0
set PAA container[0]="C:\Charon\Disks\dia0-rz24-vms-v6.2.vdisk"
set PAA container[1]="C:\Charon\Disks\dial-rz24-vms-v6.2.vdisk"
```

The emulated VAX 4000 model 106 or 108 can then boot VMS with the following command:

```
>>> BOOT DIA0
```

After logging into VMS, the "SHOW DEVICE D" command displays the following:

```
$ show devices d
Device          Device          Error  Volume          Free  Trans  Mnt
Name            Status          Count  Label           Blocks Count  Cnt
004200$DIA0:    Mounted        0      DSSI01          32022  147   1
004201$DIA1:    Online         0
```

## HSD50 storage controller

To connect a storage controller to the virtual DSSI network, the CHARON-VAX configuration file must load at least one emulated HSD50 storage controller. In most cases one emulated HSD50 storage controller per virtual DSSI network is enough. The CHARON-VAX configuration file must supply a unique reference name for that instance. Even though this name is only valid within the configuration file, it is recommended, for clarity, to use the VMS SCSSNODENAME as the instance name. The line below loads an emulated HSD50 storage controller, assigns it the instance name SCSSNODE and connects it to the primary built-in DSSI controller:

```
load HSD50 MYDISKS dssi_host=PAA
```

The HSD50 emulation has the following configuration parameters:

### dssi\_host

<b>Parameter</b>	dssi_host
<b>Type</b>	Text String
<b>Value</b>	A string value that specifies the instance name of an emulated SHAC host adapter serving the virtual DSSI network. If this value is not set, CHARON-VAX will try to locate the host adapter automatically. This automatic lookup works only if the CHARON-VAX configuration has exactly one instance of an emulated SHAC host adapter.

### dssi\_node\_id

<b>Parameter</b>	dssi_node_id
<b>Type</b>	Numeric
<b>Value</b>	An integer value that specifies the address of an emulated HSD50 storage controller on the virtual DSSI network. Possible values are from 0 through 7 (initially set to 0).

### scs\_node\_name

<b>Parameter</b>	scs_node_name
<b>Type</b>	Text String
<b>Value</b>	A string value that specifies the SCSSNODENAME of the emulated HSD50 storage controller. The string can contain as many as 6 characters. Possible characters are uppercase letters: A through Z, and the integers: 0 through 9. Initially set to the name of the emulated HSD50 controller. Therefore, the name of the emulated HSD50 controller should follow the above rules.


### scs\_system\_id

<b>Parameter</b>	scs_system_id
<b>Type</b>	Numeric
<b>Value</b>	An integer value that specifies the SCSSYSTEMID of the emulated HSD50 storage controller. Initially set to an arbitrary value that is guaranteed to be unique within the running emulated VAX 4000 model 106, 108, 700 and 705 node.

## mscp\_allocation\_class

<b>Parameter</b>	mscp_allocation_class
<b>Type</b>	Numeric
<b>Value</b>	An integer value that specifies the ALLOCLASS of the emulated HSD50 storage controller. Possible values are 0 through 255 (initially set to 0).

## container

<b>Parameter</b>	container[N]N=0...9999
<b>Type</b>	Text String
<b>Value</b>	<p>A string value that specifies the container of the storage element with MSCP unit number: N. This storage element might be either a (virtual) disk or tape. In VMS running on an emulated VAX 4000 node, these storage elements appear as HSX00 disks (DUAN:) or HST00 tapes (MUAN:). Possible values of the parameter are strings in one of the following forms:</p> <ul style="list-style-type: none"> <li> <b>Physical disk drives</b> <ul style="list-style-type: none"> <li>"\\.\PhysicalDrive&lt;X&gt;", where X is 0, 1, ...</li> <li>"\\.\PhysicalDrive(DevID =XXXX-XXXX-XXXX-XXXX-XXXX-XXXX-XXXX-XXXX)" DevID addresses the target physical disk by its WWID (hexadecimal 128-bit identifier assigned to the disk drive by its manufacturer/originator). <b>Example:</b> <pre>set DISKS container[2]="\\.\PhysicalDrive(DevID= 6008-05F3-0005-2950-BF8E-0B86-A0C7-0001) "</pre> The WWID values can be obtained from "Host Device Check" utility ("All drives" section)</li> </ul> <p> If the "Host Device Check" utility does not display any WWID, this means the target disk does not have one. Use the "\\.\PhysicalDrive&lt;N&gt;" mapping in this case.</p> </li> <li> <b>iSCSI disks</b> <ul style="list-style-type: none"> <li>"\\.\PhysicalDrive(iScsiTarget = &lt;iSCSI target&gt;, LUN = &lt;LUN number&gt;)" iScsiTarget addresses the disk by its iSCSI target name. LUN specifies LUN on connected iSCSI disk. <b>Example:</b> <pre>set DISKS container[3]="\\.\PhysicalDrive(iScsiTarget= iqn.2008-04:iscsi.charon-target-test1, LUN= 1) "</pre> </li> </ul> </li> <li> <b>Physical tape drives recognized by Windows</b> <ul style="list-style-type: none"> <li>"\\.\Tape&lt;X&gt;", where X is 0, 1, ...</li> </ul> </li> <li> <b>Floppy drives</b> <ul style="list-style-type: none"> <li>"\\.\A:" or</li> <li>"\\.\B:"</li> </ul> </li> <li> <b>CD-ROM drives (read-only)</b> <ul style="list-style-type: none"> <li>"\\.\CdRom&lt;X&gt;", where X is 0, 1, ...</li> </ul> </li> <li> <b>CHARON-VAX disk images</b> <ul style="list-style-type: none"> <li>[&lt;drive&gt;:\&lt;path-name&gt;\"]&lt;file-name&gt;[".vdisk"]</li> </ul> </li> <li> <b>CHARON-VAX tape images</b> <ul style="list-style-type: none"> <li>[&lt;drive&gt;:\&lt;path-name&gt;\"]&lt;file-name&gt;".vtape"</li> </ul> </li> </ul> <p>This parameter is initially not set, thus creating NO storage elements on the controller</p>

## media\_type

<b>Parameter</b>	media_type[N]N=0...9999
<b>Type</b>	Text String
<b>Value</b>	<p>Overrides the default (automatically determined) MSCP media type of the device.</p> <p><b>Syntax:</b></p> <pre>"&lt;device-name&gt; , &lt;device-type&gt; "</pre> <p>where:</p> <ul style="list-style-type: none"> <li>• &lt;device-name&gt; is either: "DU", "MU", "DK", "MK", "SCSI", "DI", "MI", "DSSI", "DJ" or "MJ"</li> <li>• &lt;device-type&gt; is the form of: "LLD" or "LLLD", where "L" is a letter from A through Z, and "D" is a decimal number from 0 through 99</li> </ul> <p>If not specified, the device name is set to "DI" and the device type is selected based on disk size for disk storage elements. For tape storage elements, the device name and type are set to "MI" and "TF86" respectively. Initially not specified.</p>

## geometry

<b>Parameter</b>	geometry[N]N=0...9999
<b>Type</b>	Text String
<b>Value</b>	<p>This formatted string value specifies the explicit geometry of the disk storage element with DSSI node ID: N and MSCP unit number: N. This parameter is not applicable to tape storage elements. The string format is &lt;X&gt;"/&gt; </p>

## use\_io\_file\_buffering

<b>Parameter</b>	use_io_file_buffering[N]N=0...9999
<b>Type</b>	Text String
<b>Value</b>	<p>Instructs CHARON-VAX to enable host operating system I/O cache on reading/writing operations. The following values are possible:</p> <ul style="list-style-type: none"> <li>• "true" or "write-back" - the "write back" caching mode is used</li> <li>• "write-through" - the "write through" caching mode is used</li> <li>• "false" - caching is switched off</li> </ul> <p><b>i</b> Note that this caching has a significant effect only in case of mapping to disk and tape containers, not physical drives. When enabled, host operating system I/O cache may significantly improve I/O performance of the virtual system. At the same time maintaining I/O cache requires additional host resources (CPU and memory) which may negatively affect overall performance of the virtual system. Initially is set to "false".</p> <p><b>Example:</b></p> <pre>set DISKS use_io_file_buffering[0]=true</pre>

**Example:**

```
load HSD50 DISKS dssi_host=PAA dssi_node_id=5
```

The configuration file below emulates a VAX 4000 Model 108 node with one HSD50 storage controller serving two disks and another instance of an HSD50 controller that serves a tape drive to the VAX over a virtual DSSI:

```
set session hw_model="VAX_4000_Model_108"
load virtual_serial_line/chserial OPA0 port=10003 application="putty.exe -load OPA0"
set quart line[3]=OPA0
load HSD50 DISKS dssi_host=PAA dssi_node_id=1
set DISKS container[0]="C:\Charon\Disks\dua0-rz24-vms-v6.2.vdisk"
set DISKS container[1]="C:\Charon\Disks\dua1-rz24-vms-v6.2.vdisk"
load HSD50 TAPES dssi_host=PAA dssi_node_id=2set TAPES container[3]="\\.\Tape0"
```

In this example we emulate two HSD50 instances. They are both connected to the same virtual DSSI bus, therefore we must assign them different DSSI node ID values.

The emulated VAX 4000 Model 108 can boot VMS with the following command:

```
>>> BOOT DUA0
```

After logging into VMS, the "SHOW DEVICE D" and "SHOW DEVICE M" command displays the following:

```
$ show devices d
Device          Device          Error   Volume          Free  Trans  Mnt
Name           Status          Count   Label           Blocks Count  Cnt
DISKS$DUA0:    Mounted         0       DSSI01          31932  147   1
DISKS$DUA1:    Online          0
$ show devices m
Device          Device          Error   Volume          Free  Trans  Mnt
Name           Status          Count   Label           Blocks Count  Cnt
TAPES$MUA3:    Online          0
```



# CI Subsystem

## Table of Contents

- Introduction
- CIXCD host adapter
  - port
  - host
  - xmi\_node\_id
  - ci\_node\_id
- HSJ50 storage controller
  - ci\_host
  - ci\_node\_id
  - scs\_node\_name
  - scs\_system\_id
  - mscp\_allocation\_class
  - container
  - media\_type
  - geometry
  - use\_io\_file\_buffering

## Introduction


The virtual CIXCD is the functional equivalent of a hardware CIXCD host adapter, with the exception that there is no physical layer to connect to. The current host hardware is an order of magnitude faster than the physical CI implementation, such connections - if it were possible - would greatly limit the virtual system throughput.

For data storage, the CIXCD connects to one or more virtual HSJ50 storage controllers, that are loaded as a separate component in the configuration file. To configure VAX CI clusters, the virtual CIXCDs of the multiple CHARON-VAX/66X0 instances are interconnected via TCP/IP links.

Configuring (large) virtual VAX CI clusters requires many configurable parameters and a replicated identical definition of the shared virtual HSJ50 storage controllers in each virtual VAX instance.

To connect a virtual VAX 66x0 to a virtual CI network, the CHARON-VAX configuration file must load at least one virtual CIXCD host adapter; one unit is sufficient in all practical cases.

VAX/VMS enumerates the virtual CIXCD host adapters in the order of increasing XML node IDs and assigns them the VMS internal names PAA, PAB, etc. It is recommended for clarity to keep the same naming scheme for virtual CIXCD host adapters in the configuration file.

-  The emulated CI subsystem has many configurable parameters when multiple nodes on a single CI bus are to be configured. Incorrect configuration, in particular non-identical specification of the emulated HSJ50 storage controller disks in the CI nodes, is likely to cause data corruption. It is advisable to start any field test by implementing a single node.

## CIXCD host adapter

To connect an emulated VAX 66x0 node to a virtual CI network, the CHARON-VAX configuration must load at least one emulated CIXCD host adapter.

To load the adapter and assign it an instance name of "PAA", enter the following line in the configuration file:

```
load CIXCD PAA
```

The CIXCD emulation has the following configuration parameters:

## port

<b>Parameter</b>	port[N] N=0...127
<b>Type</b>	Numeric
<b>Value</b>	An integer value that specifies the TCP/IP port number on which the emulated CIXCD host adapter listens for connections from other emulated CIXCD host adapters with address: N.  Possible values are from 1024 through 32767.  Initially not set.

## host

<b>Parameter</b>	host[N] N=0...127
<b>Type</b>	Text String
<b>Value</b>	A string value that specifies the TCP/IP hostname (and optionally the TCP/IP port number) to connect to another emulated CIXCD host adapter with address: N.  The syntax for the string is "host-name[:port-no]", with possible values for the port-no in the range of 1024 through 32767.  Initially not set.

## xmi\_node\_id

<b>Parameter</b>	xmi_node_id
<b>Type</b>	Numeric
<b>Value</b>	An integer value that specifies the location of the virtual CIXCD host adapter on the XMI bus.  Possible values are 11 through 14 (Initially set to 14).

## ci\_node\_id

<b>Parameter</b>	ci_node_id
<b>Type</b>	Numeric
<b>Value</b>	An integer value that specifies the address of the virtual CIXCD host adapter on the virtual CI network.  Possible values are 0 through 127 (Initially set to 127).

These parameters can be added to the "load" command or specified separately with the "set" command.

The example below shows how to configure a virtual CIXCD adapter with a location on the XMI bus other than the default. It declares a CIXCD adapter in slot 11 (0xB = decimal 11) of the virtual XMI bus:

```
load CIXCD PAA xmi_node_id=0xB
```

The configuration file below creates a virtual VAX 6610 (single CPU) node with one virtual HSJ50 storage controller serving two disks to the VAX over a virtual CI network:

```
set session hw_model="VAX_6000_Model_610"
set session log="vax6610.log"
set toy container="vax6610.dat"
set eeprom container="vax6610.rom"
load virtual_serial_line/chserial OPA0
set OPA0 port=10003 application="putty.exe -load OPA0"
load CIXCD PAA
load HSJ50 DISKS
set DISKS container[0]="C:\Charon\Disks\dual0-rz24-vms-v6.2.vdisk"
set DISKS container[1]="C:\Charon\Disks\dual1-rz24-vms-v6.2.vdisk"
```

## HSJ50 storage controller

The virtual HSJ50 storage controller functionally replaces a physical HSJ50 CI storage unit. It supports virtual and physical disks, tapes and removable storage devices that are mapped on local or remote host platform storage. The virtual HSJ50 cannot connect to a physical CI infrastructure.

In a single CHARON-VAX/66X0 instance without a CI cluster, the virtual HSJ50 is located as a separate entity on the same host platform. In a CI cluster, the definition of each HSJ50 is replicated exactly in each CHARON-VAX CI node. In most cases one HSJ50 storage controller per virtual CI network is enough.

When loading an instance of a virtual HSJ50 storage controller, the CHARON-VAX configuration file must supply a unique reference name for that instance. While this name is only valid within the configuration file, it is recommended, for clarity, to use the VAX/VMS SCSNODENAME as an instance name.

The line below loads an emulated HSJ50 storage controller and assigns it the instance name SCSNODE:

```
load HSJ50 MYDISKS
```

The HSJ50 emulation has the following configuration parameters:

### ci\_host

<b>Parameter</b>	ci_host
<b>Type</b>	Text String
<b>Value</b>	A string value that specifies the instance name of the emulated CIXCD host adapter serving the virtual CI network.  If this value is not set, CHARON-VAX will try to locate the host adapter automatically. Automatic lookup works only if the CHARON-VAX configuration has exactly one instance of an emulated CIXCD host adapter.

### ci\_node\_id

<b>Parameter</b>	ci_node_id
<b>Type</b>	Numeric
<b>Value</b>	An integer value that specifies the address of an emulated HSJ50 storage controller on the virtual CI network.  Possible values are 0 through 127 (initially set to 0).

## scs\_node\_name

<b>Parameter</b>	scs_node_name
<b>Type</b>	Text String
<b>Value</b>	<p>A string value that specifies the SCSNODENAME of the emulated HSJ50 storage controller.</p> <p>The string can contain up to 6 characters. Possible characters are uppercase letters A through Z and the integers 0 through 9.</p> <p>Initially set to the name of the emulated HSJ50 controller. Therefore the name of the emulated HSJ50 controller must follow the above rules.</p>


## scs\_system\_id

<b>Parameter</b>	scs_system_id
<b>Type</b>	Numeric
<b>Value</b>	<p>An integer value that specifies the SCSSYSTEMID of the emulated HSJ50 storage controller.</p> <p>Initially set to an arbitrary value that is guaranteed to be unique within the running emulated VAX 66x0 node.</p>

## mscp\_allocation\_class

<b>Parameter</b>	mscp_allocation_class
<b>Type</b>	Numeric
<b>Value</b>	<p>An integer value that specifies the ALLOCLASS of the emulated HSJ50 storage controller.</p> <p>Possible values are 0 through 255 (initially set to 0).</p>

## container

<b>Parameter</b>	container[N] N=0...9999
<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 tape drives</b> <ul style="list-style-type: none"> <li>"\\.\PhysicalDrive&lt;X&gt;", where X is 0, 1, ...</li> <li>"\\.\PhysicalDrive(DevID =XXXX-XXXX-XXXX-XXXX-XXXX-XXXX-XXXX-XXXX)"</li> </ul> <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><b>Example:</b></p> <pre>set DISKS container[2]="\\.\PhysicalDrive(DevID= 6008-05F3-0005-2950-BF8E-0B86-A0C7-0001)"</pre> <p>The WWID values can be obtained from "Host Device Check" utility ("All drives" section).</p> <p> If the "Host Device Check" utility does not display any WWID, this means the target disk does not have one. Use the "\\.\PhysicalDrive&lt;N&gt;" mapping in this case.</p> </li> <li> <b>iSCSI disks</b> <ul style="list-style-type: none"> <li>"\\.\PhysicalDrive(iScsiTarget = &lt;iSCSI target&gt;, LUN = &lt;LUN number&gt;)"</li> </ul> <p>iScsiTarget addresses the disk by its iSCSI target name. LUN specifies LUN on connected iSCSI disk.</p> <p><b>Example:</b></p> <pre>set DISKS container[3]="\\.\PhysicalDrive(iScsiTarget= iqn.2008-04:iscsi.charon-target-test1, LUN= 1)"</pre> </li> <li> <b>Physical tape drives recognized by Windows</b> <ul style="list-style-type: none"> <li>"\\.\Tape&lt;X&gt;", where X is 0, 1, ...</li> </ul> </li> <li> <b>Floppy drives</b> <ul style="list-style-type: none"> <li>"\\.\A:" or</li> <li>"\\.\B:"</li> </ul> </li> <li> <b>CD-ROM drives (read-only)</b> <ul style="list-style-type: none"> <li>"\\.\CdRom&lt;X&gt;", where X is 0, 1, ...</li> </ul> </li> <li> <b>CHARON-VAX disk images</b> <ul style="list-style-type: none"> <li>[&lt;drive&gt;:\&lt;path-name&gt;\"]&lt;file-name&gt;[".vdisk"]</li> </ul> </li> <li> <b>CHARON-VAX tape images</b> <ul style="list-style-type: none"> <li>[&lt;drive&gt;:\&lt;path-name&gt;\"]&lt;file-name&gt;".vtape"</li> </ul> </li> </ul> <p>This parameter is initially not set, thus creating NO storage elements on the controller</p>


## media\_type

<b>Parameter</b>	media_type[N] N=0...9999
<b>Type</b>	Text String
<b>Value</b>	<p>Overrides the default (automatically determined) MSCP media type of the device.</p> <p><b>Syntax:</b></p> <pre>"&lt;device-name&gt; , &lt;device-type&gt; "</pre> <p>where:</p> <ul style="list-style-type: none"> <li>• &lt;device-name&gt; is either: "DU", "MU", "DK", "MK", "SCSI", "DI", "MI", "CI", "DJ" or "MJ"</li> <li>• &lt;device-type&gt; is in the for of: "LLD" or "LLLD", where "L" is a letter from A through Z, and "D" is a decimal number from 0 through 99</li> </ul> <p>If not specified, the device name is set to "DI" and the device type is selected based on disk size for disk storage elements. For tape storage elements, the device name and type are set to "MI" and "TF86" respectively.</p> <p>Initially not specified.</p>

## geometry

<b>Parameter</b>	geometry[N] N=0...9999
<b>Type</b>	Text String
<b>Value</b>	<p>This formatted string value specifies the explicit geometry of the disk storage element with CI node ID: N and MSCP unit number: N. This parameter is not applicable to tape storage elements.</p> <p>The string format is &lt;X&gt;"/"&lt;Y&gt;["/"&lt;Z&gt;] where:</p> <ol style="list-style-type: none"> <li>1. X is the number of sectors per track</li> <li>2. Y is the number of tracks per cylinder</li> <li>3. Z (optional) is 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</li> </ol> <p>If this parameter is not set, CHARON-VAX will configure the geometry based on the most probable disk type.</p> <p>Initially not set.</p> <p>The syntax above is applicable only to disk storage elements.</p> <p>If the container is a tape image, the following format is used:</p> <p><b>Syntax:</b></p> <pre>"&lt;image-size&gt;[ , &lt;early-warning-zone-size&gt;]"</pre> <p>where:</p> <ul style="list-style-type: none"> <li>• &lt;image-size&gt; is the tape size in MB</li> <li>• &lt;early-warning-zone-size&gt; is a size (in KB) of the space left on the tape when a warning to the OS is issued. If omitted, 64K is assumed.</li> </ul>

## use\_io\_file\_buffering

<b>Parameter</b>	use_io_file_buffering[N] N=0...9999
<b>Type</b>	Text String
<b>Value</b>	<p>Instructs CHARON-VAX to enable host operating system I/O cache on reading/writing operations.</p> <p>The following values are possible:</p> <ul style="list-style-type: none"> <li>• "true" or "write-back" - the "write back" caching mode is used</li> <li>• "write-through" - the "write through" caching mode is used</li> <li>• "false" - caching is switched off</li> </ul> <p> Note that this caching has a significant effect only in case of mapping to disk and tape containers, not physical drives.</p> <p>When enabled, host operating system I/O cache may significantly improve I/O performance of the virtual system. At the same time maintaining I/O cache requires additional host resources (CPU and memory) which may negatively affect overall performance of the virtual system.</p> <p>Initially is set to "false".</p> <p><b>Example:</b></p> <pre>set DISKS use_io_file_buffering[0]=true</pre>

The following example configures a virtual HSJ50 storage controller with a non-default CI network address of 11:

```
load HSJ50 DISKS ci_node_id=0x0B
```

The configuration file below emulates a VAX 6610 node with one HSJ50 storage controller serving two disks:

```
set session hw_model="VAX_6000_Model_610"
set session log="vax6610.log"
set toy container="vax6610.dat"
set eeprom container="vax6610.rom"
load virtual_serial_line/chserial OPA0
set OPA0 port=10003 application="putty.exe -load OPA0"
load CIXCD PAA xmi_node_id=0x0C
load HSJ50 DISKS ci_node_id=0x0B
set DISKS container[0]="C:\Charon\Disks\dua0-rz24-vms-v6.2.vdisk"
set DISKS container[1]="C:\Charon\Disks\dual-rz24-vms-v6.2.vdisk"
```

When this configuration file is executed and "container[0]" points to a valid VMS system disk image, a virtual VAX 6610 can boot VAX/VMS with the following command:

```
>>> BOOT /XMI:C /NODE:B DU0
```

In the above boot command, "/XMI:C" and "/NODE:B" instruct the boot ROM to connect to the disk via the host adapter in XMI slot C (the hex value C stands for decimal 12).


Then via the storage controller with CI node id B (decimal 11), DU0 is reached (defined as container[0]) and the boot command is executed for the associated file on the host system.

After logging into VMS, the "SHOW DEVICE" command displays the following:

```
$ show devices
Device          Device          Error   Volume      Free  Trans Mnt
Name           Status          Count   Label        Blocks Count Cnt
DISKS$DUA0:    Mounted        0       DSSI01       32022  147  1
DISKS$DUA1:    Online         0

Device          Device          Error
Name           Status          Count
OPA0:          Online         0
FTA0:          Offline        0

Device          Device          Error
Name           Status          Count
PAA0:          Online         0
```

 When a tape or disk image connected to an emulated HSJ50 controller is dismounted in VAX/VMS, it is disconnected from CHARON-VAX and can be manipulated. It may even 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.



# Networking

## Table of Contents

- Configuration steps
- SGEC Ethernet Controller
  - interface
  - station\_address
  - rx\_fifo\_size
- DEQNA / DESQA / DELQA Ethernet Controller
  - address
  - interface
  - station\_address
  - rx\_fifo\_size
- DEMNA Ethernet Adapter
  - xmi\_node\_id
  - interface
  - station\_address
  - rx\_fifo\_size
- DEBNI Ethernet Adapter
  - vax\_bi\_node\_id
  - interface
  - station\_address
  - rx\_fifo\_size
- PMAD-AA TurboChannel Ethernet Adapter
  - interface
  - station\_address
  - rx\_fifo\_size
- 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
  - log
  - log\_flush\_period
  - legacy\_mode
- Supported Ethernet Q-bus Adapters for DECnet OSI (DECnet Plus)
- Ethernet Q-bus Adapter and Cluster configuration rules

## Configuration steps

To configure CHARON-VAX / CHARON-PDP networking, follow these 3 steps:

### 1. Load network adapter (if required)

If you are configuring a DEQNA, DESQA, DELQA, DEUNA, DELUA, DEMNA, DEBNI or PMADAA adapter, use the "load" command as shown below. CHARON-VAX 3100/96/98 and 4000/106/108 emulations automatically load SGEC (with the name "EZA") and therefore no "load" command is required.

#### Example:

```
load DELQA/DEQNA NIC
```

### 2. Load "packet\_port"

Load "packet\_port" to connect the SGEC, DEQNA, DESQA, DELQA, DEUNA, DELUA, DEMNA, DEBNI or PMADAA 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 SGEC, DEQNA, DESQA, DELQA, DEUNA, DELUA, DEMNA, DEBNI or PMADAA adapter to the "packet\_port" by setting the interface name.


**Example:**


```
set NIC interface = NDIS
```

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

**Examples:**

```
load packet_port/chnetwrk NIC1 interface="(disabled)"
load packet_port/chnetwrk NIC2 interface="connection:CHARON_NIC"
```

 Network booting is supported with exception of VAX6xxx models.

 Specification of "chnetwrk" can be omitted, but it is recommended to keep it for compatibility

## SGEC Ethernet Controller

The built-in SGEC controller emulator ("EZA") 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>Set the "station_address" when you need to configure a satellite (remotely booted) system that will run DECnet or when the migrated software uses the permanent address on the network adapter.</p> <p><b>Format:</b></p> <p>XX-XX-XX-XX-XX-XX</p> <p>or</p> <p>XX:XX:XX:XX:XX:XX</p> <p><b>Example:</b></p> <pre>set eza station_address="AF:01:AC:78:1B:CC"</pre>

## rx\_fifo\_size

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

### Example:

```
load packet_port/chnetwrk EZA0 interface = "connection:Charon"
set EZA interface = EZA0
set EZA station_address="0C:FE:35:AA:67:3B"
```

## DEQNA / DESQA / DELQA Ethernet Controller

CHARON-VAX / CHARON-PDP Q-bus systems provide support for DEQNA, DESQA, DELQA, DEUNA and DELUA Ethernet controllers.

Use the following command to load the instances of DEQNA, DESQA, DELQA, DEUNA or DELUA Ethernet controllers:

```
load DEQNA/DEQNA <logical name>
load DESQA/DEQNA <logical name>
load DELQA/DEQNA <logical name>
load DEUNA/DEUNA <logical name>
load DELUA/DEULA <logical name>
```

**Example:**

```
load DESQA/DEQNA XQA
```

DEQNA, DESQA, DELQA, DEUNA and DELUA offer the following configuration parameters that can be specified using the "set" command:

## address

<b>Parameter</b>	address
<b>Type</b>	Numeric
<b>Value</b>	<p>Specifies the CSR address. The address must be a valid QBUS and UNIBUS address in I/O space. Initial value is 017774440, which is the factory setting for DEQNA, DESQA and DELQA Ethernet controllers.</p> <p>Use the address parameter if loading several instances of DEQNA, DESQA, DELQA, DEUNA and DELUA Ethernet controllers.</p> <p>"address" parameter values must be unique for every instance of an DEQNA, DESQA, DELQA, DEUNA and DELUA Ethernet controller.</p> <p><b>Example:</b></p> <pre>load DEQNA/DEQNA XQA address=017774440 load DEQNA/DEQNA XQB address=017764460</pre>

## 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>"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>Set the "station_address" when you need to configure a satellite (remotely booted) system that will run DECnet or when the migrated software uses the permanent address on the network adapter.</p> <p><b>Format:</b></p> <p>XX-XX-XX-XX-XX-XX</p> <p>or</p> <p>XX:XX:XX:XX:XX:XX</p> <p><b>Example:</b></p> <pre>set QNA station_address="AF:01:AC:78:1B:CC"</pre>

## rx\_fifo\_size

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

### Example:

```
load DESQA/DEQNA QNA interface = QNA0
set QNA station_address="0C:FE:35:AA:67:3B"
load packet_port/chnetwrk QNA0 interface = "connection:Charon"
```

## DEMNA Ethernet Adapter

CHARON-VAX/66X0 systems provide support for the DEMNA Ethernet controller.

Use the following command to load an instance of the DEMNA Ethernet controller:

```
load DEMNA/DEMNA <logical name>
```

**Example:**

```
load DEMNA/DEMNA EXA
```

DEMNA Ethernet controllers offer the following configuration parameters that can be specified using the "set" command:

### xmi\_node\_id

<b>Parameter</b>	xmi_node_id
<b>Type</b>	Number
<b>Value</b>	Specifies the XMI slot in which the virtual DEMNA controller is placed. For CHARON-VAX/66X0 a free slot between 10 (A) and 14 (E) must be chosen.

### 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>Set the "station_address" when you need to configure a satellite (remotely booted) system which will run DECnet or when the migrated software uses the permanent address on the network adapter.</p> <p><b>Format:</b></p> <p>XX-XX-XX-XX-XX-XX</p> <p>or</p> <p>XX:XX:XX:XX:XX:XX</p> <p><b>Example:</b></p> <pre>set EXA station_address="AF:01:AC:78:1B:CC"</pre>

## rx\_fifo\_size

<b>Parameter</b>	rx_fifo_size
<b>Type</b>	Numeric
<b>Value</b>	<p>The "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>

**Example:**

```
load DEMNA/DEMNA EXA xmi_node_id = 11 interface = EXA0
set EXA station_address = "0C:FE:35:AA:67:3B"
load packet_port/chnetwrk EXA0 interface = "connection:Charon"
```

## DEBNI Ethernet Adapter

CHARON-VAX/63X0 systems provide support for the DEBNI Ethernet controller.

Use the following command to load an instance of the DEBNI Ethernet controller:

```
load DEBNI/DEMNA <logical name>
```

**Example:**

```
load DEBNI/DEMNA EXA
```

DEBNI Ethernet controllers offer the following configuration parameters that can be specified using the "set" command:

### vax\_bi\_node\_id

<b>Parameter</b>	vax_bi_node_id
<b>Type</b>	Number
<b>Value</b>	Specifies the VAXBI slot in which the virtual DEBNI controller is placed.

### 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>Set the "station_address" when you need to configure a satellite (remotely booted) system which will run DECnet or when the migrated software uses the permanent address on the network adapter.</p> <p><b>Format:</b></p> <p>XX-XX-XX-XX-XX-XX</p> <p>or</p> <p>XX:XX:XX:XX:XX:XX</p> <p><b>Example:</b></p> <pre>set EXA station_address="AF:01:AC:78:1B:CC"</pre>



## rx\_fifo\_size

<b>Parameter</b>	rx_fifo_size
<b>Type</b>	Numeric
<b>Value</b>	<p>The "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>

### Example:

```
load DEBNI/DEMNA EXA vax_bi_node_id = 11 interface = EXA0
set EXA station_address = "0C:FE:35:AA:67:3B"
load packet_port/chnetwrk EXA0 interface = "connection:Charon"
```

## PMAD-AA TurboChannel Ethernet Adapter

The CHARON-VAX VAXstation 4000 Model 90 system provides support for a PMAD-AA TurboChannel Ethernet controller (in addition to the preloaded SGEC "EZA").

Use the following command to load an instance of a PMAD-AA Ethernet controller:

```
load PMADAA/PMADAA <logical name>
```

### Example:

```
load PMADAA/PMADAA EXA
```

PMAD-AA TurboChannel Ethernet controllers use the following configuration parameters that can be specified using 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>Set the "station_address" when you need to configure a satellite (remotely booted) system which will run DECnet or when the migrated software uses the permanent address on the network adapter.</p> <p><b>Format:</b></p> <p>XX-XX-XX-XX-XX-XX</p> <p>or</p> <p>XX:XX:XX:XX:XX:XX</p> <p><b>Example:</b></p> <pre>set ECA station_address="AF:01:AC:78:1B:CC"</pre>

## rx\_fifo\_size

<b>Parameter</b>	rx_fifo_size
<b>Type</b>	Numeric
<b>Value</b>	<p>The "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 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>

### Example:

```
load PMADAA/PMADAA ECA interface = ECA0
set ECA station_address = "0C:FE:35:AA:67:3B"
load packet_port/chnetwrk ECA0 interface = "connection:Charon"
```

## 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 VAX/PDP11 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 pp_1
```

"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-VAX / CHARON-PDP with a keyword "connection" preceding it.</p> <p><b>Syntax:</b></p> <pre>set &lt;name&gt; interface="connection:&lt;adapter&gt;"</pre> <p><b>Example:</b></p> <pre>set pp_1 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 sets the appropriate Ethernet address automatically.</p> <p>If "false" is specified, set the Ethernet address manually.</p> <p><b>Example:</b></p> <pre>set pp_1 port_enable_mac_addr_change=false</pre>

## port\_retry\_on\_tx

<b>Parameter</b>	port_retry_on_tx
<b>Type</b>	Numeric
<b>Value</b>	<p>The "port_retry_on_tx" parameter controls the number of times a port will attempt to transmit a packet before giving up.</p> <p>By default, the value is 3.</p> <p>Increasing this value may introduce problems in carrier loss logic, because not all NIC drivers support a carrier status query.</p> <p>Typically, you do not need to increase the value.</p> <p><b>Example:</b></p> <pre>set pp_1 port_retry_on_tx=8</pre>

## port\_pending\_rx\_number

<b>Parameter</b>	port_pending_rx_number
<b>Type</b>	Numeric
<b>Value</b>	<p>The "port_pending_rx_number" parameter sets the number of pending receive buffers.</p> <p>The default value is 63. The maximum value allowed is 195.</p> <p>You may want to increase the "port_pending_rx_number" when you have very busy networking and experience problems like losing connections not related to the carrier loss.</p> <p>Typically, you do not need to change this parameter.</p> <p><b>Example:</b></p> <pre>set pp_1 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 pp_1 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 pp_1 suspend_msg_on_mac_change=false</pre>


## log

<b>Parameter</b>	log
<b>Type</b>	Text string
<b>Value</b>	<p>If this parameter is set to some valid file name or a directory where the log files for each individual session will be stored CHARON VM logs Recv and Xmit packets at the emulated port layer.</p> <p>If an existing directory is specified, CHARON VM 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 VM does not create log.</p> <p>In certain situations enabling this parameter may help to detect loss of packets.</p> <p><b>Example 1:</b></p> <pre>set pp_1 log="pp_1.log"</pre> <p><b>Example 2:</b></p> <pre>set pp_1 log="C:\Charon\Logs"</pre> <p> Only existing directory can be used as a value of the "log" parameter.</p>

## log\_flush\_period


<b>Parameter</b>	log_flush_period
<b>Type</b>	Numeric
<b>Value</b>	<ul style="list-style-type: none"> <li>• &lt;period-in-seconds&gt;</li> </ul> <p>Defines a period of flushing log to disk.</p> <p>Default period is 60 seconds (it means that every minute log file is flushed to disk)</p> <p><b>Examples:</b></p> <pre>set pp_1 log_flush_period=30</pre>

## legacy\_mode

<b>Parameter</b>	legacy_mode
<b>Type</b>	Boolean
<b>Value</b>	<p>If this parameter is set to "true" CHARON-VAX 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-VAX runs in VMware environment for example and you do not want to allow promiscuous mode.</p> <p>In other situations please do not change its default value ("false").</p> <p><b>Example:</b></p> <pre>set pp_1 legacy_mode=true</pre> <p> This parameter is required only if you do not want to allow promiscuous traffic for the VMware adapter/port. In this mode, CHARON-VAX 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;</li> <li>2. Enable the promiscuous traffic;</li> <li>3. Do not use legacy_mode option in Charon's configuration file</li> </ol>

**Example:**


```
load DEQNA/DEQNA XQA
load packet_port/chnetwrk XQA0 interface="connection:Charon"
set XQA interface=XQA0
```

-  CHARON 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.

## Supported Ethernet Q-bus Adapters for DECnet OSI (DECnet Plus)

**The only supported Q-bus device is the DELQA adapter model.**

During DECnet OSI installation or DECnet OSI interface reconfiguration, DEQNA and DESQA models are not recognized as valid devices. (DECnet OSI SPD).

-  The integrated SGEC Ethernet "EZA" on a MicroVAX 3100 & VAX 4000-106/108 is supported by DECnet OSI.  
The integrated ESA (AMD Lance 7990) device is not currently implemented in CHARON-VAX.

## Ethernet Q-bus Adapter and Cluster configuration rules

In a VMS Cluster using VMS version 5.5 and above, use only DELQA and DESQA Ethernet adapters that are supported (VMS Cluster SPD).

The default DEQNA device is not supported for SCS Cluster protocol and the emulated VAX will fail with a CLUEXIT Bugcheck.

# Specific Controllers


## Table of Contents

- [LPV11 Line Printer Controller](#)
  - [address](#)
  - [vector](#)
  - [parameters](#)
- [DUMMY\\_VCB02 device](#)

## LPV11 Line Printer Controller

CHARON-VAX / CHARON-PDP Q-bus systems provide support for the LPV11 Parallel Line Printer Controller.

The LPV11 emulation is implemented using the CHARON Qbus API (CHAPI). This interface is designed to allow third-parties to custom design emulated Qbus peripherals.

 To enable CHAPI modules, request an update to the CHARON-VAX / CHARON-PDP license. For details on CHAPI read the CHARON-VAX / CHARON-PDP CHAPI user manual, available on demand or from the STROMASYS web site for our registered partners.

Use the following command to load an instance of an LPV11 Parallel Line Printer Controller:

```
load CHAPI <logical name> dll=LPV11
```

**Example:**

```
load CHAPI LPA0 dll=LPV11
```

The LPV11 (being a CHAPI component) provides the following configuration parameters, to be specified using the "set" command:

### address

<b>Parameter</b>	address
<b>Type</b>	Numeric
<b>Value</b>	Specifies the CSR address. This address must be a valid QBUS 22-bit address in an I/O space. The initial value is 017777514, which is the factory setting for an LPV11.

### vector

<b>Parameter</b>	vector
<b>Type</b>	Numeric
<b>Value</b>	Specifies the interrupt vector. The initial value is 0200, which is the factory setting for an LPV11.




## parameters

<b>Parameter</b>	<i>parameters</i>
<b>Type</b>	Text String
<b>Value</b>	<p>Specifies the name of the file in which the LPV11 prints the text, the name of a physical printer or a TCPIP port to connect to an external utility, like <a href="#">HOSTPrint</a>, (along with some additional parameters):</p> <ul style="list-style-type: none"> <li>■ "\\.\LPT&lt;N&gt;" corresponds to the host port to connect to a physical printer</li> <li>■ "port" corresponds to the TCP/IP port to connect to an external application</li> <li>■ "application" corresponds to the application to run after opening the port. <u>Note</u>: the name of the application including its own parameters must be framed within ' characters (apostrophe).</li> <li>■ -port corresponds to the TCP/IP port the <a href="#">HOSTprint</a> utility will connect to.</li> <li>■ -fontsize corresponds to the font size to be used for printing.</li> <li>■ -font corresponds to the font to be used for printing. The font name must be framed with \ (backslash) symbols.</li> </ul>

It's possible to load as many LPV11 devices as you wish and the system running on the emulated VAX/PDP-11 allows this to be done. To load LPV11 devices use the standard QBUS parameters: "address" and "vector" to select the correct place on the QBUS.

The following example demonstrates how to load 2 instances of LPV11, mapped to 2 host system LPT ports:

```
load chapi lpv1 address = 017777514 vector = 0200
set lpv1 dll = lpv11.dll parameters = "\\.\LPT1"
load chapi lpv2 address = 017764004 vector = 0170
set lpv2 dll = lpv11.dll parameters = "\\.\LPT2"
```

 The printers connected to the host must understand output of the LPV11 driver on the emulated VAX/PDP-11. Use printers that work with the VAX/PDP-11 hardware.


The "configure" utility (VAX), available from the SRM console, is used to find out the correct QBUS addresses and interrupt vectors for multiple LPV11 instances.

It is also possible to redirect output from an LPV11 to an external application, for example, [HOSTPrint](#). This application allows printing to the Windows system printers using the desired font.

In the following example the LPV11 instance, called: lpv1, opens host port 10004 and starts the [HOSTPrint](#) application, directing it to connect to the port 10004 and use the font "Courier New", size 12 for printing:

```
load chapi lpv1 address = 017777514 vector = 0200
set lpv1 dll = lpv11.dll parameters="port = 10004 application='HOSTPrint -port=10004 -fontsize=12 -font=\Courier New\'"
```


## DUMMY\_VCB02 device

 This description applies only to MicroVAX 3600, MicroVAX 3900, VAXserver 3900 and VAXserver 3600 server emulations.

The dummy VCB02 device emulates the QBUS registers of a VCB02 graphic controller (at the secondary VCB02 device address) but does not implement graphic display functionality. Its purpose is to be able to use a workstation version of VMS. When this device is loaded, VAX/VMS requires type "D" licenses (Workstation). As the dummy VCB02 is located on the secondary address (HEX: 0x3fff02 OCT: 01777402), VMS boots using the normal console terminal without graphics capability.

The format is as follows:

```
load chapi dummy_vcb02 dll=chapi_vcb02
set dummy_vcb02 address = 017777402 vector = 0320 display = null
```

 As this dummy device cannot draw graphics, Charon-VAX can fail the initial hardware diagnostics but will boot correctly.

# Sample configuration files

## Contents

- PDP-11/93 configuration file
- VAX 4000 Model 108 configuration file
- VAX 6310 configuration file
- VAX 6610 configuration file
- "configuration\_name.icfg" configuration file

## PDP-11/93 configuration file

```

#
# Copyright (C) 1999-2018 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.
#
#=====
set session hw_model = PDP1193

#=====
#
# Choose a name for the instance, if needed, to differentiate it among other
# instances running on the same host.
#
#-----

include configuration_name.icfg

#=====
#
# Use the following commands to disable the rotating LOG files and enable
# a single LOG file. Select either append or overwrite (for each time the
# instance starts) and specify desired log path and file name.
#
#-----

set session log_method = append
#set session log_method = overwrite
#set session log = PDP1193.log

set toy container="pdp1193.dat"

#
# Specify KW11-L timer frequency here (default is 50). Note, that appropriate
# disk image supports appropriate frequency because system clock
# frequency is the system configuration parameter.
#

#set KW11 frequency = 50
#set KW11 frequency = 60

#
# Now assign console built-in serial line. Currently the emulator offers two
# possible ways using serial lines. First of them is connection to COM ports
# (via physical_serial_line). The second is to attach a third party terminal
# emulator (virtual_serial_line).
#

#load physical_serial_line TT0 line="\\.\COM1"
#load virtual_serial_line TT0 port=10003
#load virtual_serial_line TT0 port=10003 application="opa0.ht"
load virtual_serial_line TT0 port=10003 application="putty -load OPA0"
set uart line=TT0

#
# Configure an optional RQDX3 storage controller (MSCP/QBUS). It handles disk
# images, disk drives, CD-ROM drives, magneto-optical drives, floppy drives.
#

#load RQDX3 DU
#set DU container[0]="..."
#set DU container[1]="..."
#set DU container[2]="..."
#set DU container[3]="..."

# Boot from DU0 MSCP disk
#set cpu_0 auto_boot="DU0"

```

```
# this is the end of the configuration file #####
```

## VAX 4000 Model 108 configuration file

```

#
# Copyright (C) 1999-2018 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 VAX 4000 Model 108.
#
# Specify the hw_model prior to any other commands. This parameter informs
# the emulator what type of VAX it should build and enables all other
# commands.
#
#-----

set session hw_model = VAX_4000_Model_108

#=====
#
# Choose a name for the instance, if needed, to differentiate it among other
# instances running on the same host.
#
#-----

include configuration_name.icfg

#=====
#
# Use the following commands to disable the rotating LOG files and enable
# a single LOG file. Select either append or overwrite (for each time the
# instance starts) and specify desired log path and file name.
#
#-----

set session log_method = append
#set session log_method = overwrite
#set session log = VAX_4000_Model_108.log

#=====
#
# The following line tells the emulator where to preserve NVRAM content.
# The file maintains the current time of the emulated VAX (when it is not
# running) and other console parameters (such as default boot device).
#
#-----

#set toy container="vx4k108.dat"

#=====
#
# The following line tells the emulator where to store intermediate state
# of the Flash ROM and console parameters. It is highly recommended to enable
# this and the previous toy file for the emulator to be able to correctly
# preserve the saved state of the console.
#
#-----

#set rom container="vx4k108.rom"

#=====
#
# Disable or enable dynamic instruction translation by the cpu (ACE). The use
# of DIT may be also prohibited by the license. If not specified (i.e. when
# both lines remain commented out) the DIT is enabled as soon as the license

```

```

# allows to do so and is disabled otherwise ...
#
#-----

#set cpu ace_mode=false
#set cpu ace_mode=true

#=====
#
# Specify the size of RAM (default is 16MB). Note that DIT (when enabled)
# also needs certain amount of memory which grows linearly following the size
# of memory specified here. Also remember that the dongle license might limit
# the maximum amount of memory.
#
#-----

#set ram size=32
#set ram size=64
#set ram size=80
#set ram size=128
#set ram size=256
#set ram size=512

#=====
#
# Assign four built-in serial lines as necessary. Currently the emulator
# offers two ways to use built-in serial lines:
#
# 1) connection to COM ports (via physical_serial_line) and
# 2) attach a third party terminal emulator (virtual_serial_line).
#
# Once the desired connection is chosen and the corresponding line is
# enabled, connect it to the preloaded controller QUART by choosing the QUART
# line number (in square brackets). See OPA0 example, below.
#
#-----

#load physical_serial_line TTA0 line="\\.\COMn"
#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.exe -load TTA0"
#set quart line[0]=TTA0

#load physical_serial_line TTA1 line="\\.\COM1"
#load virtual_serial_line TTA1 port=10001
#load virtual_serial_line TTA1 port=10001 application="tta1.ht"
#load virtual_serial_line TTA1 port=10001 application="putty.exe -load TTA1"
#set quart line[1]=TTA1

#load physical_serial_line TTA2 line="\\.\COM2"
#load virtual_serial_line TTA2 port=10002
#load virtual_serial_line TTA2 port=10002 application="tta2.ht"
#load virtual_serial_line TTA2 port=10002 application="putty.exe -load TTA2"
#set quart line[2]=TTA2

#=====
#
# Select the connection method for the console serial line OPA0.
#
#-----

#load physical_serial_line OPA0 line="\\.\COM3"
#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 -P 10003"
set quart line[3]=OPA0

#-----
#
# Uncomment to allow 'F6' to terminate the running emulator. Closing console
# satellite application may also be used for that.
#
#-----

```

```

#set OPA0 stop_on = "F6"
#set OPA0 stop_on = "Application"
#set OPA0 stop_on = "F6, Application"

#=====
#
# The VAX 4000 Model 108 contains built-in PCI SCSI adapter called PKA
# within the configuration file.
#
#-----
#
# Uncomment to connect the emulator's DKA0 to the disk image.
#
#-----

#load virtual_scsi_disk pka_0 scsi_bus=pka scsi_id=0
#set pka_0 container="<file-name>.vdisk"

#=====
#
# Uncomment to connect the emulator's DKA100 to a host disk drive.
#
#-----

#load virtual_scsi_disk pka_1 scsi_bus=pka scsi_id=1
#set pka_1 container="\\.\PhysicalDrive0"
#set pka_1 container="\\.\PhysicalDriveN"

#=====
#
# Uncomment to connect the emulator's DKA300 to the host's CD/DVD-ROM drive.
#
#-----

#load virtual_scsi_cdrom pka_3 scsi_bus = pka scsi_id = 3
#set pka_3 container="\\.\CdRom0"
#set pka_3 container="\\.\CdRomN"

#=====
#
# Uncomment to connect the emulator's DKA400 to an .ISO file (CD/DVD image).
#
#-----

#load virtual_scsi_cdrom pka_4 scsi_bus=pka scsi_id=4
#set pka_4 container="<file-name>.iso"

#=====
#
# Uncomment to connect the emulator's MKA500 to the host's SCSI tape drive.
#
#-----

#load virtual_scsi_tape pka_5 scsi_bus=pka scsi_id=5
#set pka_5 container="\\.\Tape0"
#set pka_5 container="\\.\TapeN"

#=====
#
# Uncomment to connect the emulator's MKA600 to a .VTAPE file (tape image).
#
#-----

#load virtual_scsi_tape pka_6 scsi_bus=pka scsi_id=6
#set pka_6 container="<file-name>.vtape"

#=====
#
# If necessary, load an optional SCSI controller SCSI_B (PKB).
#
# ATTENTION! Old versions of VAX/VMS (older than 5.5-2H4) do not support
# the optional SCSI controller and may fail to boot if this option is loaded.
#
#-----

```



```

#include kzdda.icfg

#=====
#
# Uncomment to connect the emulator's DKB600, DKB601 to the host's floppy disk
# drives (A:, B:, if any).
#
#-----

#load floppy_scsi_device pkb_6 scsi_bus=pkb scsi_id=6

#=====
#
# Enable the built-in SGEN Ethernet Adapter (EZA).
#
# TIP: You need to uncomment the "set EZA ..." line and one of the
# "load packet_port ..." lines below to attach the EZA to host NIC (or not)
#
#-----

#set EZA interface = EZA0

# choose this one to leave EZA unconnected
#load packet_port EZA0 interface = "(disabled)"

# choose this one to connect EZA to host's NIC (by its connection name)
#load packet_port EZA0 interface = "connection:<connection-name>"

#=====
#
# Load an optional DHW42-AA (or DHW42-BA, or DHW42-CA) serial line controller
# (C-DAL).
#
# Only one instance of DHW42AA/BA/CA can be loaded.
#
#-----

#load DHW42AA/DHV11 TXA
#load DHW42BA/DHV11 TXA
#load DHW42CA/DHV11 TXA

#load physical_serial_line TXA0 line="\\.\COMn"
#load virtual_serial_line TXA0 port=10010
#load virtual_serial_line TXA0 port=10010 application="txa0.ht"
#set TXA line[0]=TXA0

#load physical_serial_line TXA1 line="\\.\COMn"
#load virtual_serial_line TXA1 port=10011
#load virtual_serial_line TXA1 port=10011 application="txa1.ht"
#set TXA line[1]=TXA1

#load physical_serial_line TXA2 line="\\.\COMn"
#load virtual_serial_line TXA2 port=10012
#load virtual_serial_line TXA2 port=10012 application="txa2.ht"
#set TXA line[2]=TXA2

#load physical_serial_line TXA3 line="\\.\COMn"
#load virtual_serial_line TXA3 port=10013
#load virtual_serial_line TXA3 port=10013 application="txa3.ht"
#set TXA line[3]=TXA3

#load physical_serial_line TXA4 line="\\.\COMn"
#load virtual_serial_line TXA4 port=10014
#load virtual_serial_line TXA4 port=10014 application="txa4.ht"
#set TXA line[4]=TXA4

#load physical_serial_line TXA5 line="\\.\COMn"
#load virtual_serial_line TXA5 port=10015
#load virtual_serial_line TXA5 port=10015 application="txa5.ht"
#set TXA line[5]=TXA5

#load physical_serial_line TXA6 line="\\.\COMn"
#load virtual_serial_line TXA6 port=10016
#load virtual_serial_line TXA6 port=10016 application="txa6.ht"
#set TXA line[6]=TXA6

```

```

#load physical_serial_line TXA7 line="\\.\COMn"
#load virtual_serial_line TXA7 port=10017
#load virtual_serial_line TXA7 port=10017 application="txa7.ht"
#set TXA line[7]=TXA7

#####
#
# Configure an optional RQDX3 storage controller (MSCP/QBUS). It handles disk
# images, disk drives, CD-ROM drives, magneto-optical drives, floppy drives.
#
#-----

#load RQDX3 DUA

#set DUA container[0]="<file-name>.vdisk"
#set DUA container[1]="\\.\PhysicalDriveN"
#set DUA container[2]="\\.\CdRomN"
#set DUA container[3]="<file-name>.iso"
#set DUA container[4]="\\.\H:"

#load RQDX3 DUB address=...
#load RQDX3 DUC address=...

#####
#
# Configure an optional TQK50 tape storage controller (TMSCP/QBUS). It handles
# tape images, and physical tape drives attached to the host.
#
#-----

#load TQK50 MUA

#set MUA container[0]="<file-name>.vtape"
#set MUA container[1]="\\.\TapeN"

#load TQK50 MUB address=...
#load TQK50 MUC address=...

#####
#
# Load optional DELQA QBUS Ethernet Adapter (XQA).
#
# TIP: You need to uncomment the "load DELQA ..." line and one of the
# "load packet_port ..." lines below to attach the XQA to host NIC (or not)
#
#-----

#load DELQA XQA interface = XQA0

# choose this one to leave XQA unconnected
#load packet_port XQA0 interface = "(disabled)"

# choose this one to connect XQA to host's NIC (by its connection name)
#load packet_port XQA0 interface = "connection:<connection-name>"

#####
#
# Configure an optional DHV11 (or DHQ11, CXY08, CXA16, CXB16) serial line
# controller (QBUS). The address and vector must be set as required by the
# guest operating system.
#
#-----

#load DHV11/DHV11 TXA
#load DHQ11/DHV11 TXA
#load CXY08/DHV11 TXA
#load CXA16/DHV11 TXA
#load CXB16/DHV11 TXA

#load physical_serial_line TXA0 line="\\.\COMn"
#load virtual_serial_line TXA0 port=10010
#load virtual_serial_line TXA0 port=10010 application="txa0.ht"
#set TXA line[0]=TXA0

#load physical_serial_line TXA1 line="\\.\COMn"
#load virtual_serial_line TXA1 port=10011

```

```
#load virtual_serial_line TXA1 port=10011 application="txa1.ht"
#set TXA line[1]=TXA1

#load physical_serial_line TXA2 line="\\.\COMn"
#load virtual_serial_line TXA2 port=10012
#load virtual_serial_line TXA2 port=10012 application="txa2.ht"
#set TXA line[2]=TXA2

#load physical_serial_line TXA3 line="\\.\COMn"
#load virtual_serial_line TXA3 port=10013
#load virtual_serial_line TXA3 port=10013 application="txa3.ht"
#set TXA line[3]=TXA3

#load physical_serial_line TXA4 line="\\.\COMn"
#load virtual_serial_line TXA4 port=10014
#load virtual_serial_line TXA4 port=10014 application="txa4.ht"
#set TXA line[4]=TXA4

#load physical_serial_line TXA5 line="\\.\COMn"
#load virtual_serial_line TXA5 port=10015
#load virtual_serial_line TXA5 port=10015 application="txa5.ht"
#set TXA line[5]=TXA5

#load physical_serial_line TXA6 line="\\.\COMn"
#load virtual_serial_line TXA6 port=10016
#load virtual_serial_line TXA6 port=10016 application="txa6.ht"
#set TXA line[6]=TXA6

#load physical_serial_line TXA7 line="\\.\COMn"
#load virtual_serial_line TXA7 port=10017
#load virtual_serial_line TXA7 port=10017 application="txa7.ht"
#set TXA line[7]=TXA7

#load DHV11 TXB address=...
#load DHQ11 TXB address=...
#load CXY08 TXB address=...
#load CXA16 TXB address=...
#load CXB16 TXB address=...

# this is the end of the configuration file #####
```

## VAX 6310 configuration file

```

#
# Copyright (C) 1999-2018 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 VAX 6000 Model 310.
#
#-----

set session hw_model = VAX_6310

#=====
#
# Choose a name for the instance, if needed, to differentiate it among other
# instances running on the same host.
#
#-----

include configuration_name.icfg

#=====
#
# Use the following commands to disable the rotating LOG files and enable
# a single LOG file. Select either append or overwrite (for each time the
# instance starts) and specify desired log path and file name.
#
#-----

set session log_method = append
#set session log_method = overwrite
#set session log = VAX_6310.log

#=====
#
# The following line tells the emulator where to preserve NVRAM content.
# The TOY file maintains the current time of the emulated VAX (when it is not
# running) and other console parameters (such as default boot device).
#
# Both files must be enabled to correctly preserve the console settings.
#
#-----

#set toy container = "charon.dat"
#set eeprom container = "charon.rom"

#=====
#
# Disable or enable dynamic instruction translation by the cpu (ACE). The use
# of DIT may be also prohibited by the license. If not specified (i.e. when
# both lines remain commented out) the DIT is enabled as soon as the license
# allows to do so and is disabled otherwise ...
#
#-----

#set cpu ace_mode=false
#set cpu ace_mode=true

#=====
#
# Specify the size of RAM (default 32MB). Note that the dongle license might
# limit the maximum amount of memory.
#
#-----

```

```

#set ram size = 64
#set ram size = 128
#set ram size = 256
#set ram size = 512

#####
#
# Select the connection method 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 -P 10003"

#-----
#
# Uncomment to allow 'F6' to terminate the running emulator. Closing console
# satellite application may also be used for that.
#
#-----

#set OPA0 stop_on = "F6"
#set OPA0 stop_on = "Application"
#set OPA0 stop_on = "F6, Application"

#####
#
# DWMBB XMI VAXBI Adapter
#
# Slots 1 - 15 (1 - F) of the VAXBI are able to handle I/O adapters.
#
#-----

load DWMBB XBA xmi_node_id = 14

#####
#
# KDB50 VAXBI Disk Controller
#
#-----

#load KDB50 DUA vax_bi_node_id = 1

#####
#
# Uncomment to connect the emulator's DUA0 to the disk image.
#
#-----

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

#####
#
# Uncomment to connect the emulator's DUA1 to the physical disk drive.
#
#-----

#set DUA container[1] = "\\.\PhysicalDriveN"
#set DUA container[1] = "\\.\A:"

#####
#
# Uncomment to connect the emulator's DUA9 to host's CD/DVD-ROM drive or
# to an .ISO file (CD/DVD image).
#
#-----

#set DUA container[9] = "<file-name>.iso"
#set DUA container[9] = "\\.\CdRomN"

#####
#
# Load optional DEBNI VAXBI Ethernet Adapter (ETA). Allocate slot 2 of VAXBI.

```

```

#
# TIP: You need to uncomment the "load DEBNI ..." line and one of the
# "load packet_port ..." lines below to attach the ETA to host NIC (or not)
#
#-----

#load DEBNI ETA vax_bi_node_id = 2 interface = ETA0

# choose this one to leave ETA unconnected
#load packet_port ETA0 interface = "(disabled)"

# choose this one to connect ETA to host's NIC (by its connection name)
#load packet_port ETA0 interface = "connection:<connection-name>"

#=====
#
# DWBUA VAXBI UNIBUS Adapter
#
#-----

load DWBUA UBA vax_bi_node_id = 14

#=====
#
# TUK50 UNIBUS Tape Controller
#
#-----

#load TUK50 MUA

#=====
#
# Uncomment to connect the emulator's MUA0 to the physical tape drive or
# to a .VTAPE file (tape image).
#
#-----

#set MUA container[0] = "<file-name>.vtape"
#set MUA container[0] = "\\.\TapeN"

# this is the end of the configuration file #####

```

# VAX 6610 configuration file

```

#
# Copyright (C) 1999-2018 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 VAX 6000 Model 610.
#
#-----

set session hw_model = VAX_6610

#=====
#
# Choose a name for the instance, if needed, to differentiate it among other
# instances running on the same host.
#
#-----

include configuration_name.icfg

#=====
#
# Use the following commands to disable the rotating LOG files and enable
# a single LOG file. Select either append or overwrite (for each time the
# instance starts) and specify desired log path and file name.
#
#-----

set session log_method = append
#set session log_method = overwrite
#set session log = VAX_6610.log

#=====
#
# To enable automatic boot, define the default boot device in the VAX
# console and uncomment the line below.
#
#-----

#set xmi boot = auto

#=====
#
# The following line tells the emulator where to preserve NVRAM content.
# The TOY file maintains the current time of the emulated VAX (when it is not
# running) and other console parameters (such as default boot device).
#
# Both files must be enabled to correctly preserve the console settings.
#
#-----

#set toy container = "charon.dat"
#set eeprom container = "charon.rom"

#=====
#
# Disable or enable dynamic instruction translation by the cpu (ACE). The use
# of DIT may be also prohibited by the license. If not specified (i.e. when
# both lines remain commented out) the DIT is enabled as soon as the license
# allows to do so and is disabled otherwise ...
#
#-----

```

```

#set cpu ace_mode=false
#set cpu ace_mode=true

#-----
#
# Specify the size of RAM (default 512MB). Note that DIT (when enabled)
# also needs certain amount of memory which grows linearly following
# the size of memory specified here. Also remember that the dongle
# license might limit the maximum amount of memory.
#
# The valid settings are: 32,64,128,256,512,768,1024, ... 3072,
#
#-----

#set ram size = 512
#set ram size = 768
#set ram size = 1024
#set ram size = 2048
#set ram size = 3072

#-----
#
# Select the connection method 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 -P 10003"
#load virtual_serial_line OPA0 port = 10003 application = "c:\kea\user\opa0.ktc"

#-----
#
# Uncomment to allow 'F6' to terminate the running emulator. Closing console
# satellite application may also be used for that.
#
#-----

#set OPA0 stop_on = "F6"
#set OPA0 stop_on = "Application"
#set OPA0 stop_on = "F6, Application"

#-----
#
# Uncomment to enable emulation of KDM70 storage controller.
#
#-----

#load KDM70 PUA xmi_node_id = 11

#-----
#
# Uncomment to connect the emulator's DUA0 to a .VDISK file (disk image).
#
#-----

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

#-----
#
# Uncomment to connect the emulator's DUA1 to a host disk drive.
#
#-----

#set PUA container[1] = "\\.\PhysicalDrive1"
#set PUA container[1] = "\\.\PhysicalDrive<N>"

#-----
#
# Uncomment to connect the emulator's DUA3 to an .ISO file (CD/DVD image).
#
#-----

```



```

#set PUA container[3] = "<file-name>.iso"

#####
#
# Uncomment to connect the emulator's DUA4 to the host's CD/DVD-ROM drive.
#
#-----

#set PUA container[4] = "\\.\CdRom0"
#set PUA container[4] = "\\.\CdRom<N>"

#####
#
# Uncomment to connect the emulator's MUA5 to a .VTAPE file (tape image).
#
#-----

#set PUA container[5] = "<file-name>.vtape"

#####
#
# Uncomment to connect the emulator's MUA6 to a host tape drive.
#
#-----

#set PUA container[6] = "\\.\Tape0"
#set PUA container[6] = "\\.\Tape<N>"

#####
#
# Support of CI:
#
# Load CIXCD adapter into slot 12 (C) of the XMI.
#
#-----

#load CIXCD PAA xmi_node_id = 12 ci_node_id = 0x01

#####
#
# Support of CI:
#
# Connect HSJ50 storage controller to the CIXCD adapter PAA.
#
#-----

#load HSJ50 PUA ci_node_id = 0x0B mscp_allocation_class = 1

#####
#
# Uncomment to connect the emulator's DUA0 to the disk image.
#
#-----

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

#####
#
# Uncomment to connect the emulator's DUA1 to a host disk drive.
#
#-----

#set PUA container[1] = "\\.\PhysicalDrive<N>"

#####
#
# Load optional DEMNA XMI Ethernet Adapter (EXA). Allocate slot 13 of XMI.
#
# TIP: You need to uncomment the "load DEMNA ..." line and one of the
# "load packet_port ..." lines below to attach the EXA to host NIC (or not)
#
#-----

#load DEMNA EXA xmi_node_id = 13 interface = EXA0

```

```
# choose this one to leave EXA unconnected
#load packet_port EXA0 interface = "(disabled)"

# choose this one to connect EXA to host's NIC (by its connection name)
#load packet_port EXA0 interface = "connection:<connection-name>"

#=====
#
# Load optional DEMNA XMI Ethernet Adapter (EXB). Allocate slot 14 of XMI.
#
# TIP: You need to uncomment the "load DEMNA ..." line and one of the
# "load packet_port ..." lines below to attach the EXB to host NIC (or not)
#
#-----

#load DEMNA EXB xmi_node_id = 14 interface = EXB0

# choose this one to leave EXB unconnected
#load packet_port EXB0 interface = "(disabled)"

# choose this one to connect EXB to host's NIC (by its connection name)
#load packet_port EXB0 interface = "connection:<connection-name>"

# this is the end of the configuration file #####
```

## "configuration\_name.icfg" configuration file

```
#
# Copyright (C) 1999-2019 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.
#
#=====
#set session configuration_name = My_Virtual_Machine
# this is the end of the configuration file #####
```

# CHARON-VAX/PDP for Windows deinstallation

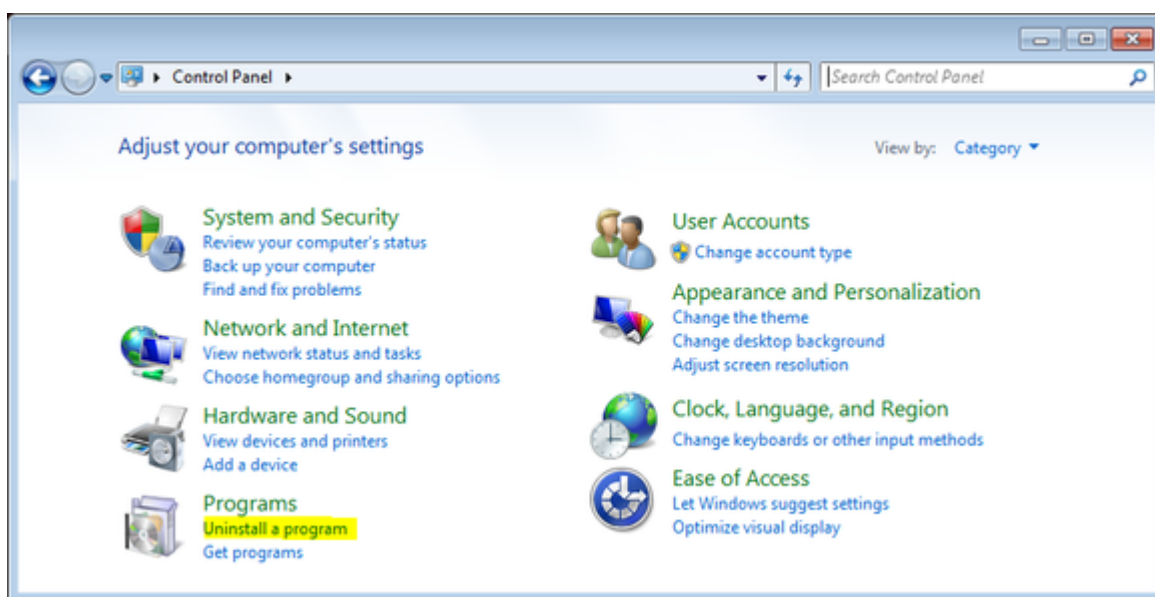
## Deinstallation procedure

Perform a clean shutdown of the running guests, open the CHARON Virtual Machine Manager, stop all running CHARON-VAX / CHARON-PDP virtual machines (VM) and remove them.

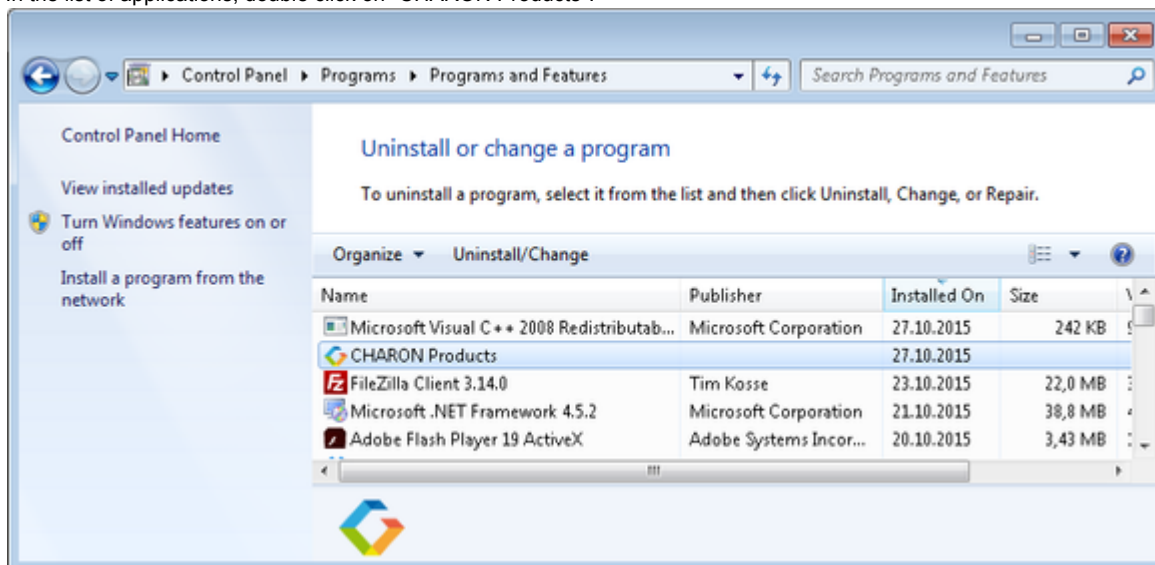
In case of a complete deinstallation:

- Remove all the CHARON virtual machines.
- Using the "Network Control Center" utility that can be opened from the "CHARON Virtual Machine Manager", release all the network interfaces back to the CHARON host.
- Remove any specific CHARON drivers installed manually (if any).

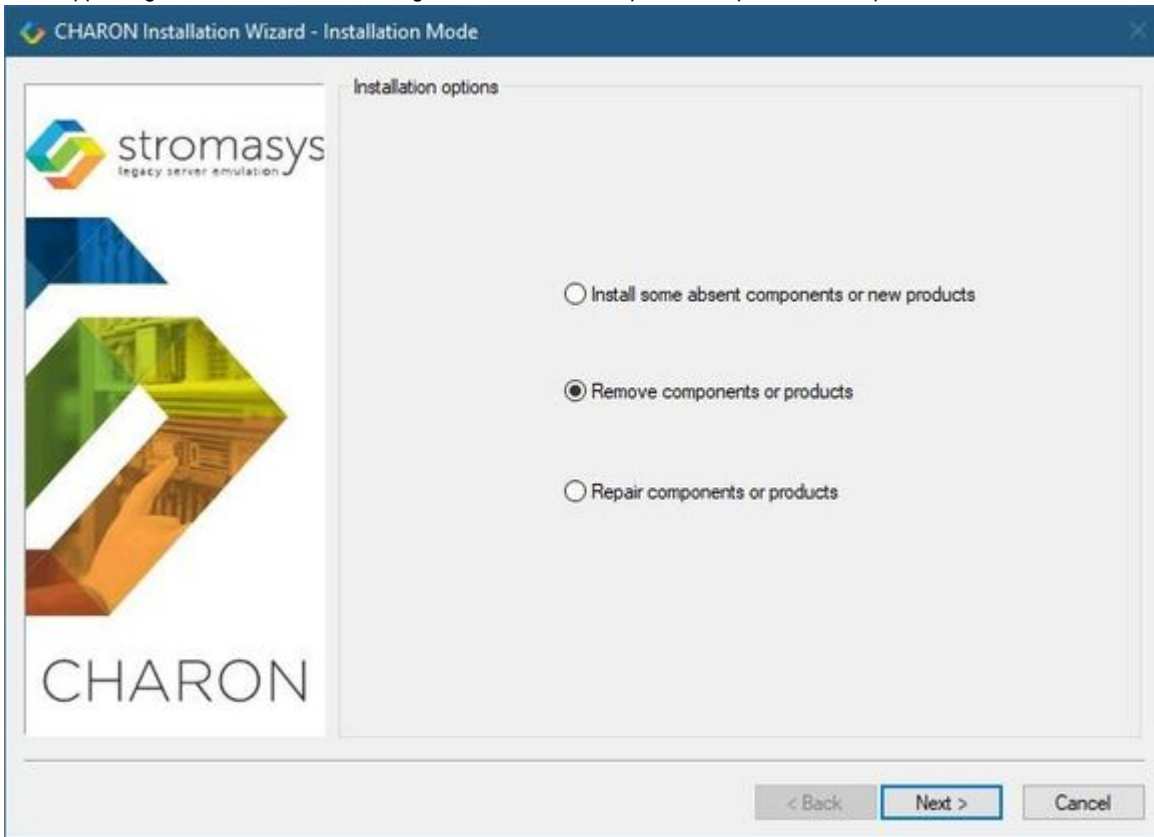
Open up the "Control Panel" and select "Uninstall a program":



In the list of applications, double click on "CHARON Products":

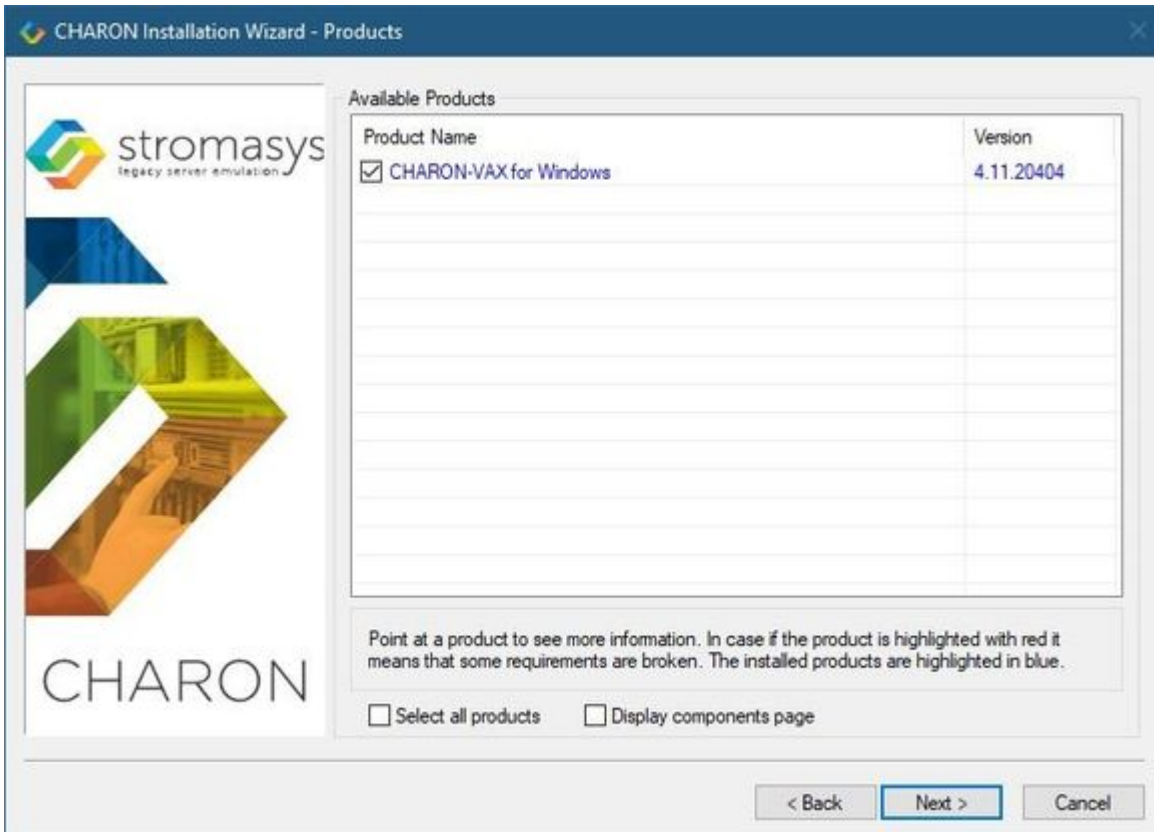


In the appearing CHARON installation dialog, select "Remove components or products" and press the "Next" button:

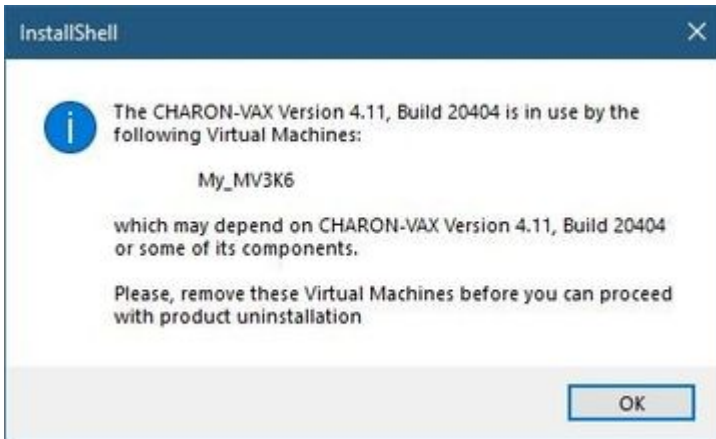


In the following dialog, select the CHARON product to be removed (or changed). Check "Select all the products" for deinstallation (or changing) of all installed CHARON products. Do not select "Display components page" unless you need to remove just some components of the CHARON product.

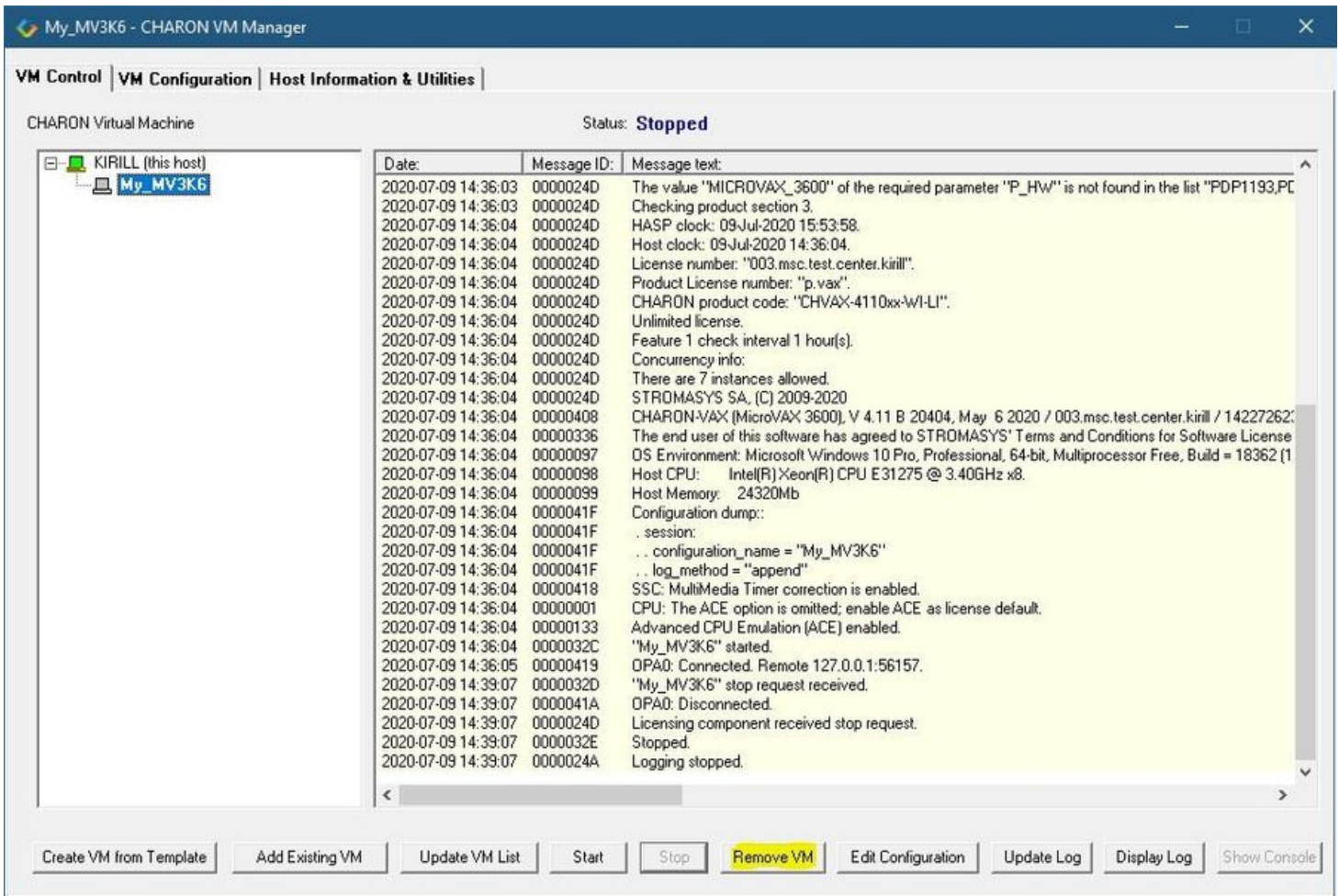
Press the "Next" button:



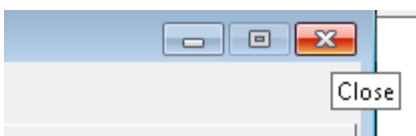
If there are some CHARON virtual machines that use this particular version of CHARON the following error message will be displayed:



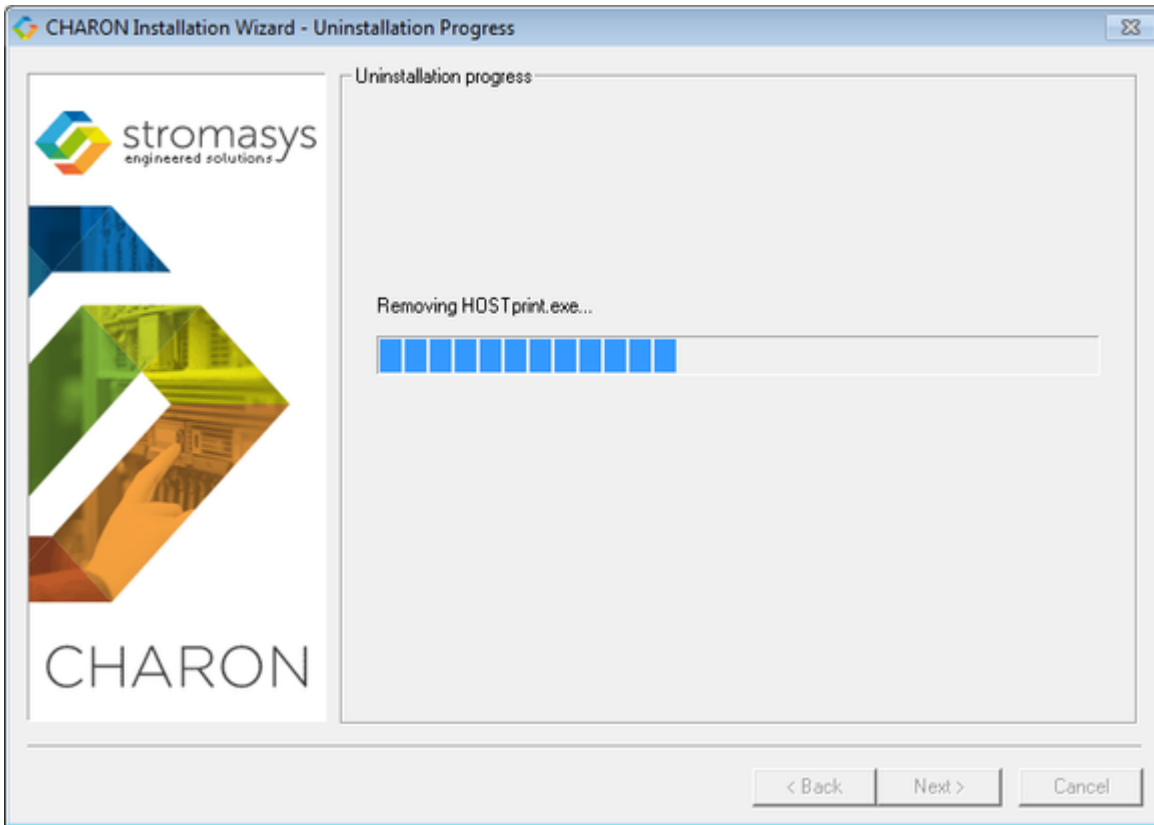
In this case, open "CHARON Virtual Machine Manager", select the CHARON Machine(s) listed above and either press the "Remove VM" button or open the VM configuration tab and set this virtual machine to use another Charon version if more than one version is installed:



Close the "CHARON Virtual Machine Manager" with a click on the "x" button and confirm you want to exit:



Press the "Next" button, the uninstallation process will begin:

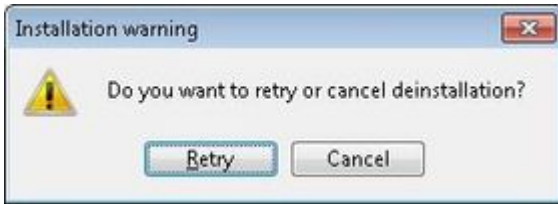


If "CHARON Virtual Machine Manager" is still running the following dialog will be displayed:

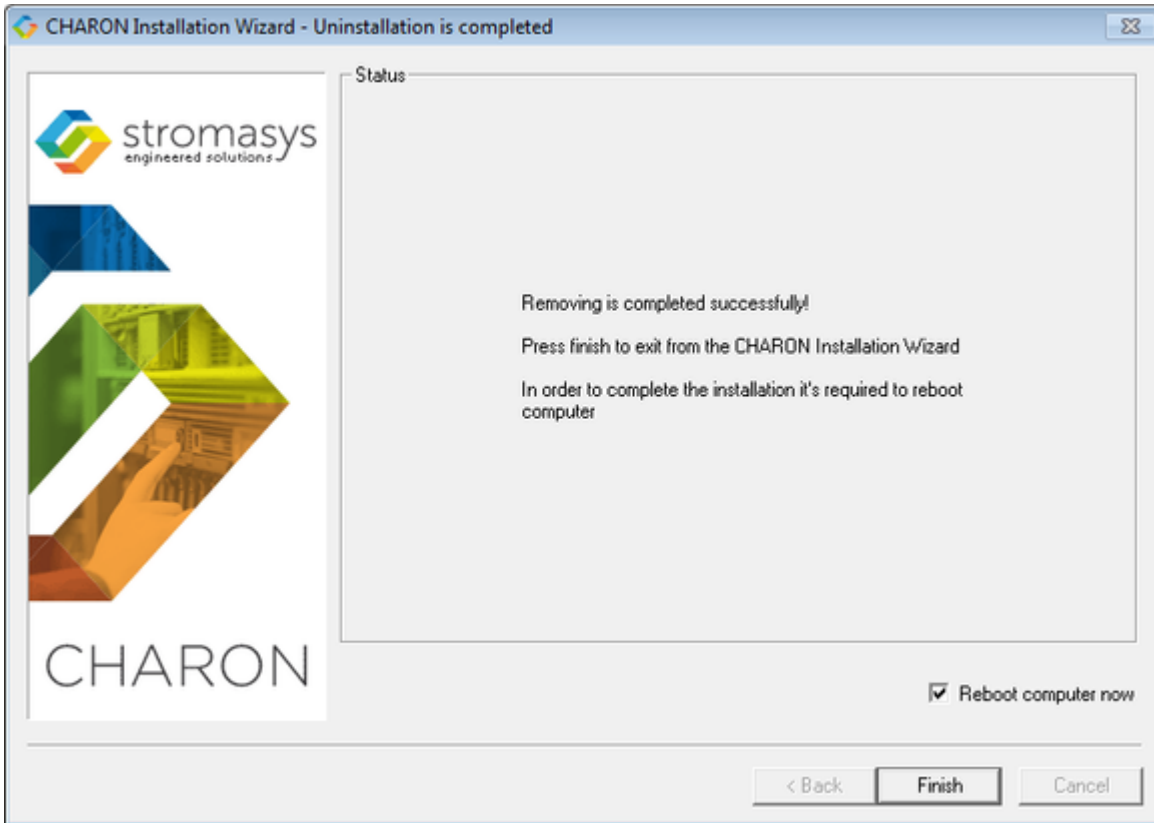


Do not close the uninstallation dialog but exit from "CHARON Virtual Machine Manager" the way described above.

Press the "OK" button then the "Retry" one in the appearing dialog:



Once uninstallation (or changing) is done, the following dialog will be displayed:



Keep the "Reboot computer now" selected and press the "Finish" button to reboot the CHARON host.



# Appendixes

## Contents

- Log monitoring - logmond best practice

# Log monitoring - logmond best practice

## Table of contents

- [Description](#)
- [Recommendations](#)
- [Notes](#)
- [Service setup example](#)
  - [Setup and start your virtual machine.](#)
  - [Install and start the log monitor service](#)
- [Running the log monitor from the command line](#)
  - [Starting in background mode](#)
  - [Debugging](#)
- [nolicense.bat considerations](#)
- [Alerts customization examples](#)
  - [Creating a Windows Event](#)
  - [Sending an email via powershell](#)
    - [Example using an office365 account](#)
    - [Example using a gmail account](#)
- [Related articles](#)


## Description


The Log Monitor & Dispatcher, known as LOGMOND, is a special program which monitors a guest LOG file produced by Charon and executes a customized 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 00000424 ("Detected removal of a license") or 0000002A or 00000351 for backward compatibility, it submits the `nolicense.bat` procedure (this BAT file must be created manually). It also invokes the `license_changed.bat` script, if it exists, when it detects a message with code 0000040B / "License has changed. License detected and online".

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.

The tool requires a specific file "`nolicense.bat`", and optionally a file named "`license_changed.bat`", containing some specific instructions to be taken in situation of license absence or license change. These files must be created in the virtual machine's home directory.

 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.

 **A new utility is available for download that offers more monitoring capabilities than logmond.**

Please see [CHARON on Windows - Charon Log Monitor](#) and contact us to get the download link if needed.

## Recommendations

- Create a specific folder to hold the Charon configuration and log files, the `nolicense.bat` file, the `license_changed.bat` if needed, and any specific action script you would add.
- Use the rotating log files mechanism or use a single log file with "append" mode.
- Install and start the Charon log monitor service, named "`EmulatorLogMonitor`": doing so you will not have to start `logmond` manually in interactive mode and newly added virtual machines services will be automatically taken into account. Use interactive mode for tests and debugging purpose.

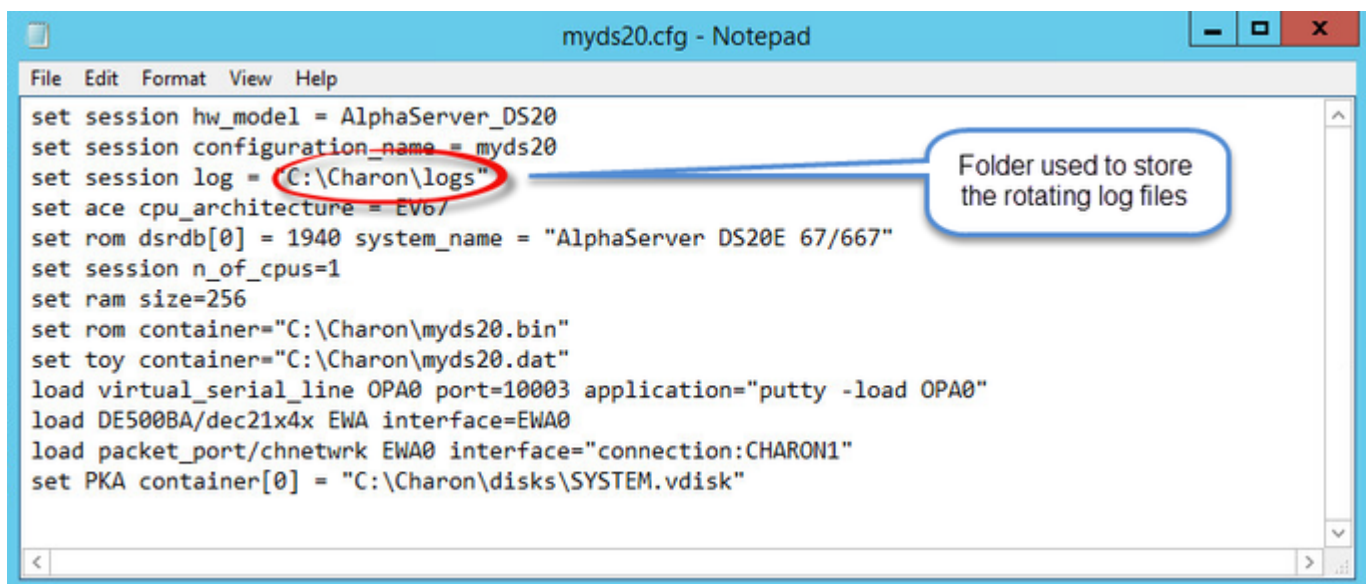
## Notes

- The log monitor utility will detect the absence or change of the dongle only at a specified interval, called the license check interval and defined by the license. Its default is set to 1 hour. 💡 In case you need to perform tests, you can send us a fresh C2V file and ask for an update to reduce this interval.
- Once started the logmond utility will lock the virtual machine log file, you will not be able to move it, copy it or delete it. You will however be able to open it using the notepad utility or other log editors like `baretail` for example which will allow you to have a continuous view of the log with highlights (see: [Charon Log files - Using baretail on Windows for syntax highlighting](#)) or `vim` (see: [Charon Log files - Using vim for syntax highlighting on Linux and Windows](#)).

## Service setup example

Example below is based on a Charon-AXP V4.9 B19402 running on a Windows 2012 R2 server and emulating an AlphaServer DS20. All files (configuration, bin, rom and `nolicense.bat`) are located in `C:\Charon`, rotating log files are located in `C:\Charon\logs`

The configuration file is defined as follows:



```

set session hw_model = AlphaServer_DS20
set session configuration_name = myds20
set session log = "C:\Charon\logs"
set ace cpu_architecture = EV6/
set rom dsrdb[0] = 1940 system_name = "AlphaServer DS20E 67/667"
set session n_of_cpus=1
set ram size=256
set rom container="C:\Charon\myds20.bin"
set toy container="C:\Charon\myds20.dat"
load virtual_serial_line OPA0 port=10003 application="putty -load OPA0"
load DE500BA/dec21x4x EWA interface=EWA0
load packet_port/chnetwrk EWA0 interface="connection:CHARON1"
set PKA container[0] = "C:\Charon\disks\SYSTEM.vdisk"

```

## Setup and start your virtual machine.

- Start the "Charon Virtual Machines Manager"
- Create your virtual machine and start it

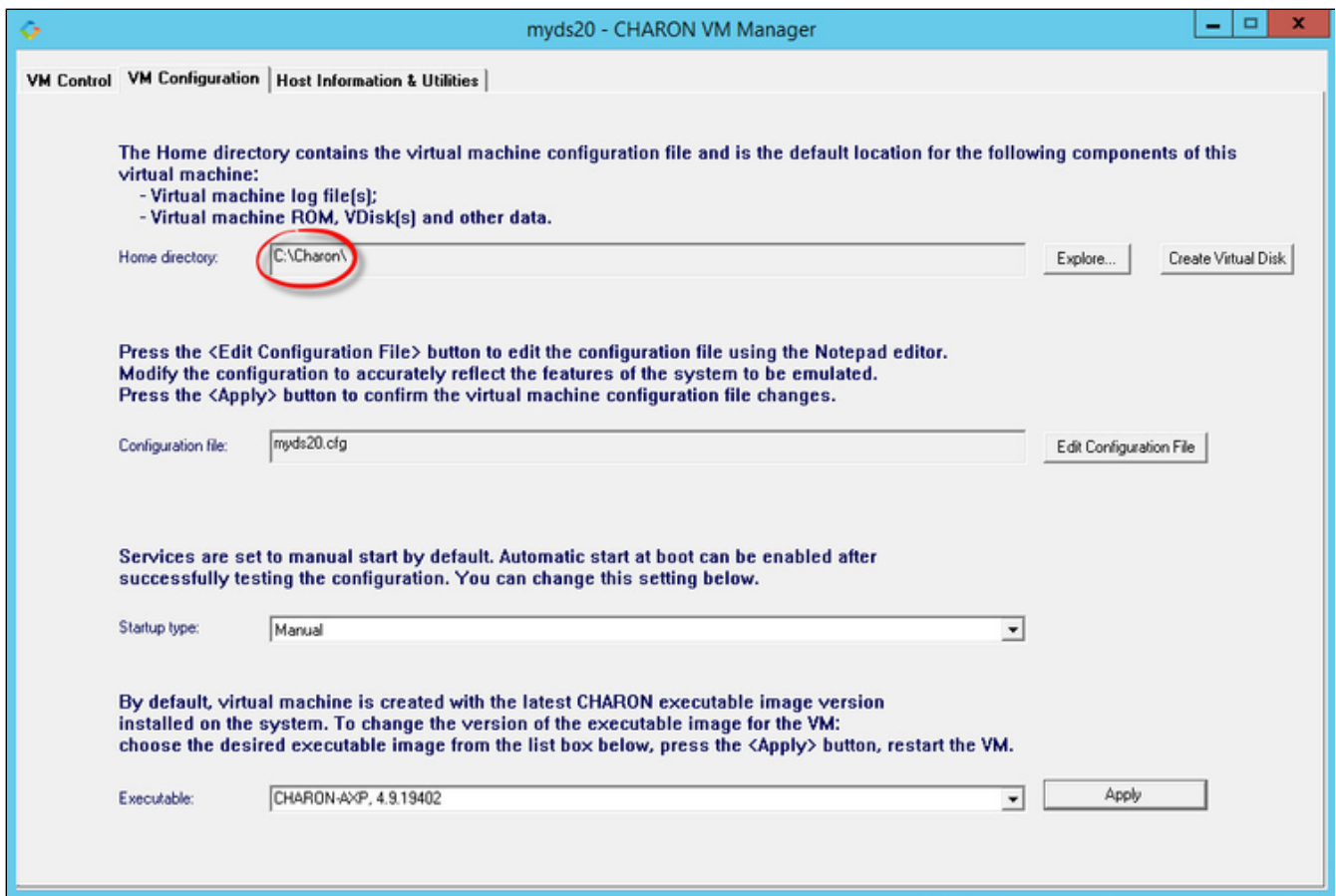
📘 See [Running and managing Charon-AXP for Windows](#) or [Running and managing Charon-VAX and Charon-PDP for Windows](#) for details.

## Install and start the log monitor service

- First, prepare a `nolicense.bat` file. See [nolicense.bat considerations](#) chapter further for more information if needed.

⚠️ The `nolicense.bat` file has to be placed in the virtual machine "Home directory". To locate this "Home directory" open the "Charon Virtual Machines manager" utility, select the virtual machine and switch to the "VM Configuration" tab

**Example:**



Use the notepad for example to create the `nolicense.bat` script. We will here add a line in a log file:

```
echo %date% %time% dongle removed >>C:\Charon\logmond.log
```

The `license_changed.bat` script can also contain quite the same line:

```
echo %date% %time% license has changed >>C:\Charon\logmond.log
```

- Install the log monitoring service and start it:

**⚠ These command lines must be executed as an administrator in order to manage the services**

```
Microsoft Windows [Version 6.3.9600]
(c) 2013 Microsoft Corporation. All rights reserved.

C:\Users\Administrator>"C:\Program Files\CHARON\Build_19402\x64\logmond.exe" -r
C:\Users\Administrator>"C:\Program Files\CHARON\Build_19402\x64\logmond.exe" -i
C:\Users\Administrator>"C:\Program Files\CHARON\Build_19402\x64\logmond.exe" -u
C:\Users\Administrator>powershell -command "get-process -includeusername logm* ! format-table -autosize"
Handles WS(K) UM(M) CPU(s) Id UserName ProcessName
-----
62 3100 44 0.02 3956 NT AUTHORITY\SYSTEM logmond
74 3216 47 0.00 4068 NT AUTHORITY\SYSTEM logmond

C:\Users\Administrator>_
```

- Notes:

- The `"logmond -r"` command removes the service if any
- The `"logmond -i"` command installs the service

- The "logmond -u" command starts the service
  - The powershell commands shows 2 logmond processes, one scans for new services, the other is monitoring the log file of the current service (myds20)
  - The EmulatorLogMonitor service running the logmond processes will discover already installed Charon virtual machines services. If a new virtual machine service is added, it's log file will be automatically discovered and monitored.
  - The logmond.exe file is located by default in the "C:\Program Files\Charon\Build\_19402\x64" folder for Charon-AXP/VAX/PDP V4.9, "C:\Program Files\Charon\Build\_20200\x64" for V4.10.
- In this test, two dongles were connected, the active one has been removed (1) producing the license changed message (2 and 3) then the last active dongle has been removed (4 and 5).
- ❗ During the tests the license check interval was set to 5 minutes therefore, when the dongle is disconnected, a message is sent telling there are only 4 minutes remaining before the DS20 stops

The screenshot displays the CHARON VM Manager interface for the 'myds20' virtual machine. The 'Host Information & Utilities' tab is active, showing a log window titled 'logmond.log (C:\Charon) - GVIM'. The log window contains the following messages:

Date	Message ID	Message text
2018-09-10 16:50:02	00000402	Device arrival with interface = "\\?\USB#VID_0529&PID_0001#7&25c389c1&0&1#{a5dcbf10-6530-1...
2018-09-10 16:50:02	00000402	Device arrival with interface = "\\?\USB#VID_0529&PID_0001#7&25c389c1&0&1#{29db6d11-bc3c-1...
2018-09-10 16:50:02	00000402	Device arrival with interface = "\\?\USB#VID_0529&PID_0001#7&25c389c1&0&1#{bce9d0f1-f7db-11...
2018-09-10 16:50:03	00000402	Device arrival with interface = "\\?\Aladdin#Vid0529&Pid0001&HaspHL#8&e9a2b73&d&00#{44207c5...
2018-09-10 16:50:03	00000402	Device arrival with interface = "\\?\USB#Hasp#8&e9a2b73&d&00#{7ab02247-3798-44cd-8472-2435...
2018-09-10 16:50:29	0000024D	Communication with the license key "804839763" lost. <b>1</b>
2018-09-10 16:53:29	0000024D	Performing regular check of the current license.
2018-09-10 16:53:29	0000024D	Unable to log in to the key "804839763", feature 0.
2018-09-10 16:53:29	0000024D	HASP runtime [50]. Unable to locate any Feature matching scope.
2018-09-10 16:53:29	0000024D	Failed to login at the Sentinel HASP key: "804839763".
2018-09-10 16:53:29	0000024D	Checking the available license key "1022239689".
2018-09-10 16:53:29	0000024D	Found license key: "1022239689".
2018-09-10 16:53:29	0000024D	Checking product section 0.
2018-09-10 16:53:29	0000024D	License number: "1001082".
2018-09-10 16:53:29	0000024D	CHARON product code: "CHAXP-470xx-WI-LI".
2018-09-10 16:53:29	0000024D	HASP Date limited license, limited to: 10-Jan-2019 00:55:00.
2018-09-10 16:53:29	0000024D	Feature 1 check interval 5 minute(s).
2018-09-10 16:53:29	0000024D	Current license changed from key:"804839763" id:"100439763" key:"1022239689" id:"1001082".
2018-09-10 16:53:29	0000024D	License has changed. License detected and online. <b>2</b>
2018-09-10 16:58:24	00000403	Device removal with interface = "\\?\Aladdin#Vid0529&Pid0001&HaspHL#8&e9a2b73&d&00#{44207c5...
2018-09-10 16:58:24	00000403	Device removal with interface = "\\?\USB#Hasp#8&e9a2b73&d&00#{7ab02247-3798-44cd-8472-243...
2018-09-10 16:58:24	00000403	Device removal with interface = "\\?\USB#VID_0529&PID_0001#7&25c389c1&0&1#{bce9d0f1-f7db-11...
2018-09-10 16:58:24	00000403	Device removal with interface = "\\?\USB#VID_0529&PID_0001#7&25c389c1&0&1#{a5dcbf10-6530-1...
2018-09-10 16:58:24	00000403	Device removal with interface = "\\?\USB#VID_0529&PID_0001#7&25c389c1&0&1#{29db6d11-bc3c-1...
2018-09-10 16:58:36	0000024D	Communication with the license key "1022239689" lost.
2018-09-10 16:58:36	0000024D	Performing regular check of the current license.
2018-09-10 16:58:36	0000024D	Unable to log in to the key "1022239689", feature 0.
2018-09-10 16:58:36	0000024D	HASP runtime [7]. Sentinel protection key not available.
2018-09-10 16:58:36	0000024D	Failed to login at the Sentinel HASP key: "1022239689".
2018-09-10 16:58:36	0000024D	Connection to license key:"1022239689" id:"1001082" lost. Application will stop at 11-Sep-2018 04:58...
2018-09-10 16:58:36	00000424	session: Detected removal of the license. Normal operation is supported till 11-Sep-2018 04:58:36.

The log window also shows a summary of license changes:

```

on 09/10/2018 16:53:40.52 license changed
on 09/10/2018 16:58:40.54 dongle removed

```

The interface includes buttons for VM management: Create VM from Template, Add Existing VM, Update VM List, Start, Stop, Remove VM, Edit Configuration, Update Log, Display Log, and Show Console.

## Running the log monitor from the command line

- ❗ The "logmond" utility has several parameters available. Use the "logmond -h" command to have a complete list.

## 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 Charon configuration file.
3. Start the Log Monitor & Dispatcher using the following command line as an example:

For single log file:

```
...> "C:\Program Files\Charon\Build_19402\x64\logmond" C:\my_Charon.log
```

For rotating log file:

```
...> "C:\Program Files\Charon\Build_19402\x64\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\Build_19402\x64\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.

## Debugging

To diagnose LOGMOND using trace feature:

1. Stop LOGMOND service using "--stop" (or "-d" or "--down"). This is not absolutely necessary though, as multiple running LOGMOND utilities do not affect each other.
2. Start "almost-as-a-service" LOGMOND with diagnostic trace using two command line switches: "-e" and "-t". This starts master task of LOGMOND which enumerates all Charon services and for each of them invokes slave task of LOGMOND. The master task of LOGMOND looks for installing or removal Charon services.
  - a. When Charon service is installed, master creates new slave.
  - b. When Charon service is removed, slave is terminated to. The slave task monitors LOG of particular Charon service.

 The "-t" switch enables trace windows, so you can see the process of monitoring.
3. To terminate "almost-as-a-service" LOGMOND master and all its slaves, bring focus to master's trace window, and press "Ctrl-C" ([X] button closes master task, but slaves continue to run unmastered).

The LOGMOND still supports backward compatibility mode, when it does NOT monitor ALL Charon services, but only monitors a particular folder or even a particular file. So, if you want to diagnose how it monitors a file, you can use the same "-t" option:

```
C:\...>logmond.exe -t -l <path to Charon log file>
```

If you want to diagnose monitoring rotating log:


```
C:\...>logmond.exe -t -l <path to Charon log directory> -p <log prefix>
```

If you want to diagnose monitoring particular service:

```
C:\...>logmond.exe -t -s <service name>
```



When you install LOGMOND to run as a service it implicitly runs with "-e" option but without "-t". Alternatively you may install LOGMOND as a service with explicitly specified name of log file (for backward compatibility)

 You better use "-e" together with "-t" to diagnose monitoring Charon services. Unless you run instances with Launcher, in which case you will have to invoke LOGMOND with "-l" with "-t" options.


## nolicense.bat considerations

LOGMOND always looks for `nolicense.bat` and `license_changed.bat` files in the current working directory. What is current working directory for LOGMOND depends on how it is invoked.

If you run LOGMOND with just path to log file, the LOGMOND inherits current working directory from command prompt it is invoked from. So, before you run LOGMOND this way, change prompt's working directory to where you keep your NOLICENSE.BAT file.


If you run LOGMOND with name of Charon service, the LOGMOND changes its current working directory to the Charon service's Home directory. So, keep the `nolicense.bat` and `license_changed.bat` files in the virtual machine's home directory.

If you run LOGMOND with "-e" option, each slave receives a particular name of Charon service, and therefore changes to the service's Home directory. So, keep the `nolicense.bat` and `license_changed.bat` files in the virtual machine's home directory.

 Stromasys highly recommends to use logmond as a service, installing it with the "-i" option then starting it with "-u".


## Alerts customization examples

You will find below some examples of alerts that can be sent from the `nolicense.bat` file

 **Reminder:** Interactive applications are not allowed

### Creating a Windows Event

```
powershell -command "New-Eventlog -Logname application -Source Charon -Erroraction silentlycontinue;Write-Eventlog -Logname application -Source Charon -Entrytype Error -Eventid 314 -Message 'Charon license dongle disconnected'"
```

 It is recommended to use the powershell command to write events instead of the "eventcreate" command line as some other events are created with powershell scripts. Using the eventcreate command in this case will fail as the source, Charon, will not be accepted.

**Example:**

Level	Date and Time	Source	Event ID	Task Cate...
Error	10/20/2016 11:30:07 AM	CHARON	314	(1)
Error	10/20/2016 11:30:00 AM	CHARON	300	(1)
Error	10/20/2016 11:26:26 AM	CHARON	314	(1)
Error	10/20/2016 11:25:10 AM	CHARON	314	(1)
Error	10/20/2016 11:15:49 AM	CHARON...	314	None
Information	10/20/2016 11:15:00 AM	CHARON	300	(1)
Error	10/20/2016 11:10:00 AM	CHARON	300	(1)

Event 314, CHARON

General Details

CHARON license dongle disconnected

Log Name: Application

Source: CHARON	Logged: 10/20/2016 11:30:07 AM
Event ID: 314	Task Category: (1)
Level: Error	Keywords: Classic
User: N/A	Computer: bruno-PC
OpCode:	
More Information: <a href="#">Event Log Online Help</a>	

**i** The "Event Id" can be set at your convenience. A valid ID is any number from 1 to 65535.

## Sending an email via powershell

An alert email can be sent using the "Send-MailMessage" powershell command however it is important to know no interactive command can be executed (for asking the sender's credentials for example).

You can either ask the `nolicense.bat` file to invoke a powershell script or pass the commands from the command line

**!** Examples below are given with Powershell V4.0 installed on a Windows 2012 R2 server, some commands may not be appropriate to your Windows distribution.

**💡** To determine which version of Powershell is installed and upgrade if necessary, see [Powershell version, upgrade, enabling scripts execution, tips and tricks](#)

**i** To run PowerShell scripts (files that end with `.ps1`), you must first set the execution policy to Unrestricted (This operation has to be done once). To do so, open a command line window (`cmd.exe`) as an Administrator and use the following command:

```
c:\Charon>powershell -command "Set-ExecutionPolicy Unrestricted"
```

**i** The ExecutionPolicy can also be set to "RemoteSigned". In this case the `.ps1` script files will have to be unblocked as described below. If you are still prompted to allow for execution of the script, please run the following command to unblock the `.ps1` file you want to execute:

```
c:\Charon>powershell -command "Unblock-File -path c:\Charon\sendmail.ps1"
```





## Example using an office365 account

Create a powershell script file, named `sendmail.ps1` for example and located in "C:\Charon" folder:

```
# Update the email addresses below:
$From = "<monitoring-account>@<somewhere>"
$To = "<someone>@<somewhere>"

# Send the email
send-mailmessage -to $To -from $From -subject "Charon detected removal of the license" -body "Please check" -smtpserver smtp.office365.com -usessl -port 587 -delivery none
```

  If you must specify multiple recipients, you can specify them using an array with comma separated fields.

### Example1:

```
$To = @( "Kirk <Kirk@uss-enterprise.fed>", "Spock <spock@uss-enterprise.fed>" )
```

### Example2:

```
$To = @( "Kirk@uss-enterprise.fed", "Spock@uss-enterprise.fed", "McCoy@uss-enterprise.fed" )
```


Update the `nolicense.bat` file as follows:

```
powershell -NonInteractive -File C:\Charon\sendmail.ps1
```

The problem here is the email account mentioned in the `$Myemail` variable must be able to send anonymous emails otherwise the following error can occur:

```
"Client was not authenticated to send anonymous mail during MAIL FROM"
```

If the account cannot send anonymous emails, you can perform an interactive test by asking for the credentials:

 The "`<monitoring-account>@<somewhere>`" and "`<recipient>@<somewhere>`" values must be adapted to your configuration

```
# Update the email addresses below:
$From = "<monitoring-account>@<somewhere>"
$To = "<someone>@<somewhere>"

# The command below will open a window for you to store the password
$Creds=(get-credential -credential "$From")

# Send the email (credentials are then required each time)
send-mailmessage -to $To -from $From -subject "Charon detected removal of the license" -body "Please check" -smtpserver smtp.office365.com -usessl -port 587 -delivery none -credential $Creds
```

This method cannot be used when executed within a service due to the popup window that will be opened to ask for the email account password.

To be able to send an email, we can then store the credentials in an encrypted and protected file. This is done using the `Export-CliXML` powershell command. As this file will be bound to the server where the command is issued and to the user account who created the file, we need to perform the operations as "system" user account because this user is the one running the `logmond` processes that will invoke the `nolicense.bat` file.

It is necessary then to use `psexec` from the Windows Sysinternals kit. It can be downloaded here: <https://technet.microsoft.com/en-us/sysinternals/psexec.aspx>

Once extracted from the zip file, to the "C:\Charon" folder for example, run the following command from the command line running as Administrator:


```
C:\...\> C:\Charon\psexec -i -s cmd.exe
```


and generate the CliXML file as described below:

```
C:\Windows\system32>powershell
Windows PowerShell
Copyright (C) 2014 Microsoft Corporation. All rights reserved.


PS C:\Windows\system32> $From="<monitoring-account>@<somewhere>"
PS C:\Windows\system32> $Creds=(get-credential -credential $From)
PS C:\Windows\system32> $Creds | Export-CliXML C:\Charon\creds.clixml
PS C:\Windows\system32> exit

C:\Windows\system32>exit
```

 If it is no more needed, remove the "psexec.exe" file.

 If the sender's email address or its password has to be changed, the .clixml file will have to be recreated.


Now update the powershell script as follows:

 The "<monitoring-account>@<somewhere>" and "<recipient>@<somewhere>" values must be adapted to your configuration

```
# Update the email addresses below:
$From = "<monitoring-account>@<somewhere>"
$To = "<someone>@<somewhere>"


$Creds = Import-CliXml C:\Charon\creds.clixml

send-mailmessage -to $To -from $From -subject "Charon detected removal of the license" -body "Please check" -credential
$Creds -smtpserver smtp.office365.com -usesssl -port 587 -delivery none
```

 Once updated, the script will only work when executed by the "system" account and on the server where the Export-CliXML powershell command was issued.

## Example using a gmail account


To send an email using a gmail account, perform the same operations as described in the [Example using an office365 account](#) chapter above and replace the `smtpserver` value by `smtp.gmail.com` as shown below:


 The "`<monitoring-account>@gmail.com`" and "`<recipient>@<somewhere>`" values must be adapted to your configuration

```
# Update the email addresses below:
$From = "<monitoring-account>@gmail.com"
$To = "<someone>@<somewhere>"

$Creds = Import-CliXml C:\Charon\creds.clixml

send-mailmessage -to $To -from $From -subject "Charon detected removal of the license" -body "Please check" -credential
$Creds -smtpserver smtp.gmail.com -uessl -port 587 -delivery none
```

 Only the `smtpserver` value must be changed

 Google may block sign-in attempts when using powershell and `send-mailmessage`. In this case the sender will receive a "Sign in attempt prevented" alert email. To allow emails to be sent:

1. Create a dedicated gmail account
2. Allow less secure apps to access your account. See this article: <https://support.google.com/accounts/answer/6010255?hl=en>

## Related articles

-  [CHARON on Windows - Automated License Expiration Check](#)
-  [CHARON on Windows - Charon Log Monitor](#)
-  [CHARON on Windows - Charon Log Monitor - Release Notes](#)
-  [CHARON on Windows - Rotating log files cleaning](#)
-  [CHARON on Windows - Rotating log files cleaning - Release Notes](#)