stromasys
legacy server emulation

CHARON-VAX and CHARON-PDP V4.9 for Windows Users Guide

Contents

Introduction

Table of Contents

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.
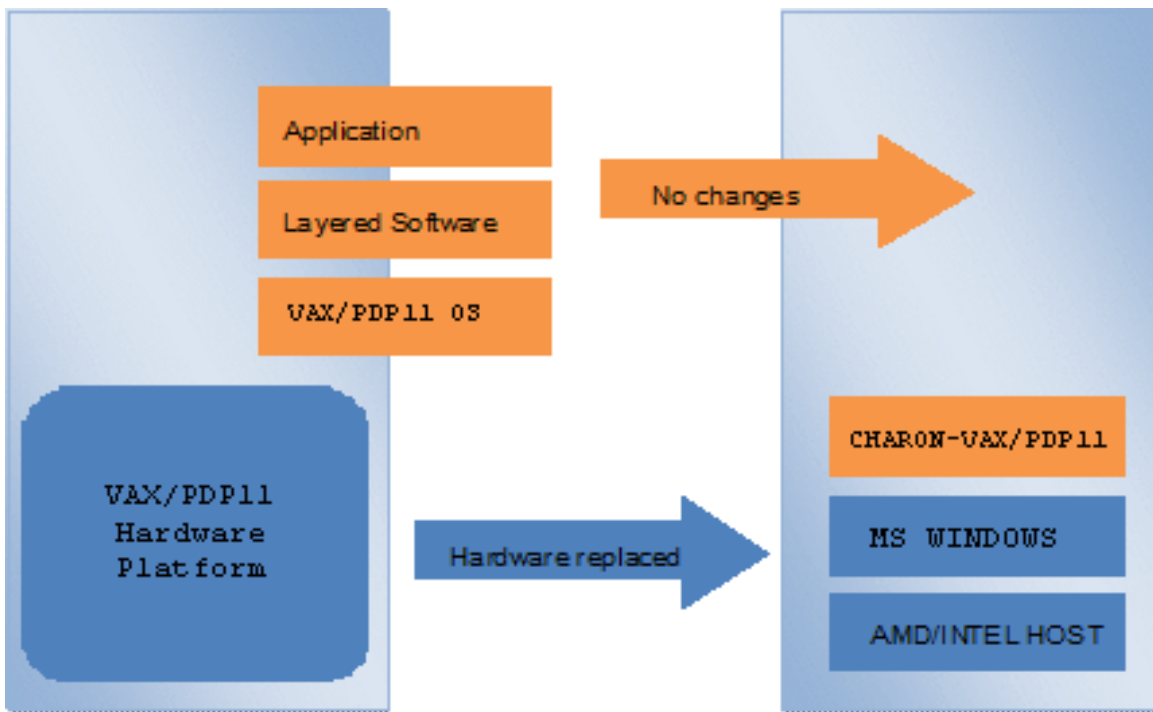
This approach is best understood when the VAX and PDP11 Hardware Virtualization Software is viewed as a special interface between the old VAX and PDP11 software and a new hardware platform. Basically, the CHARON software presents a VAX and PDP11 hardware interface to the original VAX and PDP11 software, 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 product is 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.
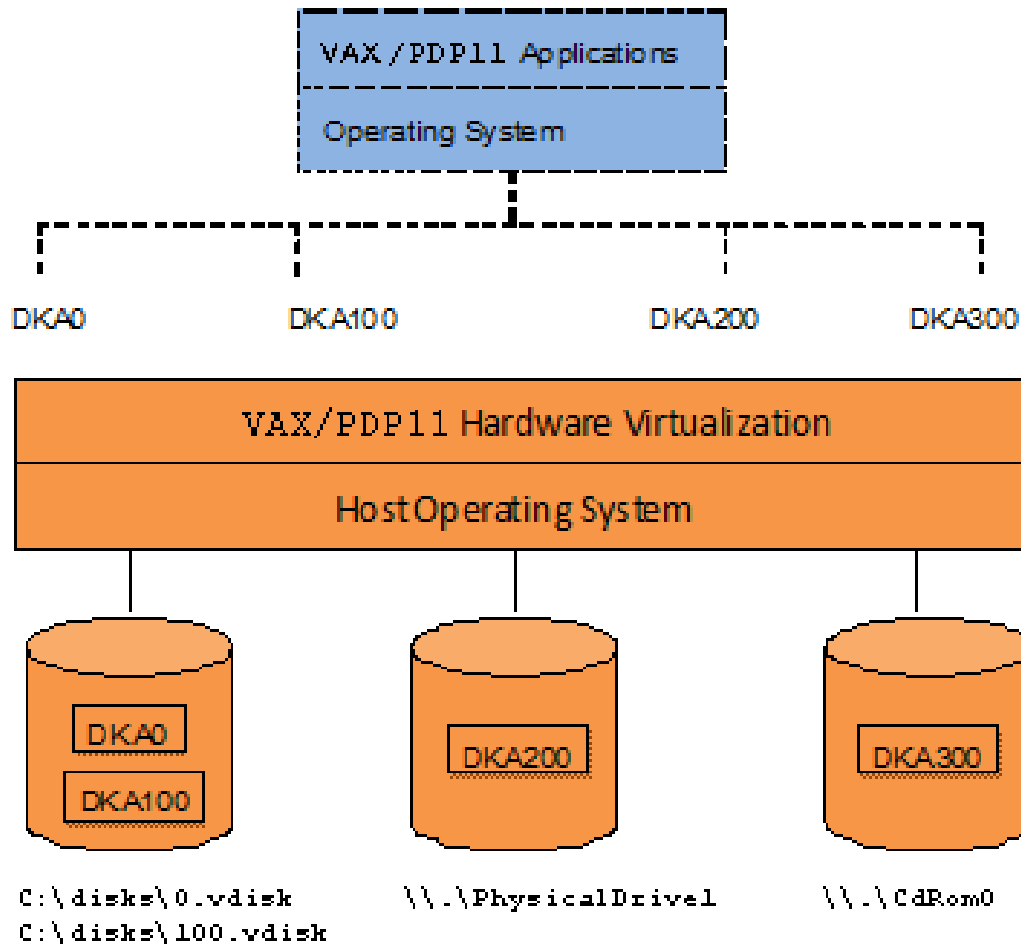
With the use of current storage technology, disks do not have to be physically attached to the Host platform, they can also reside on a SAN or iSCSI storage structure.

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

Conventions

Throughout the document(s) these conventions are followed

Notation	Description
$	The dollar sign in interactive examples indicates an operating system prompt for VMS. The dollar sign can also indicate non superuser prompt for UNIX / Linux.
#	The number sign represents the superuser prompt for UNIX / Linux.
>	The right angle bracket in interactive examples indicates an operating system prompt for Windows command (cmd.exe).
User input	Bold monospace type in interactive examples indicates typed user input.
<path>	Bold monospace type enclosed by angle brackets indicates command parameters and parameter values.
Output	Monospace type in interactive examples, indicates command response output.
[]	In syntax definitions, brackets indicate items that are optional.
...	In syntax definitions, a horizontal ellipsis indicates that the preceding item can be repeated one or more times.
dsk0	Italic monospace type, in interactive examples, indicates typed context dependent user input.

The following definitions apply

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

CHARON-VAX and CHARON-PDP for Windows installation

Table of contents

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.

Hardware Requirements

Number of CPU cores

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

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	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 3 GHz.

Operative 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.

> 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

CHARON-VAX / CHARON-PDP networking assumes dedicated host Ethernet adapters; their number must be equal to the emulated adapters to be configured in CHARON-VAX / CHARON-PDP11. One adapter (optionally) can be left to the host for TCP/IP networking etc.

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

Starting with ESXi 6.5, it is not possible to select E1000 adapter when a new virtual machine is created using Windows Server 2012 and 2016 templates.

The following workarounds are avaible:

- import the virtual machine from an older version of ESXi
- do not select Windows Server 2012 or 2016 during virtual machine creation but "Windows 10/64bit" or "Other/Other 64bit" for example.

Software Requirements

- Microsoft Windows Server 2016 Standard and Datacenter Editions, 64 bit version
- Microsoft Windows Server 2012 R2 Standard and Datacenter Editions, 64 bit version (please notice that only R2 revision is supported)
- Windows 10 Pro and Enterprise Editions, 32 bit and 64 bit version
- 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 bit and 64 bit versions
- VMware ESXi 5.5 and 6.x up to 6.7 (requires a supported Windows operating system on top of a ESXi virtual machine)
- Microsoft Hyper-V (requires a supported Windows operating system on top of a Hyper-V virtual machine)

Host system preparation

All antivirus, screen saver, automatic (scheduled) backup and any other CPU consuming software as well as software that is able to temporarily lock CHARON files (such as automatic indexing) must be turned off, uninstalled and disabled. 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 maintainence periods established by the system administrator. Before applying new updates one must shutdown the operating system running on CHARON and stop all the running CHARON instances and services.

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

- *On server side (where network license will reside)*: open port 1947 for both TCP and UDP
- *On 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 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 found applet.
2. Find the "AlwaysDisableMCN" parameter in the following way:

```
HKEY_LOCAL_MACHINE
 + SYSTEM
    + CurrentControlSet
       + Enum
          + SCSI
             + Disk&Ven_DEC&Prod_RWZ53_____(C)DEC  <- This one depends on Vendor and Model of your MO drive
                + 5&fd233cf&0&000500 <- This one depends on hardware connection (SCSI ID of MO drive, location of SCSI HBA
on the host's mainboard, etc ...)
                   + Device Parameters
                      + MediaChangeNotification
                         + AlwaysDisableMCN = DWORD:00000000
```

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

4. Reboot the host system.

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 your Windows User's Guide on details.

> 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.

Upgrade cases:

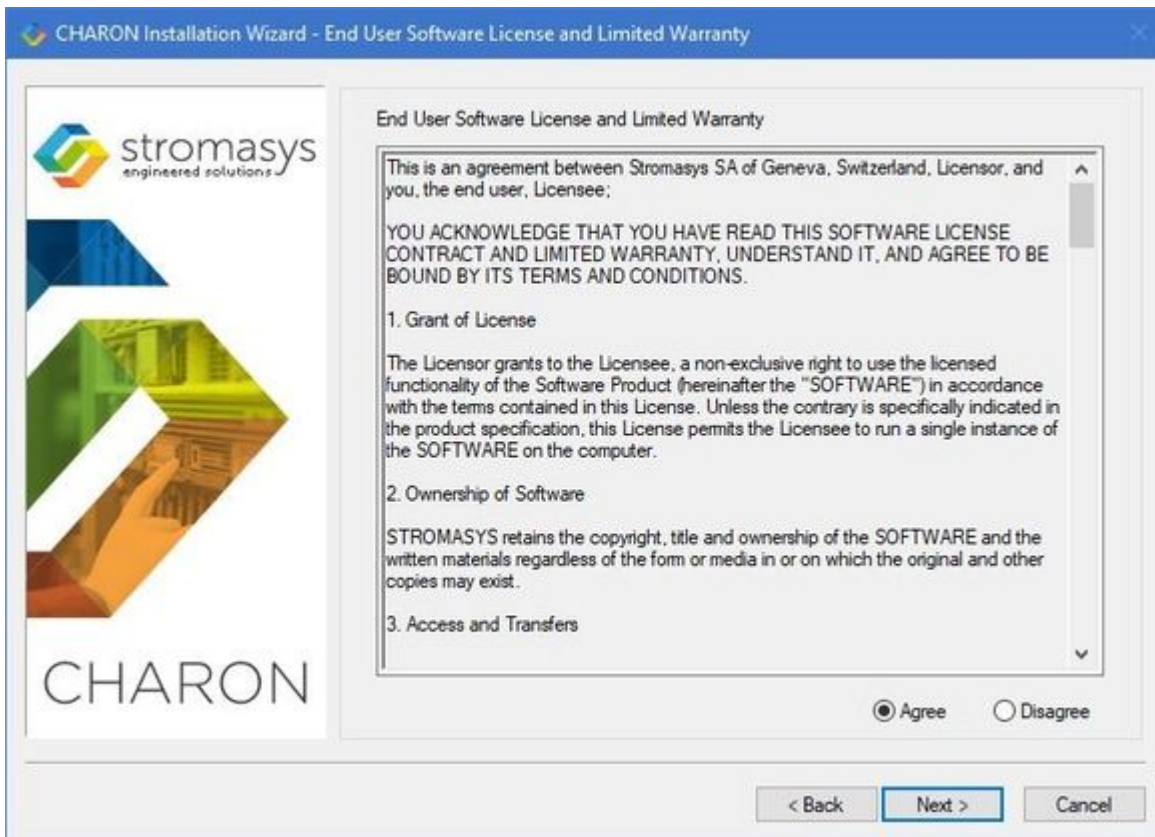| From version	Action
4.7 and below	First uninstall all CHARON products.
4.8	The version 4.9 can be installed directly. ⚠ Please note it is mandatory to update all existing CHARON Virtual Machines to make them run the latest version using the Virtual Machines Manager, VM Configuration tab - otherwise they will remain at V4.8.

Distribution preparation

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

📄 HASP.def	15.05.2018 14:54
📄 InstallShell.exe	15.05.2018 14:54
📄 InstallShellCHS.dll	15.05.2018 14:54

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 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.

> The CHARON products may be highlighted with a red color. 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.
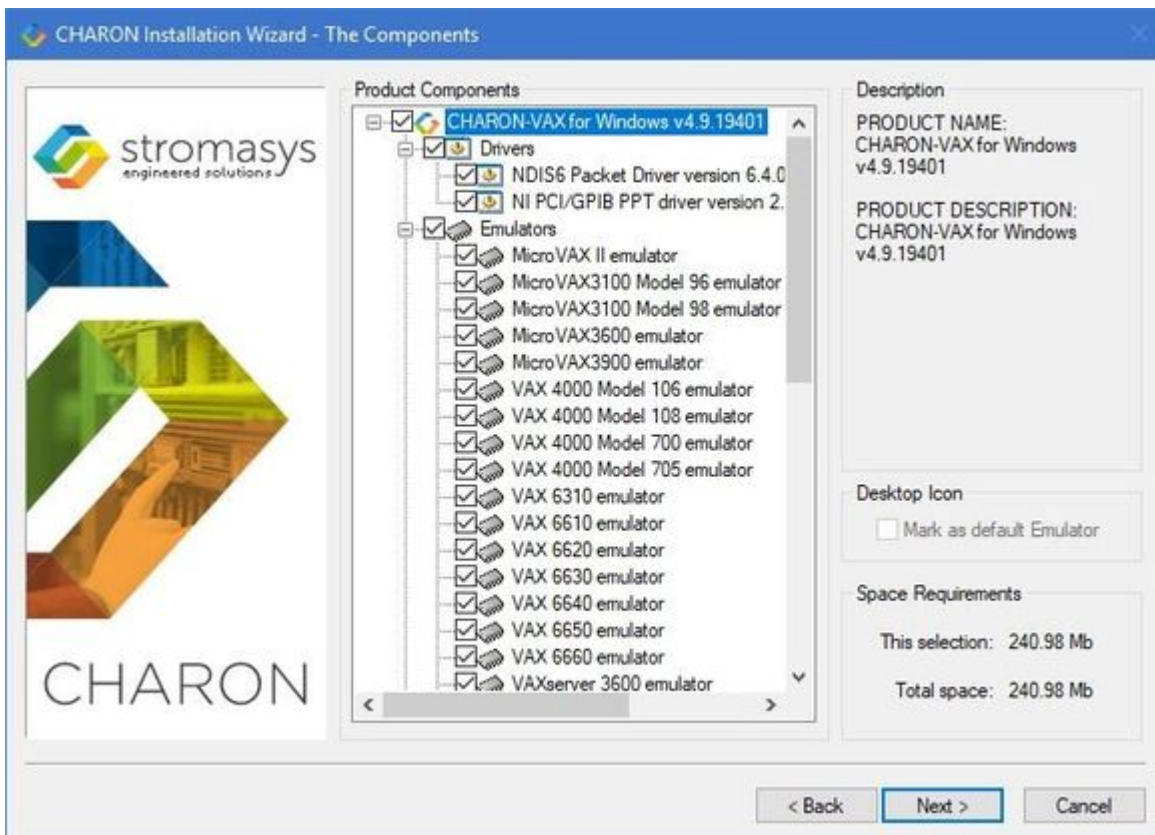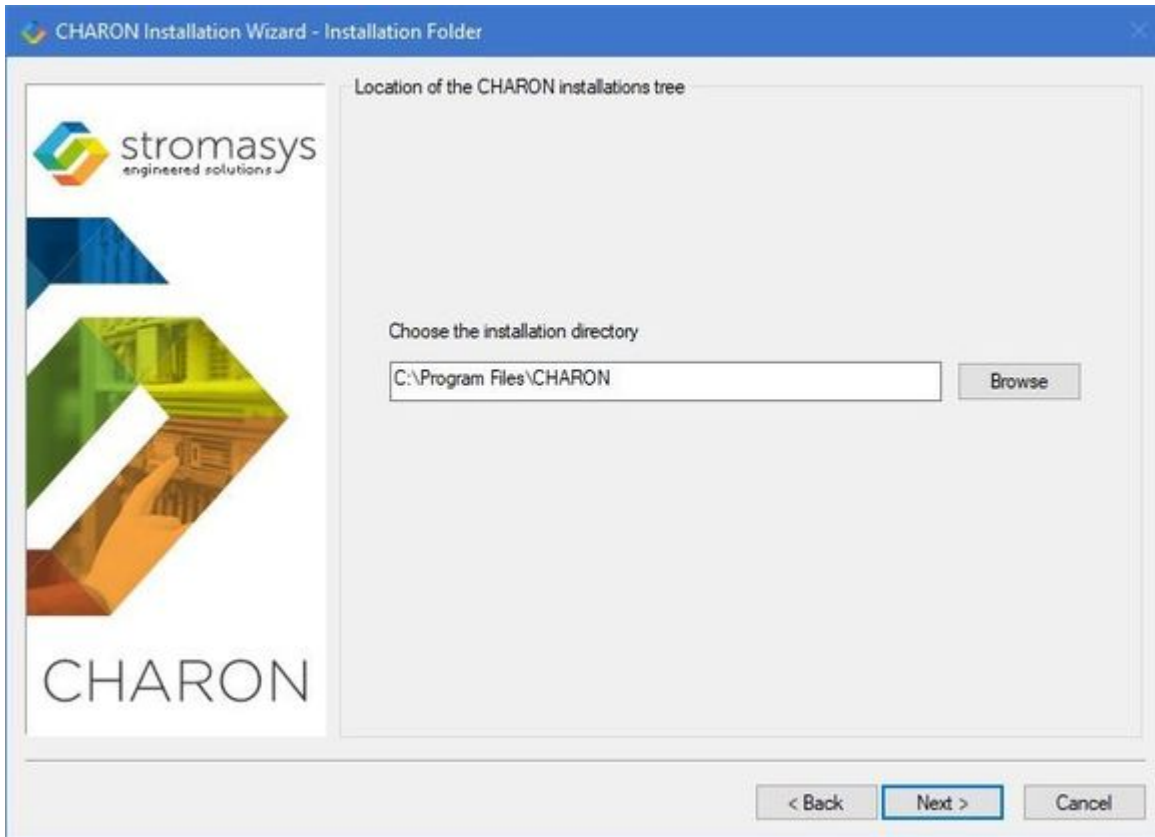
Press the "Next" button.

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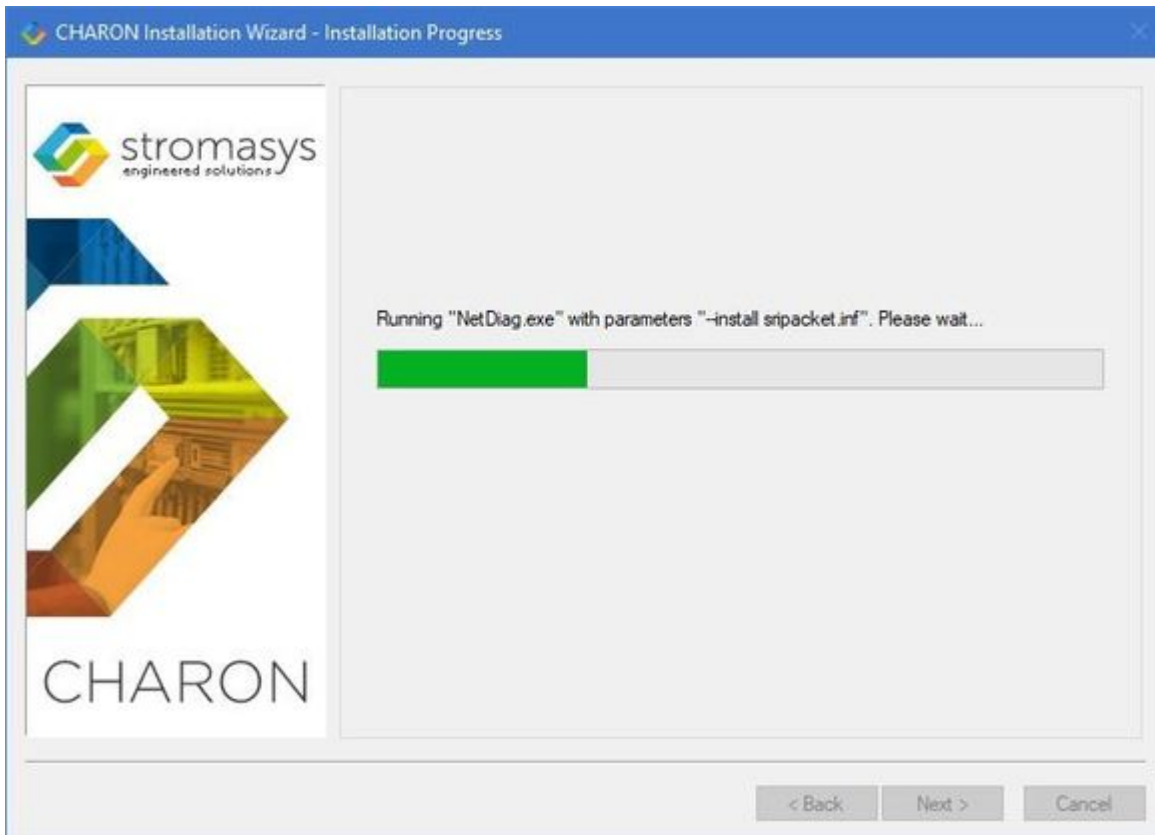
During the CHARON installation procedure, you will have to confirm the installation of the CHARON-specific network driver. Press the "Install" button:

Once the installation is done, the following dialog will be displayed:
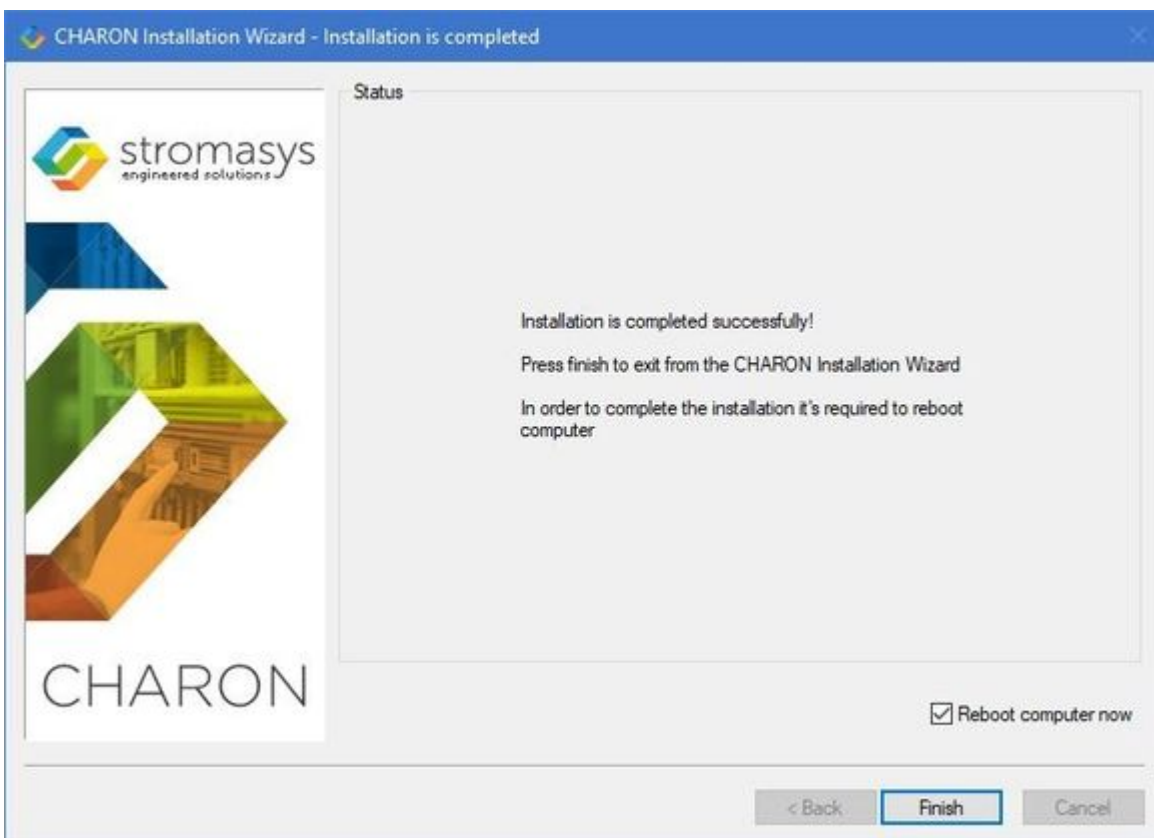
⚠ It is strongly recommended to reboot the CHARON host system immediately: leave the "Reboot computer now" box checked and press the "Finish" button.

Once the host system is up again, you will notice a new shortcut, "CHARON Virtual Machines Manager", created on the desktop during the installation procedure:

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

> In case of upgrade from V4.8, update all existing CHARON Virtual Machines to make them run the latest version - otherwise they will remain at V4.8. See the following chapters for details.

The CHARON installation procedure has created the following structure under the "Start" menu:
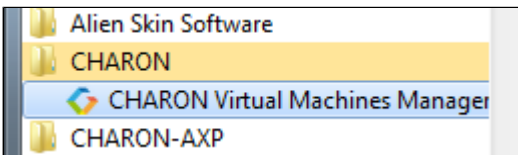
CHARON-VAX / CHARON-PDP home directory

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
InstallShell	Contains the CHARON installation procedure executables
Logs	Contains the CHARON installation log
Redistributables	Contains the redistributables needed for CHARON running (already installed, except the license alert script)
Virtual Machines Manager	Contains the most up-to-date version of CHARON Virtual Machines Manager
Virtual Machine Templates	Contains templates for creating CHARON Virtual Machines
Virtual Machines	Contains the CHARON Virtual Machines
Virtual Disk Images	Contains the CHARON virtual disk images, for example "charoncp012" and "idle_vms_pkg_v3.0"

The "Virtual Machine Templates" directory contains template configuration files to build CHARON Virtual Machine using the CHARON Virtual Machines 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.

> 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 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.

> Network HASP (red dongles) licenses have no restrictions with respect to remote access.


Software license

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

1. Double click the "CHARON Virtual Machines 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:
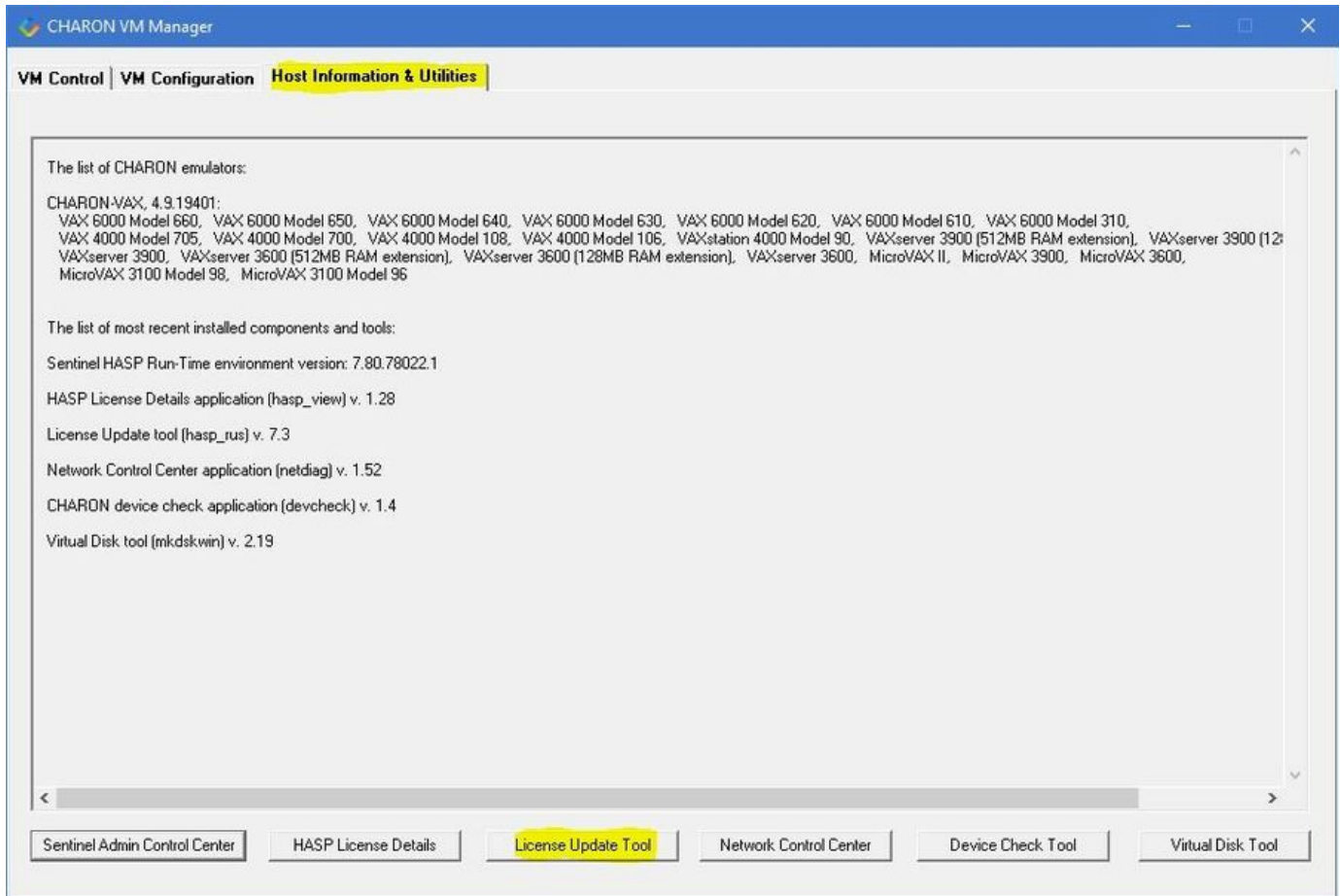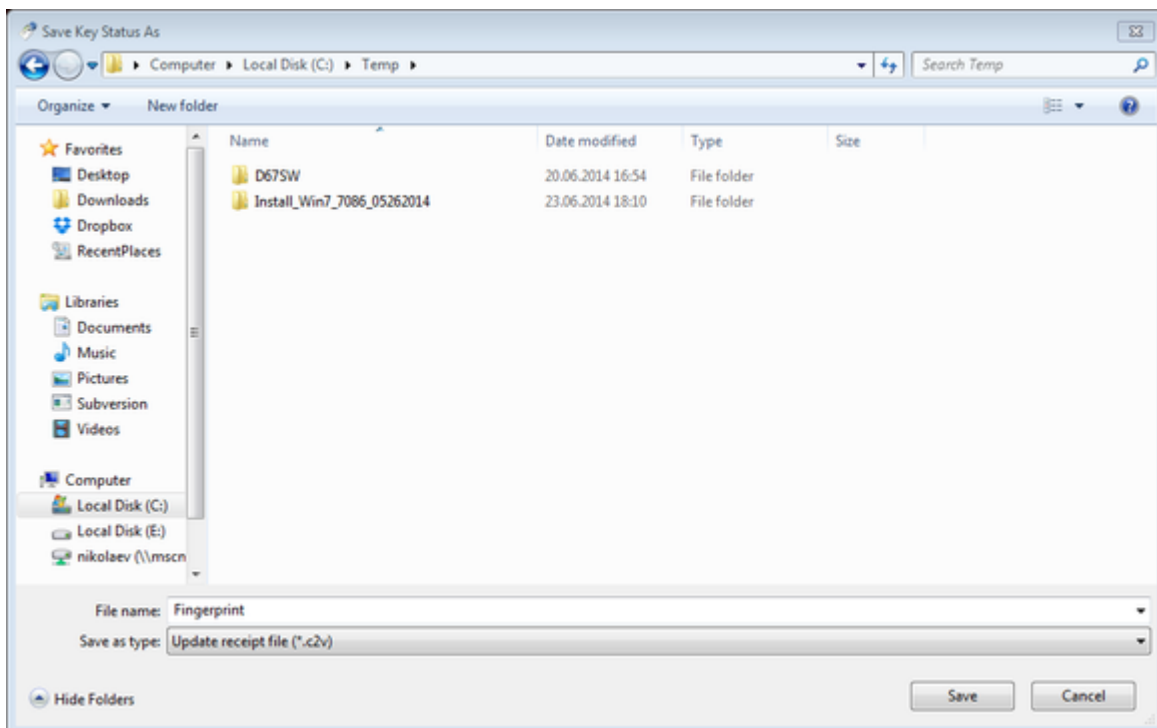
The following default dialog window will appear:
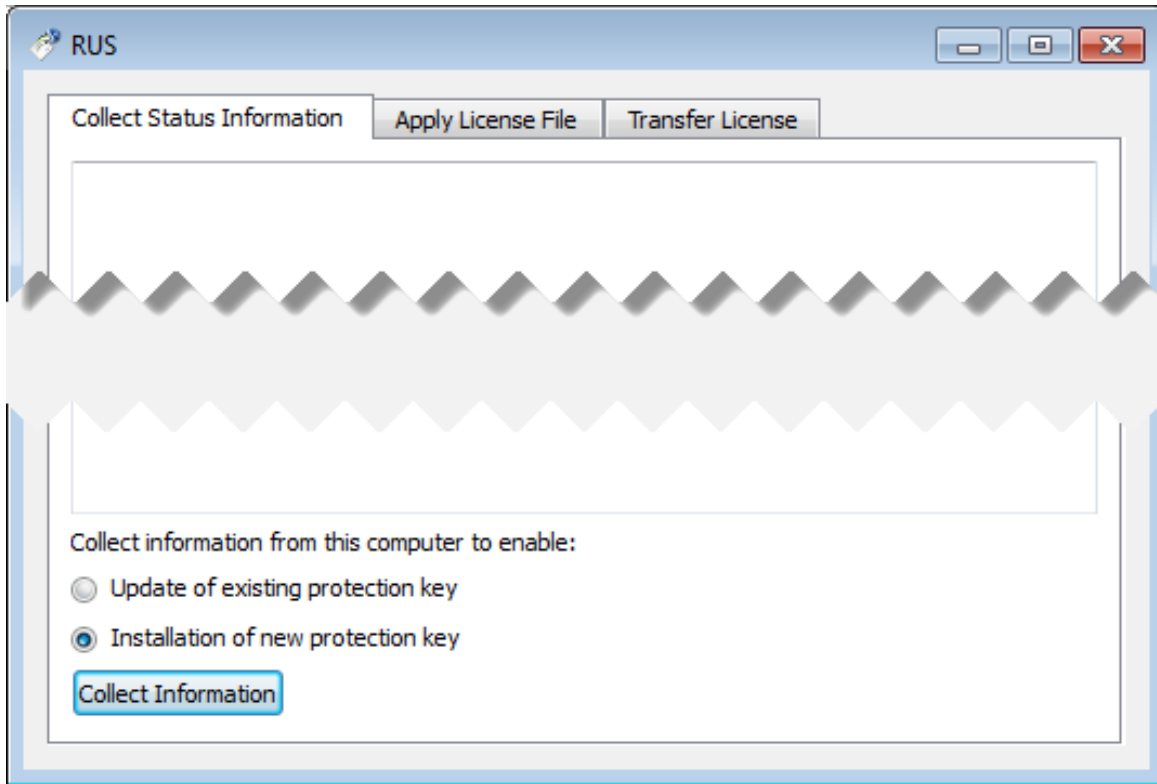
If you're installing a "Provisional" (demo) license, jump to step 5

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:
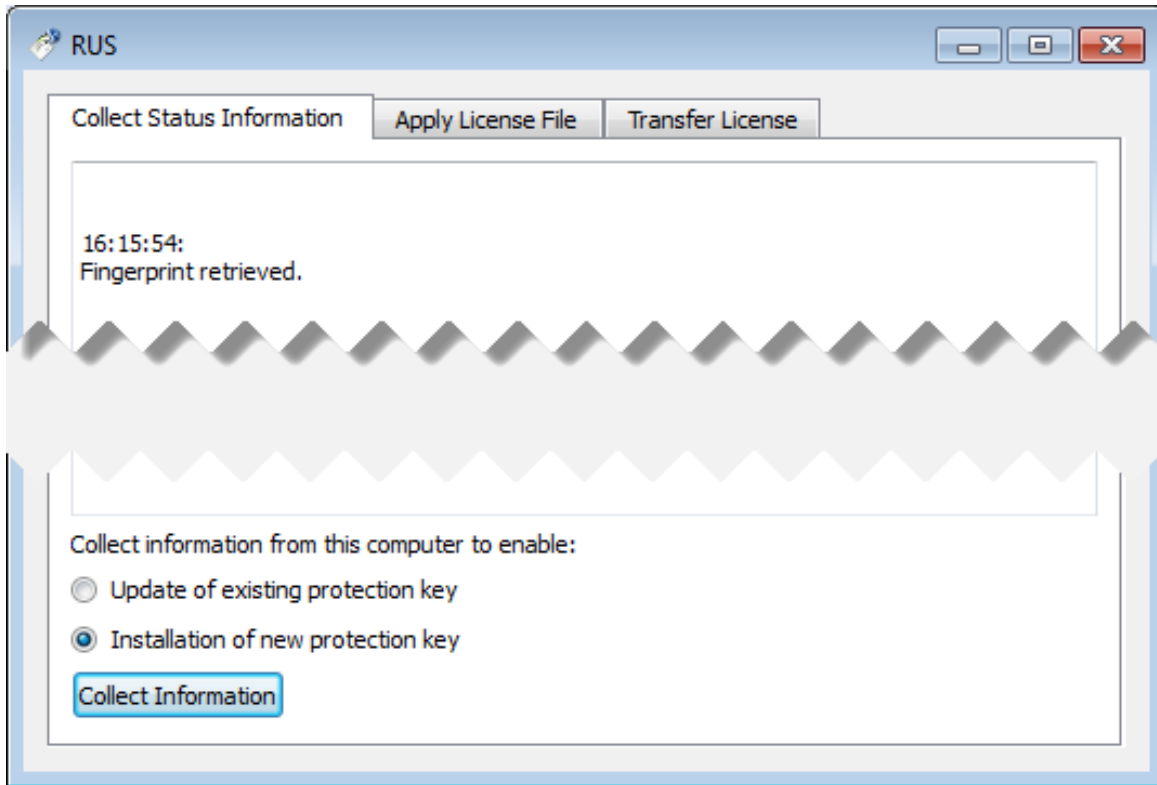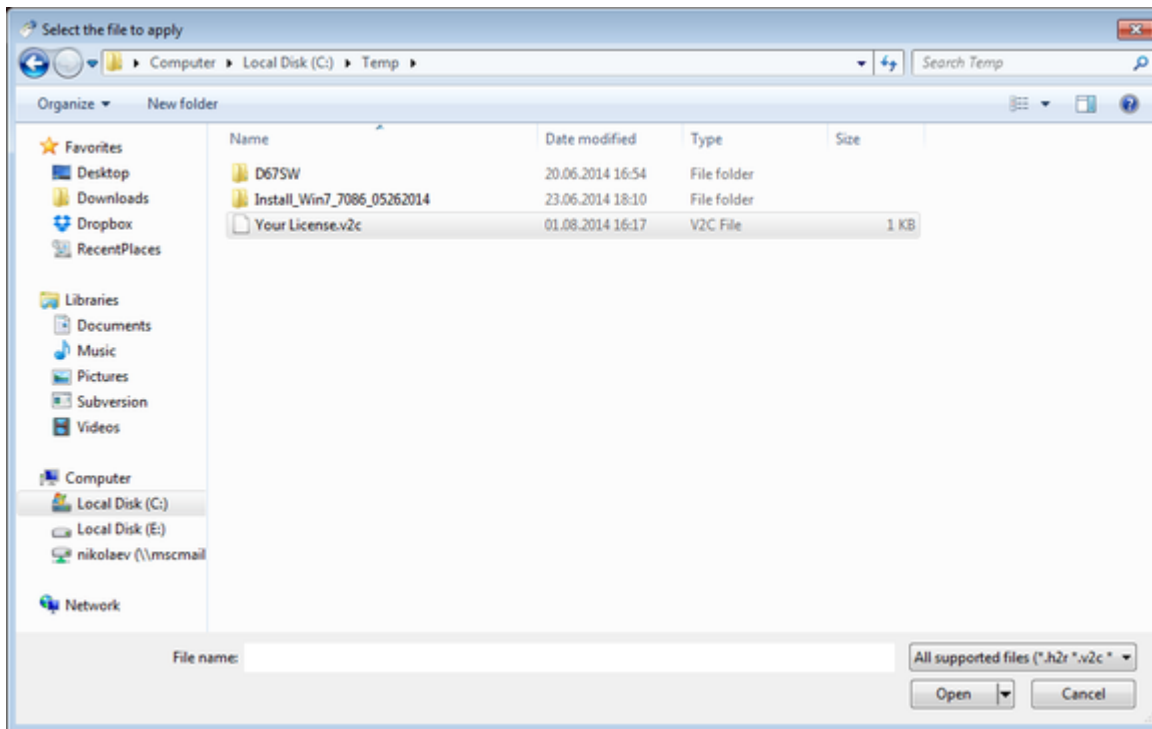

4. Send the resulting file to STROMASYS. In return STROMASYS will provide you with a ".v2c" file, for example "Your License.v2c".

5. Copy the received v2c file to CHARON host in the folder of your choice and select the "Apply License File" tab:
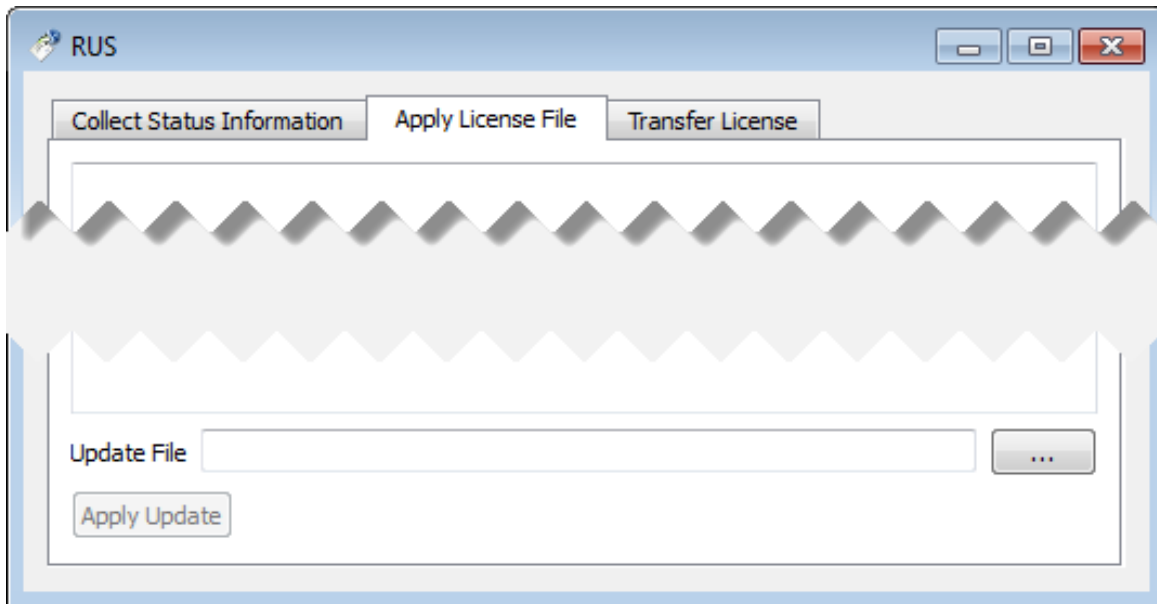
6. Press the "..." button and browse for the received v2c file.
   Example:

Press the "Open" button to apply the license.
7. From the "CHARON Virtual Machines 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.
8. Ensure that the software license is now visible in the "**Sentinel Keys**" section of the **ACC**.

Software licenses are always network licenses. Hence, they have no restrictions with respect to being displayed or accessed via a remote connection.

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 Machines 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:
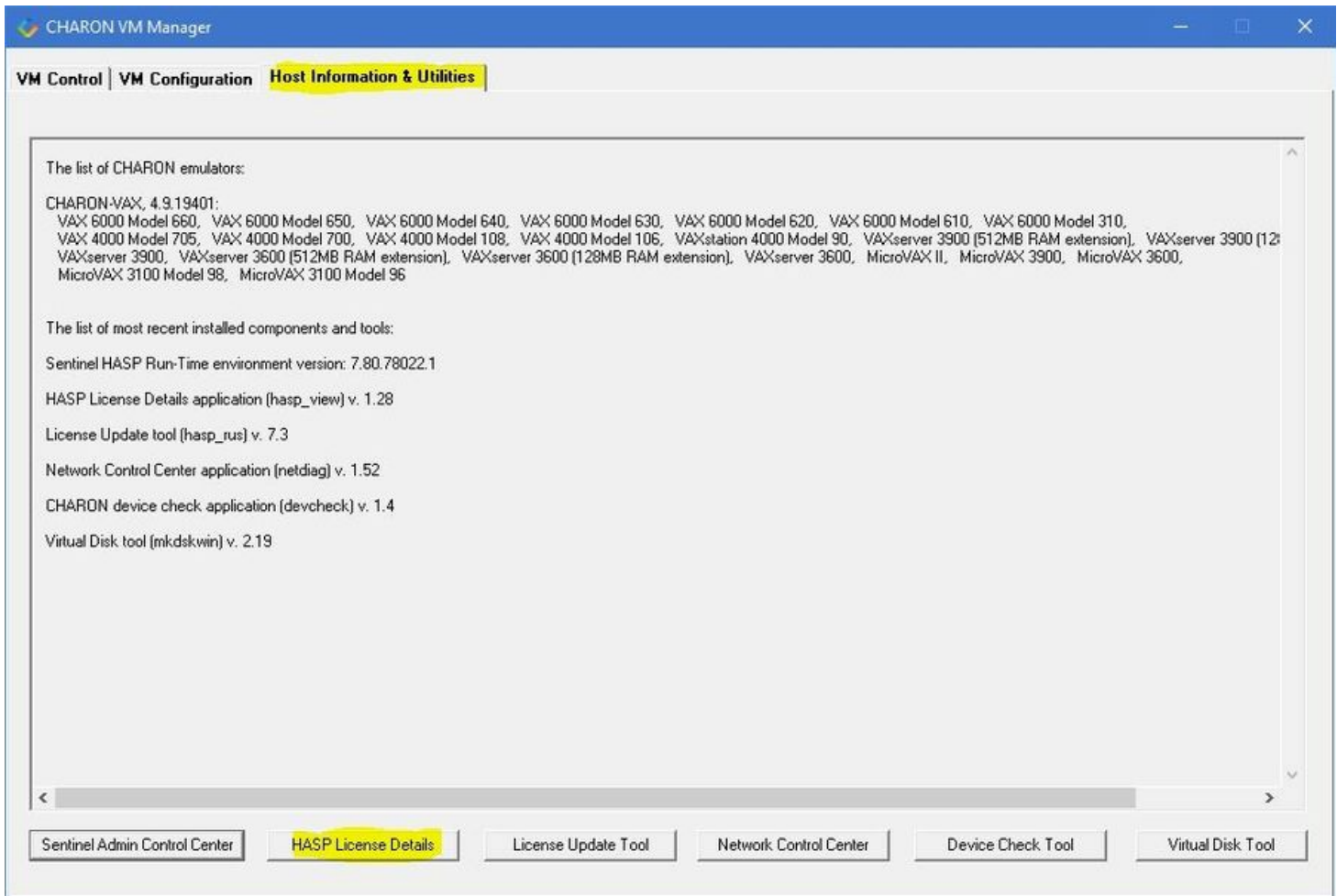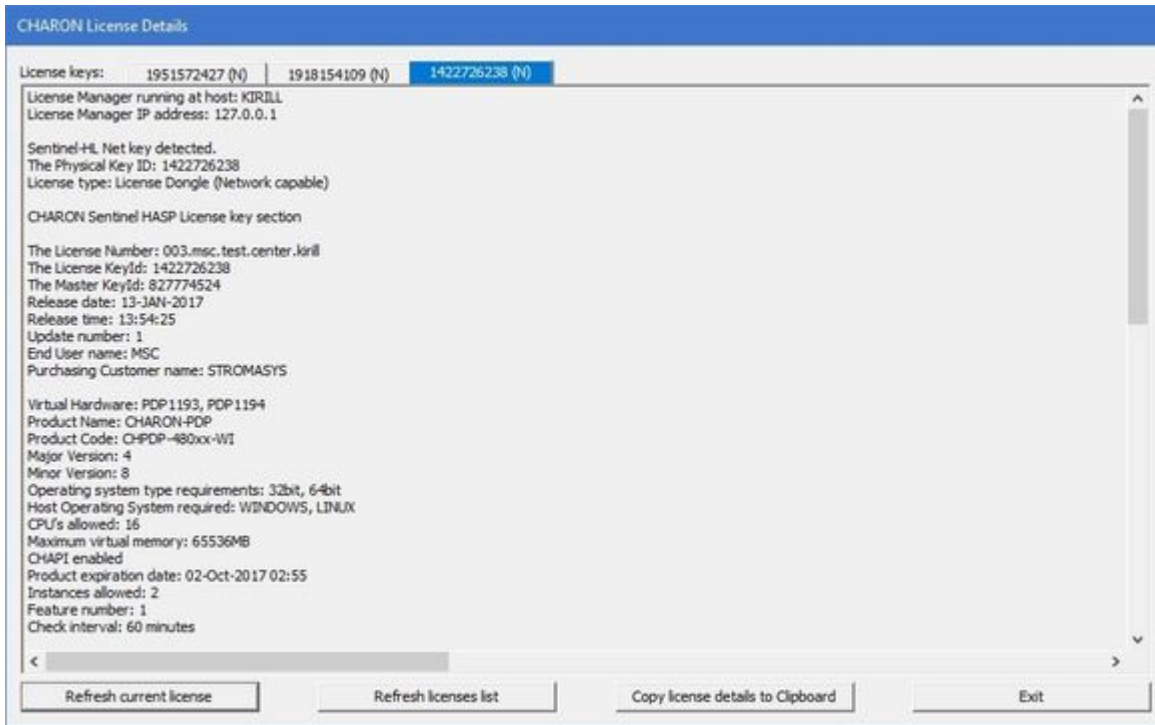
- Verify the the license content is displayed correctly and no error message is shown
- Verify the content of the license looks correct. Check for example the license number, the major and minor versions, the minimum and maximum build numbers, the CHARON-AXP products and allowed hardware (CHARON-AXP 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 Machines 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 "**Sentin el 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.

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.

> 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

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 Machines 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:
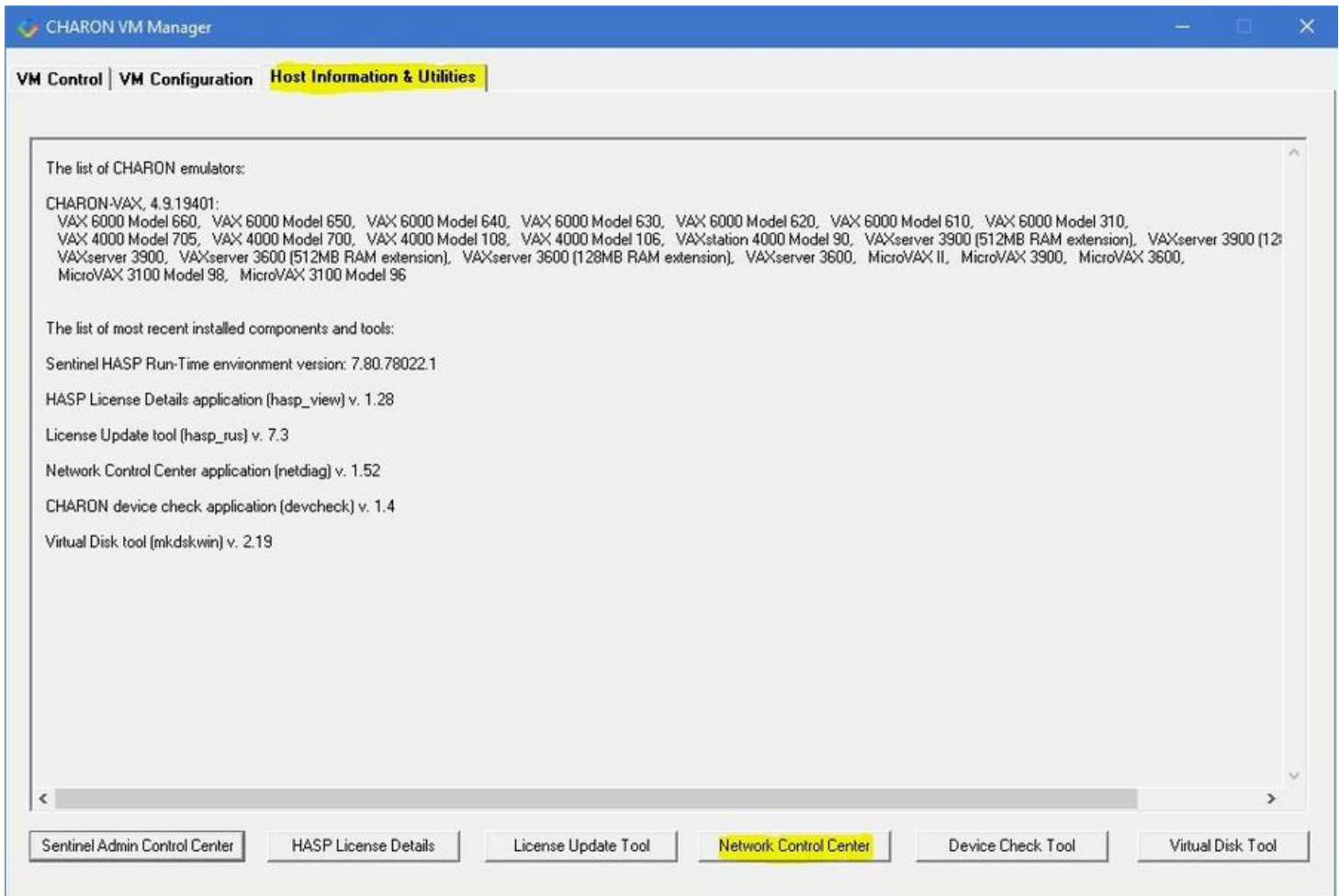
In the appearing dialog, select "Configure NIC":
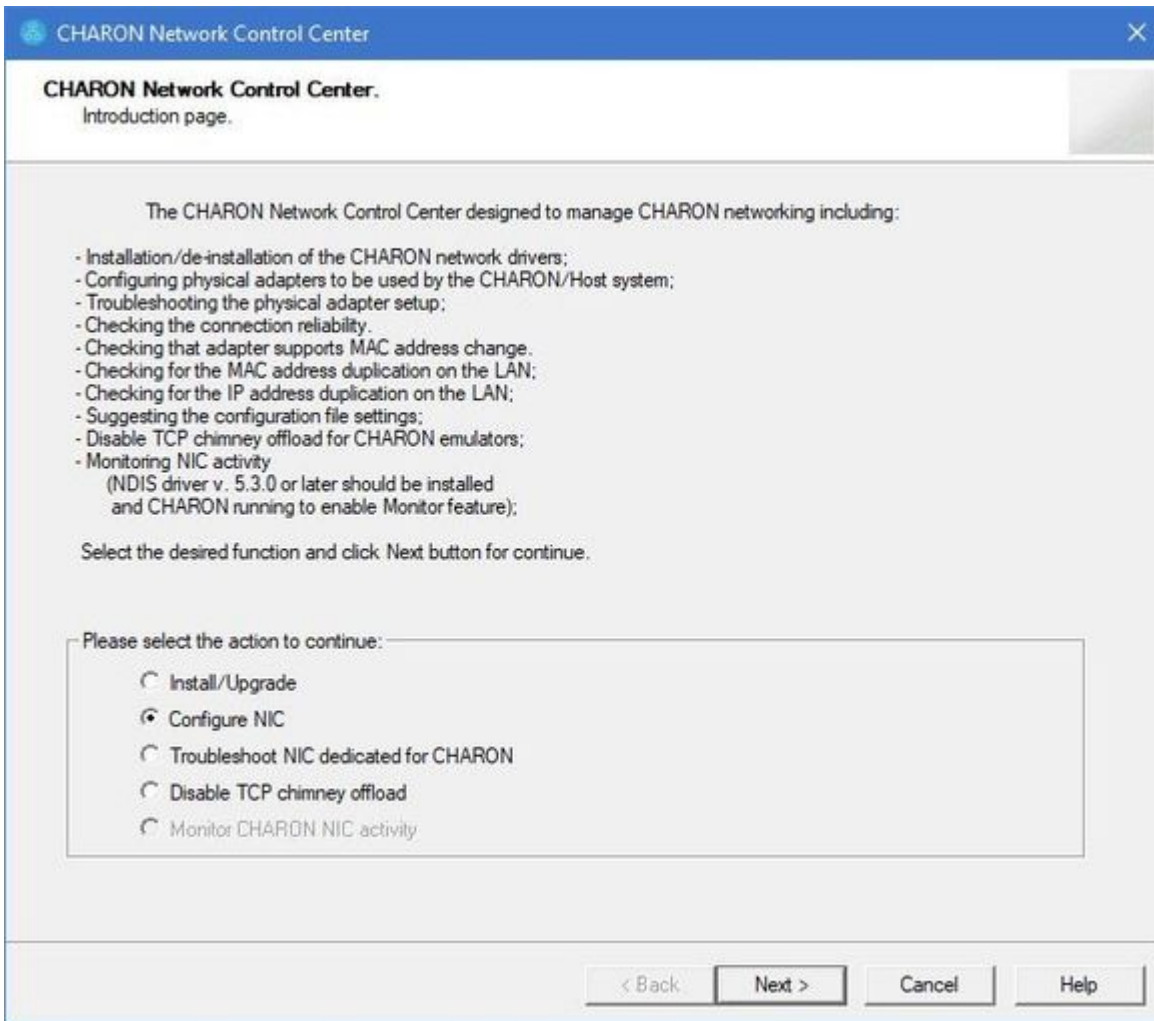
Press the "Next" button, the following dialog will appear:
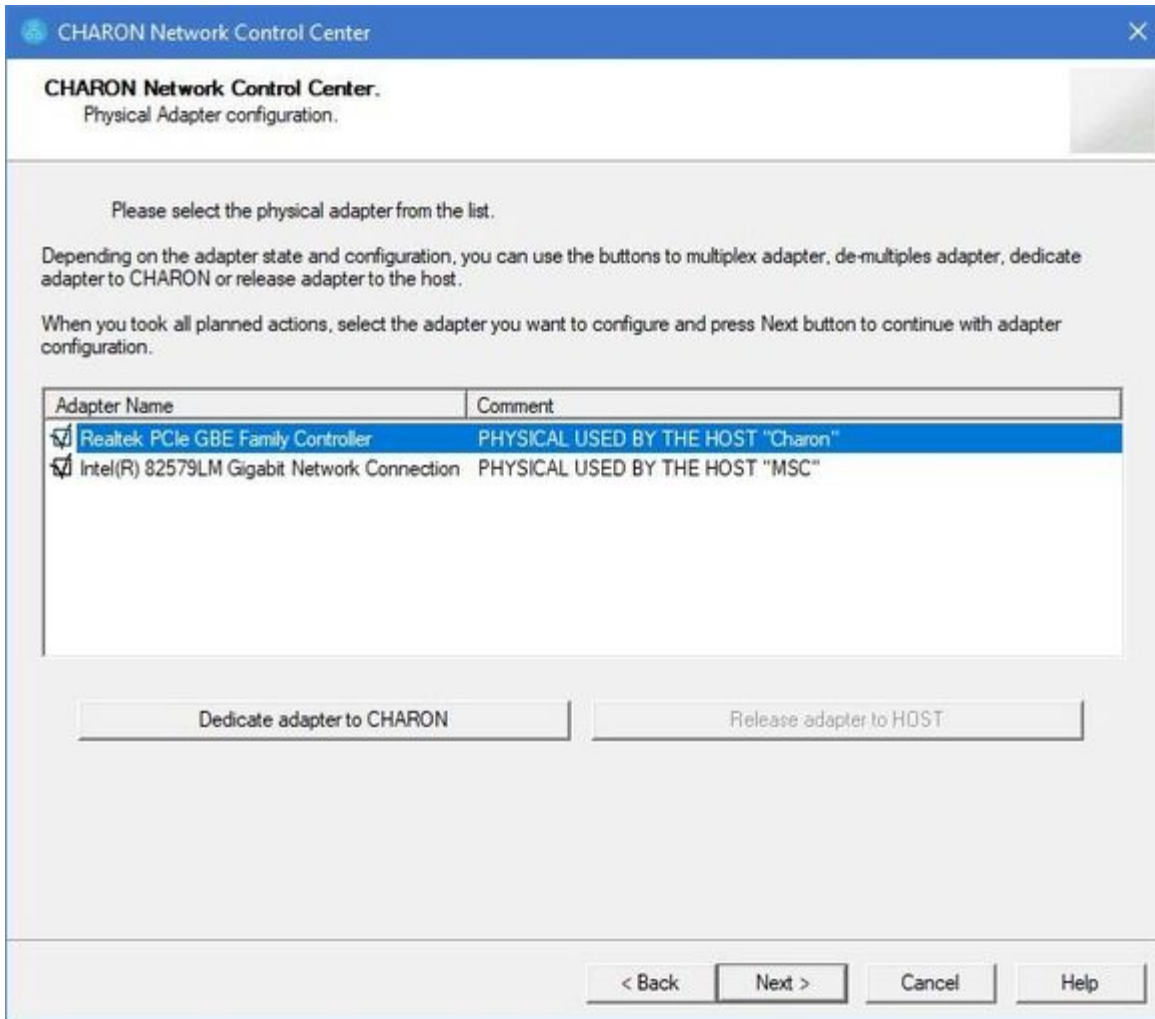
Select the interface to be dedicated to CHARON (in our example it is Realtek PCIe GBE Family Controller named "Charon") and press the "Dedicate adapter to CHARON" button.

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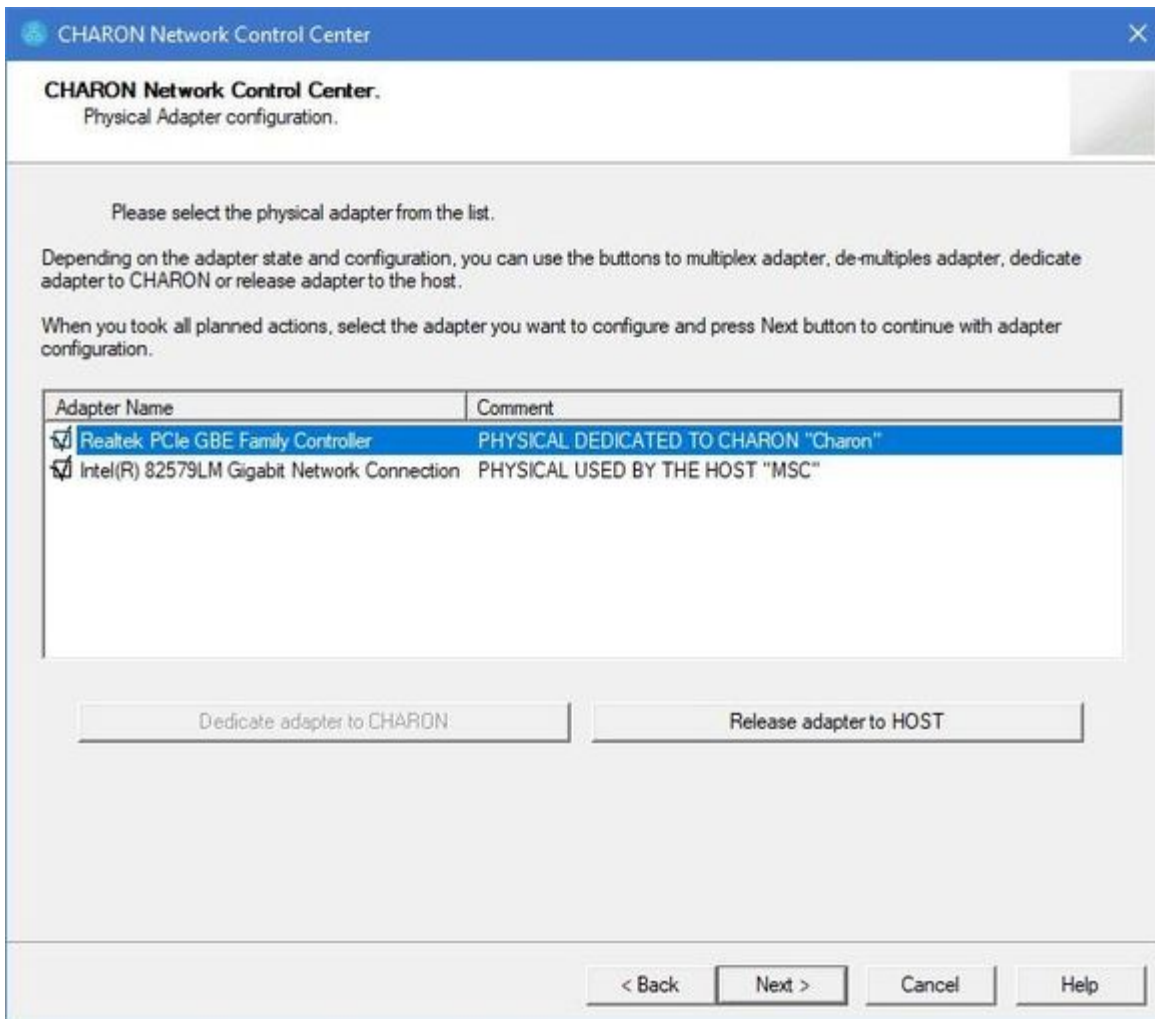
ℹ️ It is possible to release the interface back to the CHARON host by selecting the target interface and pressing the "Release adapter to HOST" button.

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:

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.

Restart the "Network Control Center" utility again and select "Disable TCP chimney offload for CHARON":
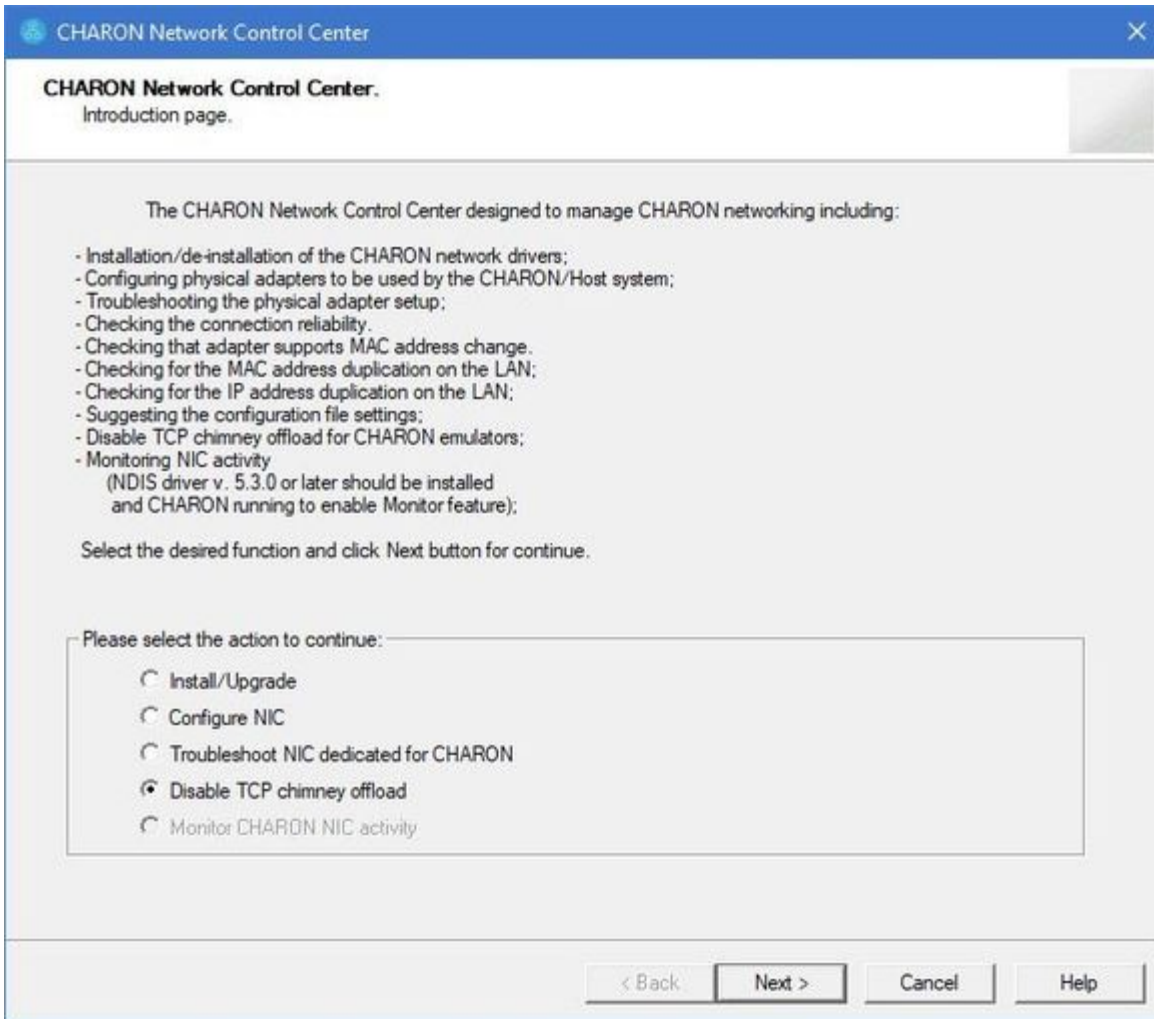
Press the "Next" button to apply the "offload" parameters settings.

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.

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.

> If the CHARON-VAX / CHARON-PDP user belongs to some domain, you have to add this user to the CHARON-GRP group as it is described in this article and then reboot 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 standard privileges. Please consult with your Windows User's Guide on details.
3. Add this user to the CHARON-GRP group as it is described in this article and then reboot CHARON host.
4. Login as the created user.

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

Starting CHARON Virtual Machines Manager

CHARON execution is controlled by the "CHARON Virtual Machines 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 Machines Manager.

Desktop shortcut:

"Start" menu shortcut:
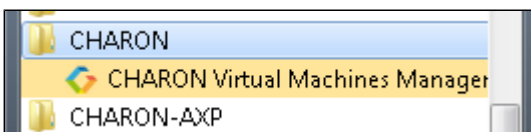
The CHARON Virtual Machines Manager runs automatically at Windows startup. If it is minimized to tray, it can be re-opened in the following way:
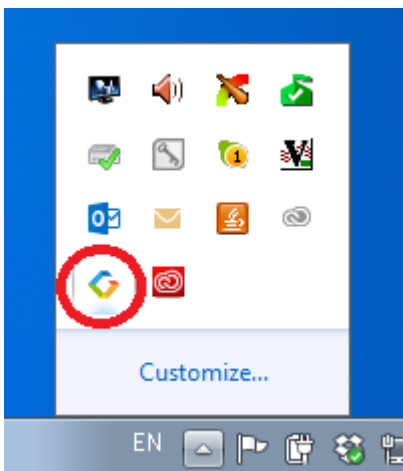
After clicking the shortcuts, the main dialog of CHARON Virtual Machines Manager appears.

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.

> By default the CHARON templates do not contain any specification for disks and network; you need to update the configuration file created from the template to specify your specific disk and network configuration.

Start the CHARON Virtual Machines Manager (see above) and press the "Create VM from Template" button:
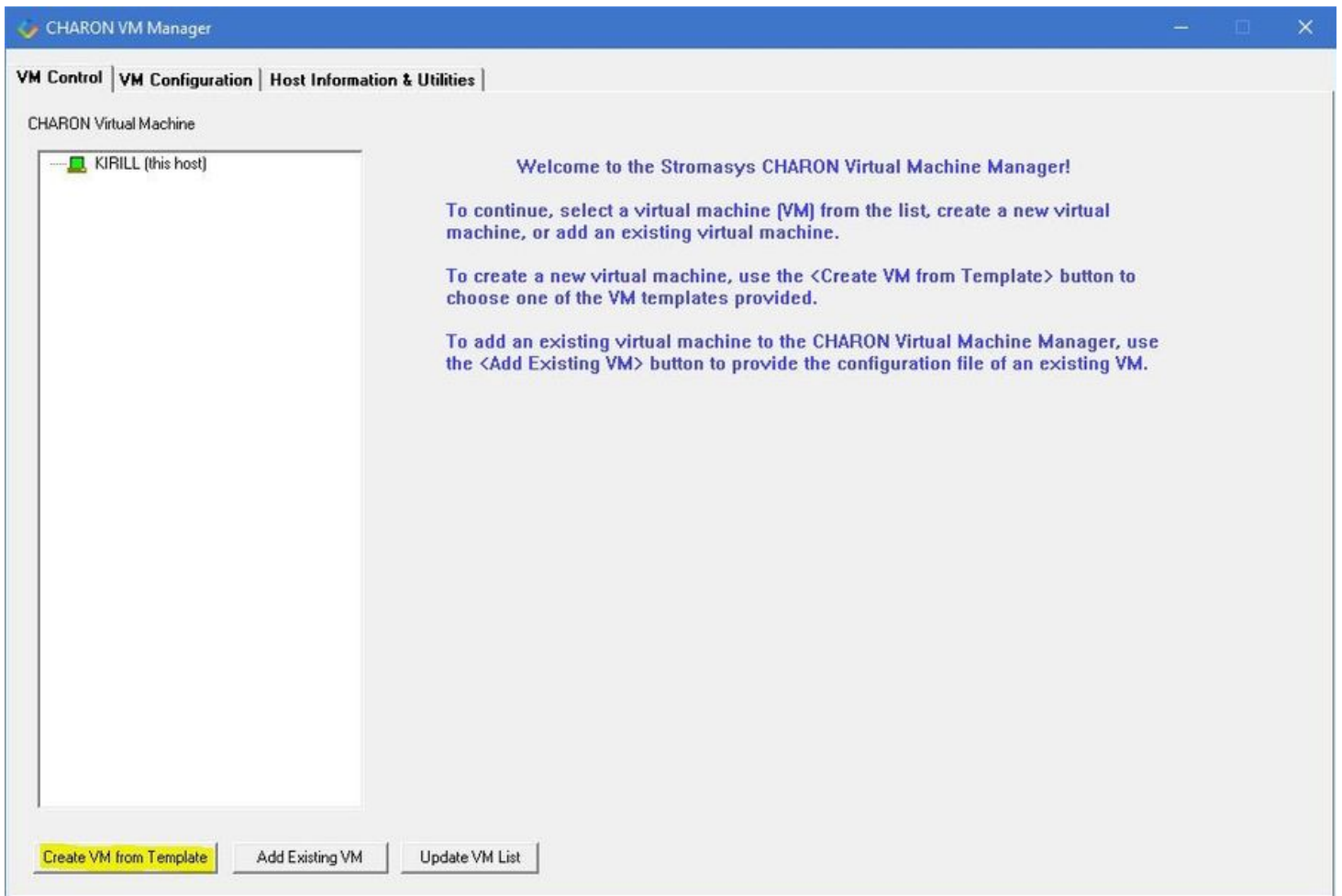
In the appearing dialog select the VAX/PDP11 hardware model:

Create Virtual Machine from a Template

List of installed products

Select hardware model:

MicroVAX 3100 Model 96

| MicroVAX 3100 Model 96 |
| MicroVAX 3100 Model 98 |
| MicroVAX 3600 |
| MicroVAX 3900 |
| MicroVAX II |
| VAX 4000 Model 106 |
| VAX 4000 Model 108 |
| VAX 4000 Model 700 |
| VAX 4000 Model 705 |
| VAX 6000 Model 310 |
| VAX 6000 Model 610 |
| VAX 6000 Model 620 |
| VAX 6000 Model 630 |
| VAX 6000 Model 640 |
| VAX 6000 Model 650 |
| VAX 6000 Model 660 |
| VAXserver 3600 |
| VAXserver 3600 (128MB RAM extension) |
| VAXserver 3600 (512MB RAM extension) |
| VAXserver 3900 |
| VAXserver 3900 (128MB RAM extension) |
| VAXserver 3900 (512MB RAM extension) |

Change   Explore...

Co

Change   Explore...

Create   Cancel

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.

Specify a name for the VM and press the "Create" button:
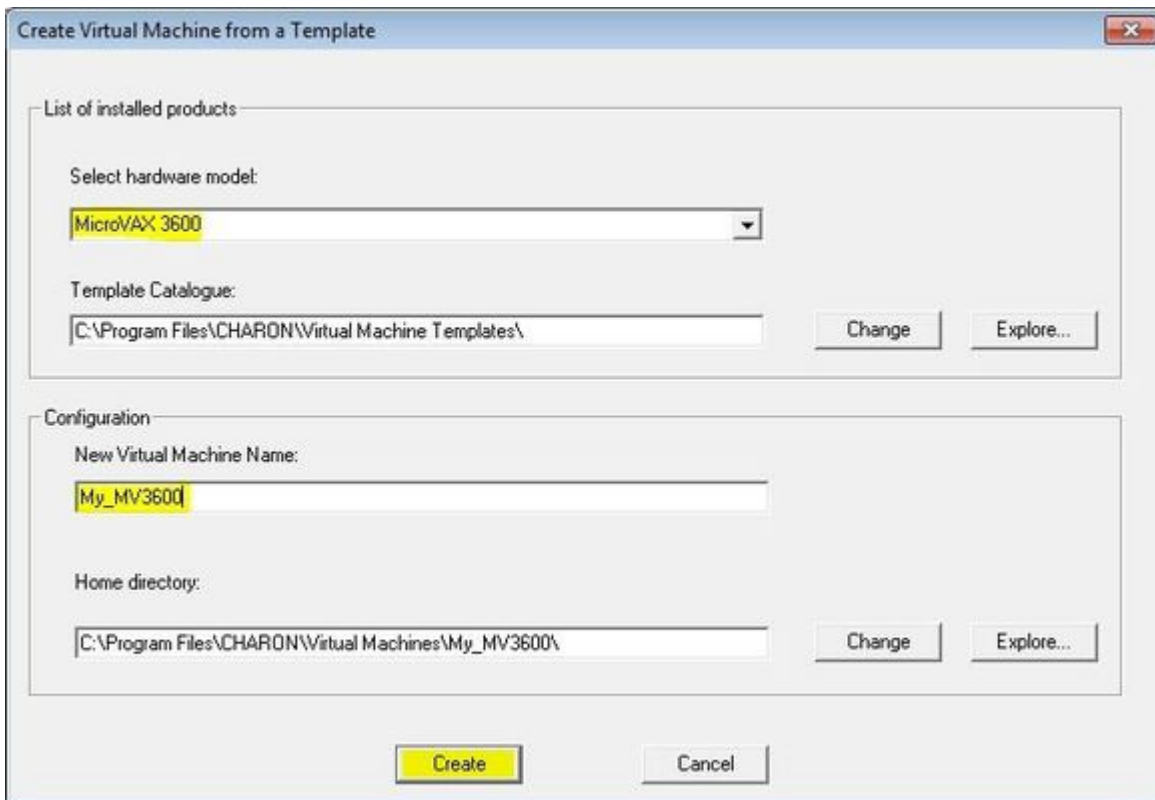
Create Virtual Machine from a Template

List of installed products

Select hardware model:

MicroVAX 3600

Template Catalogue:

C:\Program Files\CHARON\Virtual Machine Templates\   Change   Explore...

Configuration

New Virtual Machine Name:

My_MV3600

Home directory:

C:\Program Files\CHARON\Virtual Machines\My_MV3600\   Change   Explore...

Create   Cancel

The CHARON Virtual Machines Manager will report the VM has been created successfully:
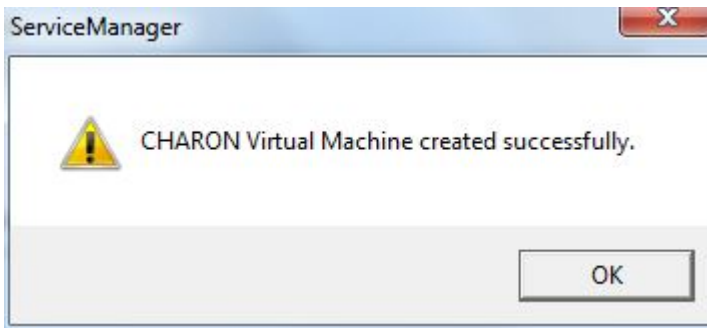
It will then appear in the left pane:
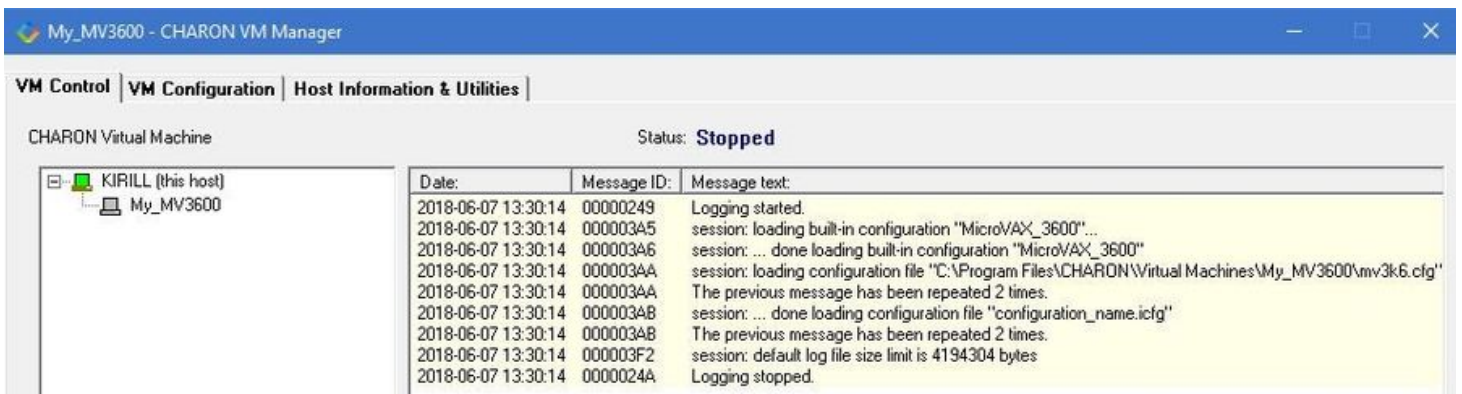
Creating a CHARON VM using an existing configuration file

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

Open the CHARON Virtual Machines Manager (see above) and press the "Add Existing VM" button:
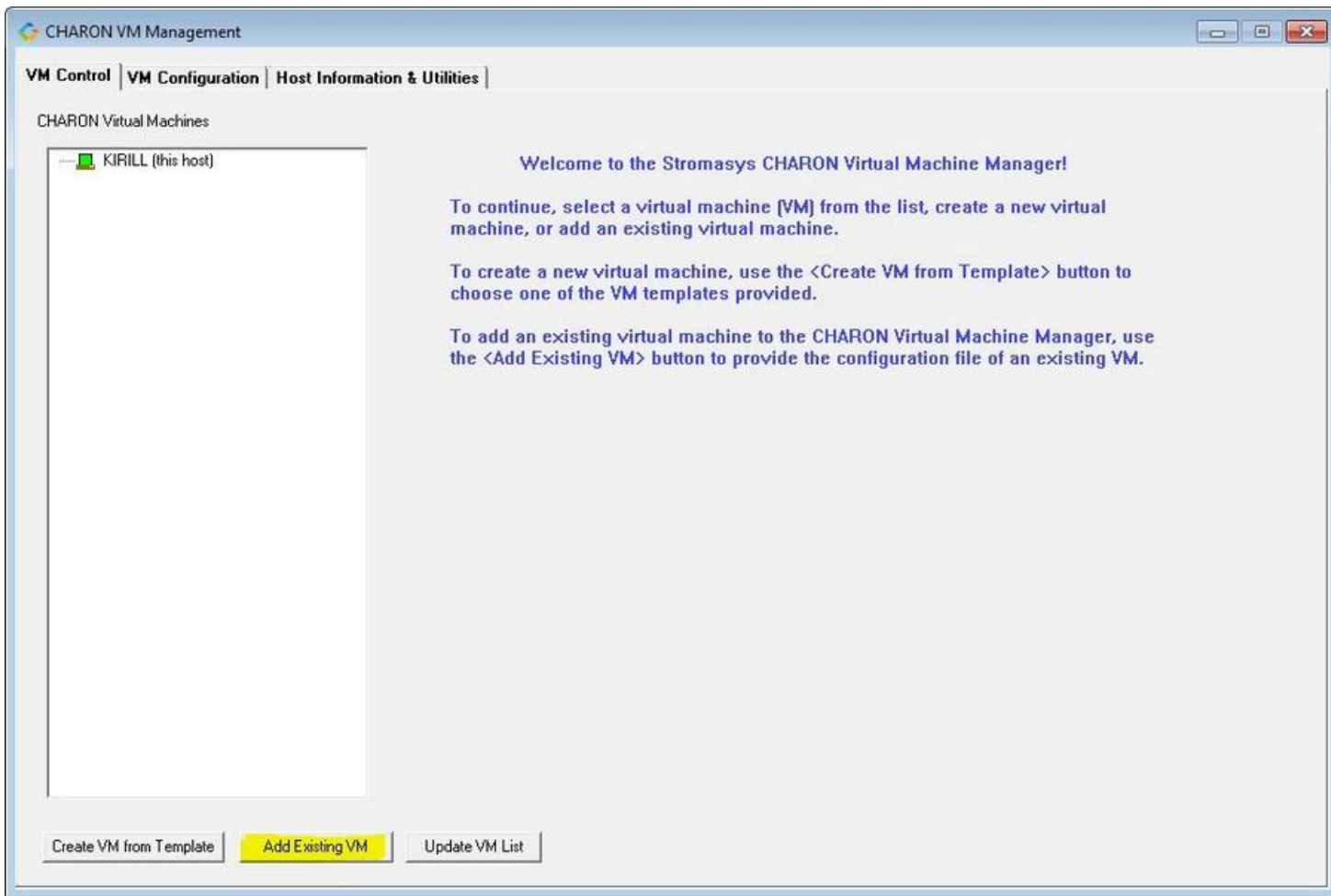
In the appearing dialog select the target configuration file by pressing the "Browse" button:

Select the configuration file and press the "Open" button:
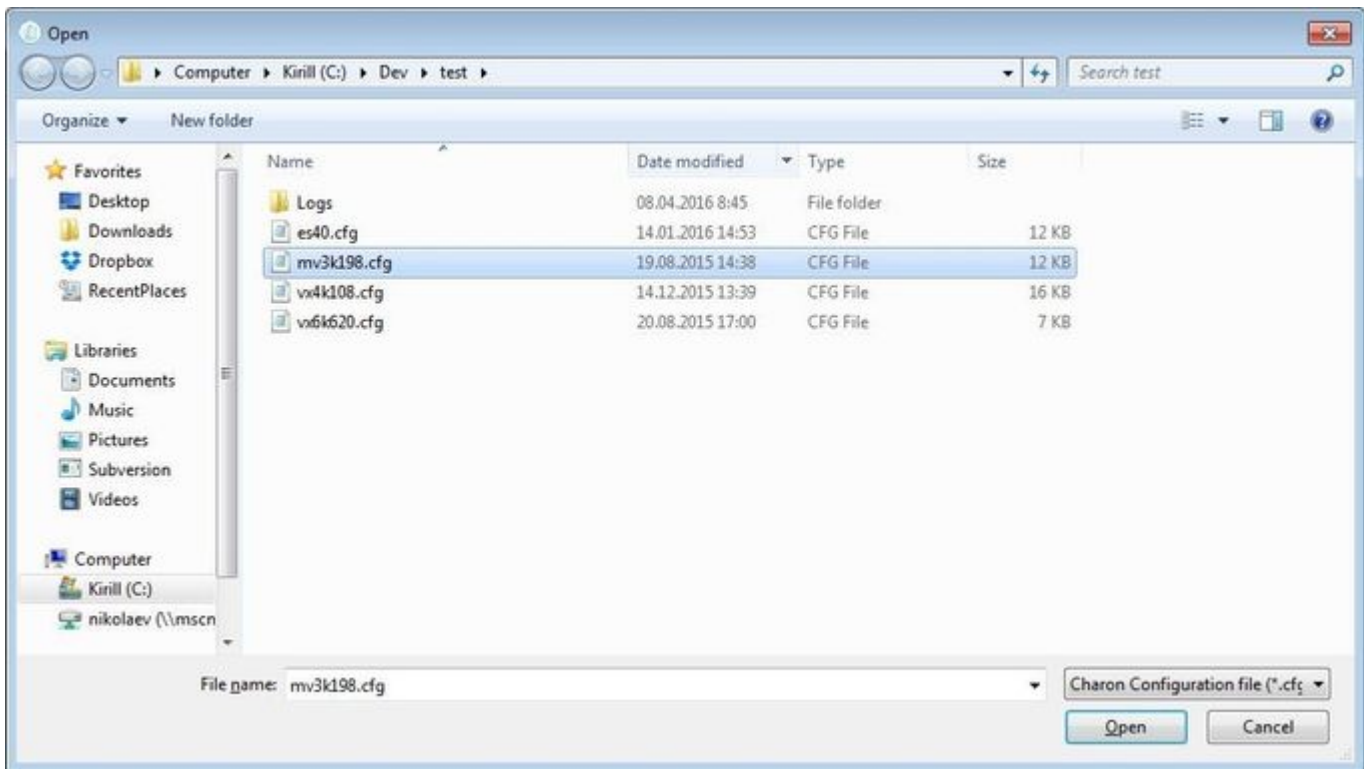
Never select a configuration file template provided in the Charon installation folder as these files could be overwritten with patching.

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:
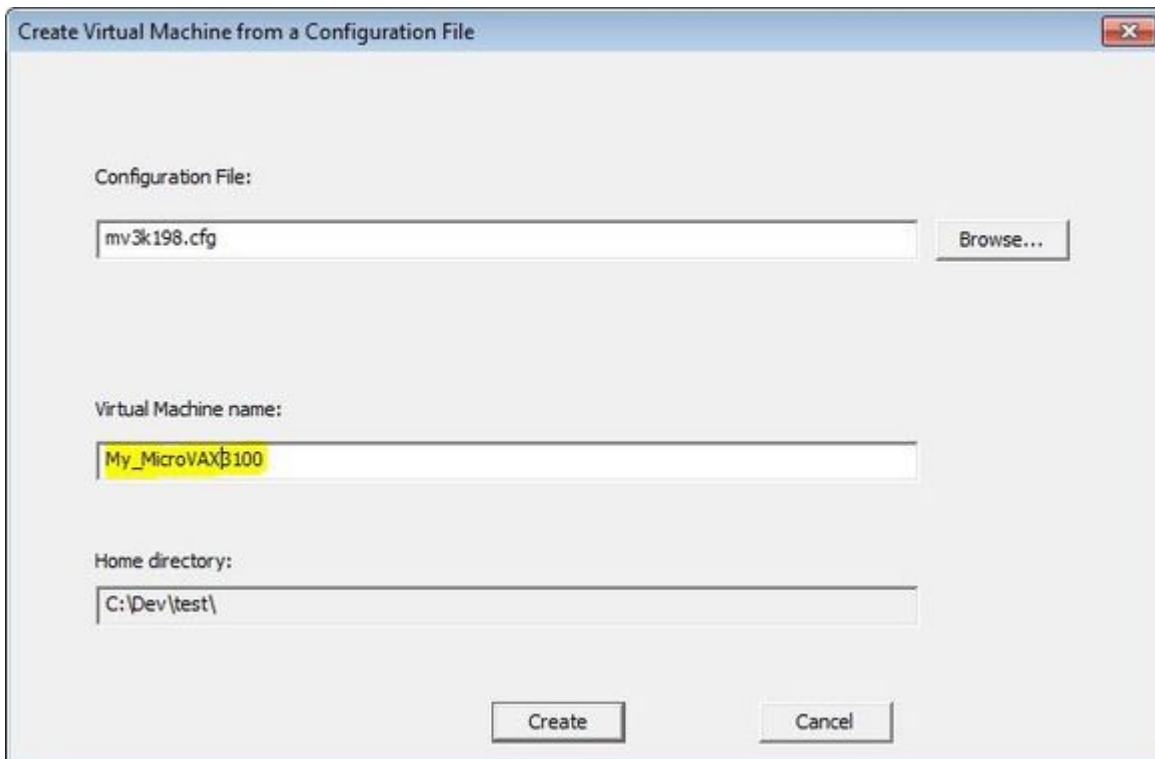
The CHARON Virtual Machines Manager will report the VM has been created successfully:
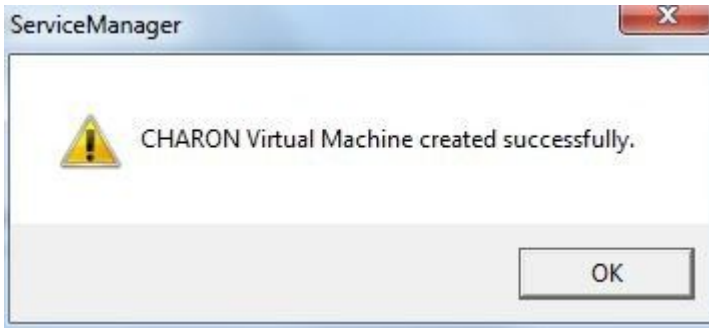
It will then appear in the left pane:
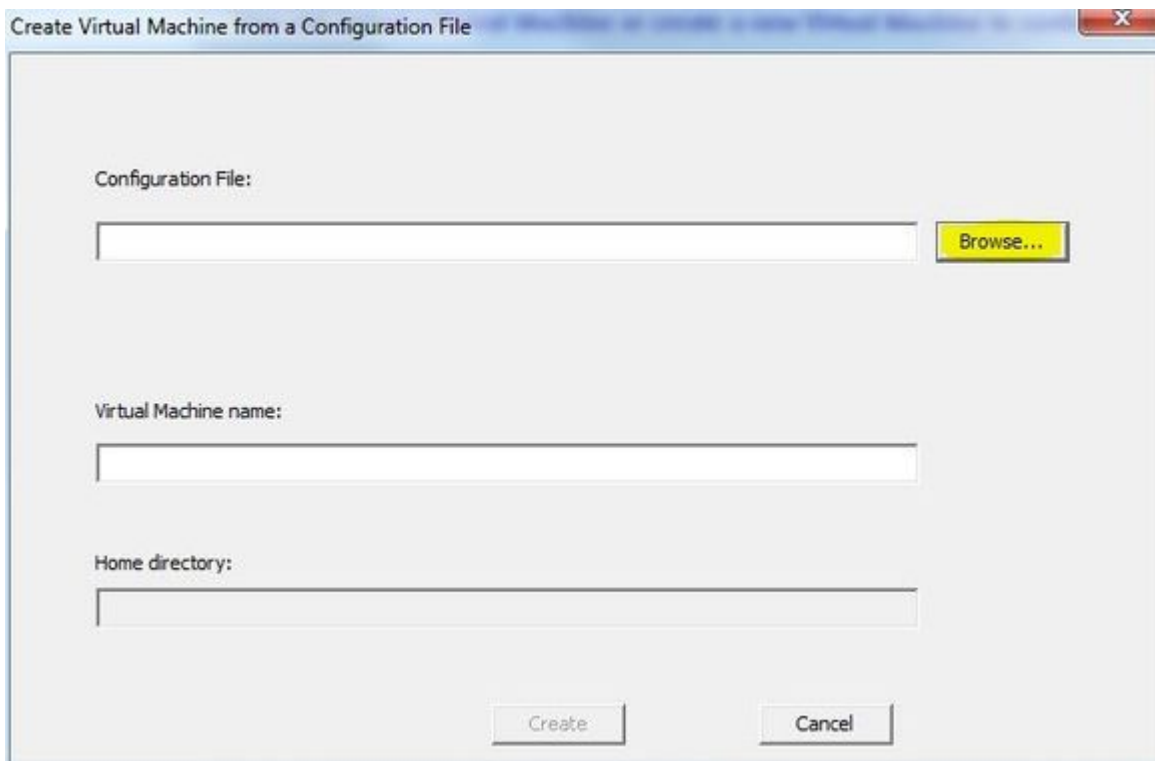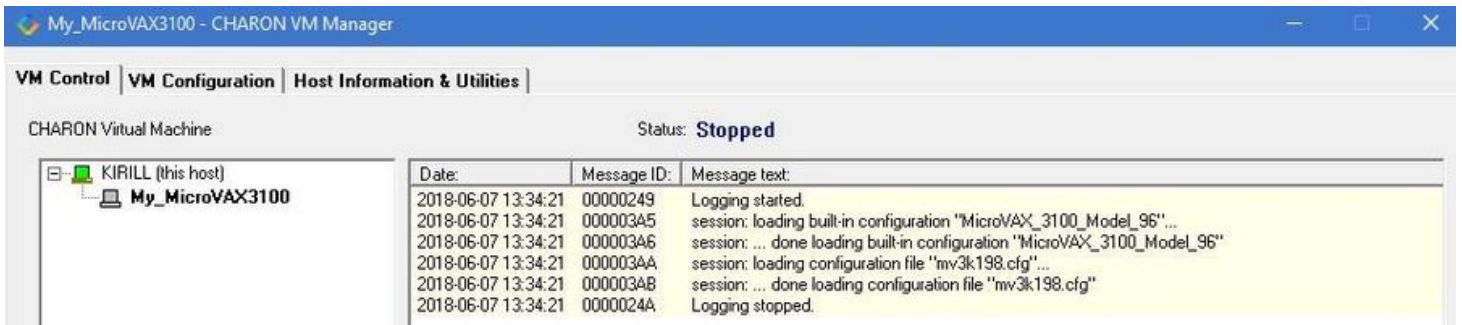

Running a CHARON Virtual Machine

Select the target CHARON VM in the left pane and press the "Start" button:
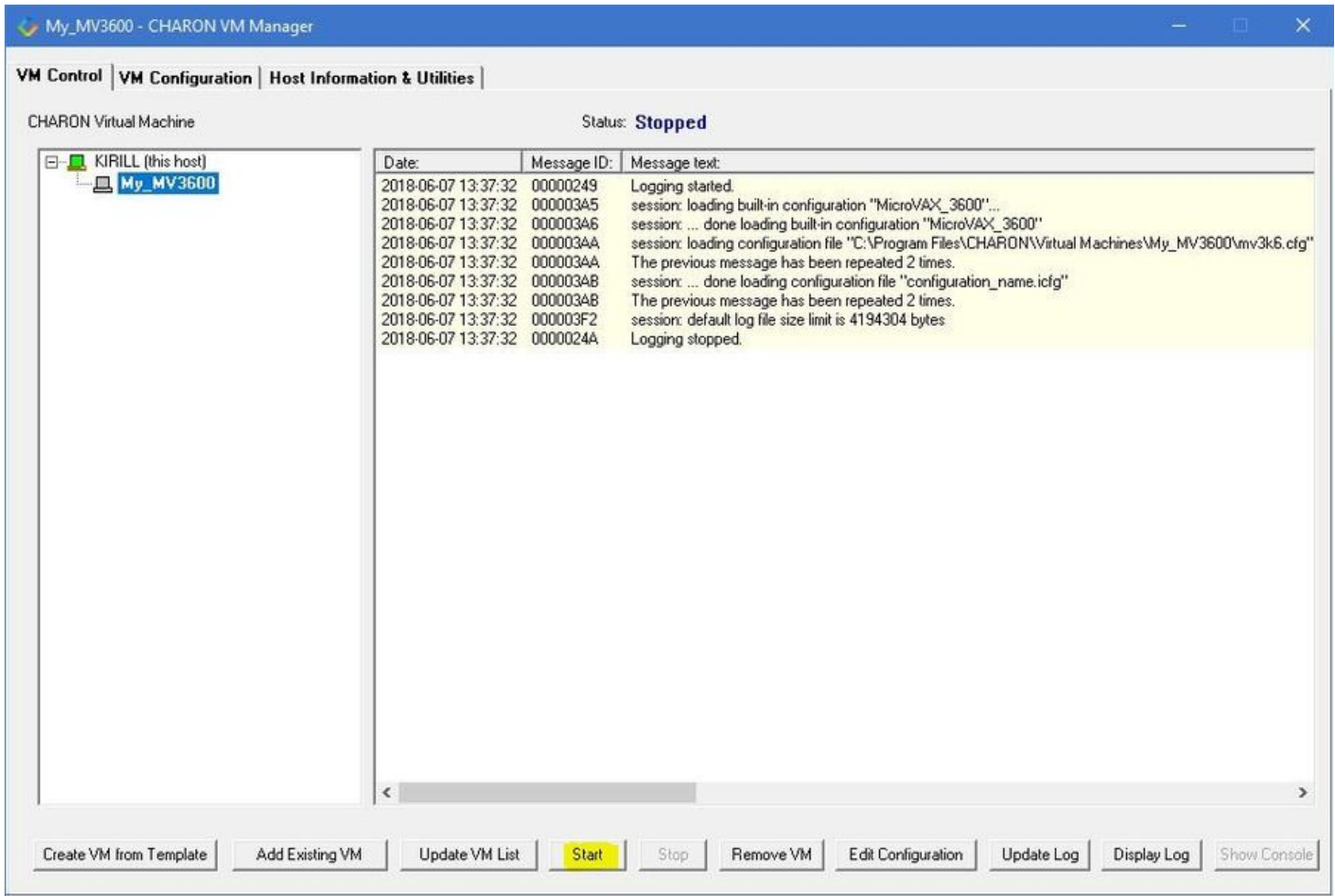
CHARON Virtual Machines Manager will display the execution log:

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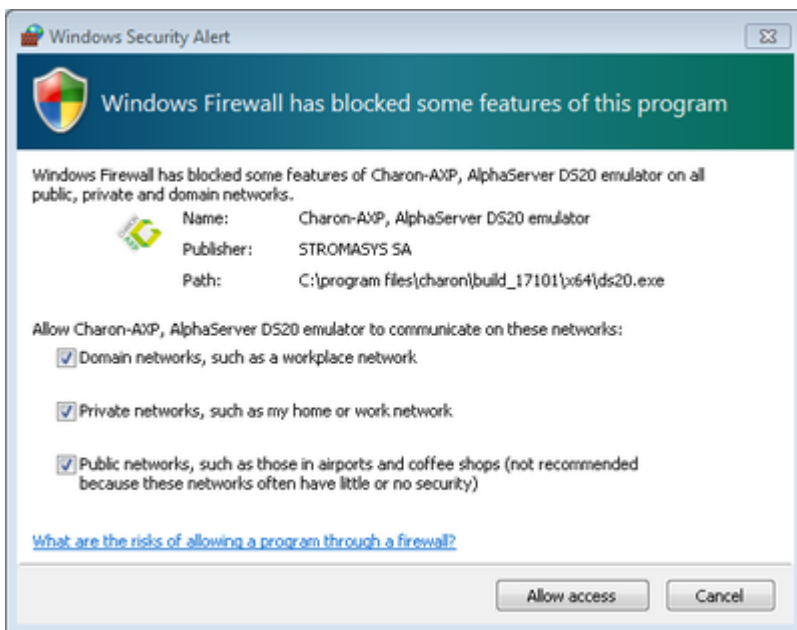
Check the desired options and press the "Allow access" button.

If you closed the CHARON VM console, it is always possible to re-open it by pressing the "Show Console" button:
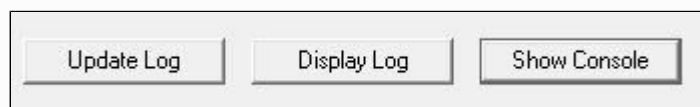
ⓘ Closing the console will not stop the CHARON Virtual Machine

Stopping a CHARON Virtual Machine

⚠ Note: before stopping the virtual machine from the CHARON Virtual Machines 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:

Removing a CHARON Virtual Machine

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

The Virtual machine has to be stopped before removing it (see Stopping a CHARON Virtual Machine chapter above)

A confirmation dialog will be displayed:
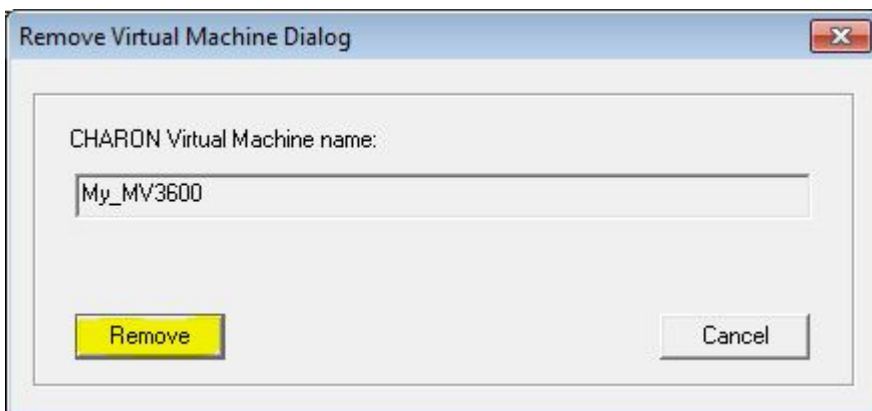
Press the "Remove" button and confirm the CHARON VM removal by pressing the "Yes" button:

Modifying a CHARON Virtual Machine

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

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:

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: `C:\Program Files\CHARON\Virtual Machines\My_MV3600\`   Explore...   Create Virtual Disk

The MkDisk utility will be called for this operation:
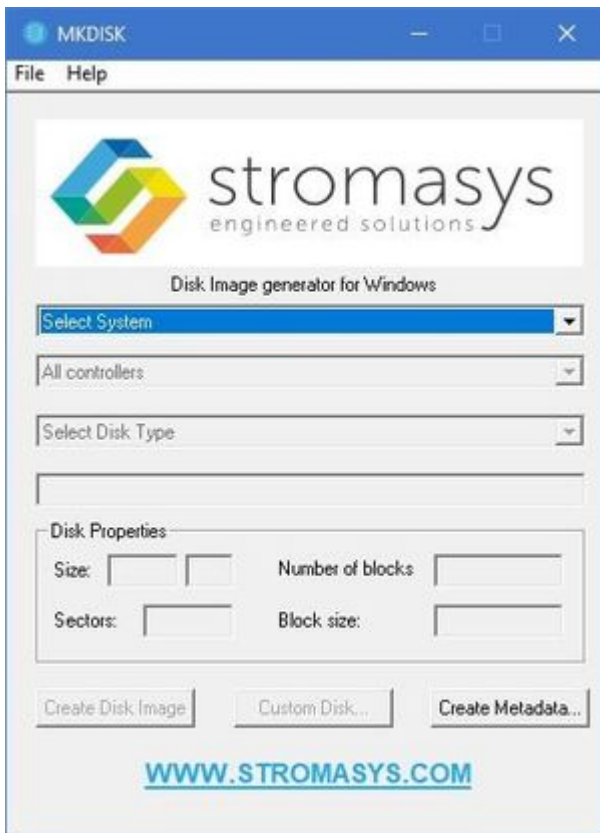
Refer to the "Utilities" section of this User's Guide for more information about the MkDisk utility.

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: `mv3k196.cfg`   Edit Configuration File

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

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 Machines 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:

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 to a specific version, only the patched version will appear, not the vanilla one. If you need to rollback to a vanilla version, please uninstall the patch according to the documentation provided with the patch.

Select the target 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, 4.9.19401   ▼   Apply

Service functions

Additional functionalities are available and described below:

Function	Description

Update Log	The CHARON Virtual Machines 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.
	Right click on top of the CHARON Virtual Machines Manager window and select "Settings":
	In the appearing dialog you can change the interval:
	If it is required to see most up-to-date version of the log, press the "Update Log" button.
Display Log	Available from the VM control tab, bottom right.
	Creates a separate window containing the selected CHARON VM log
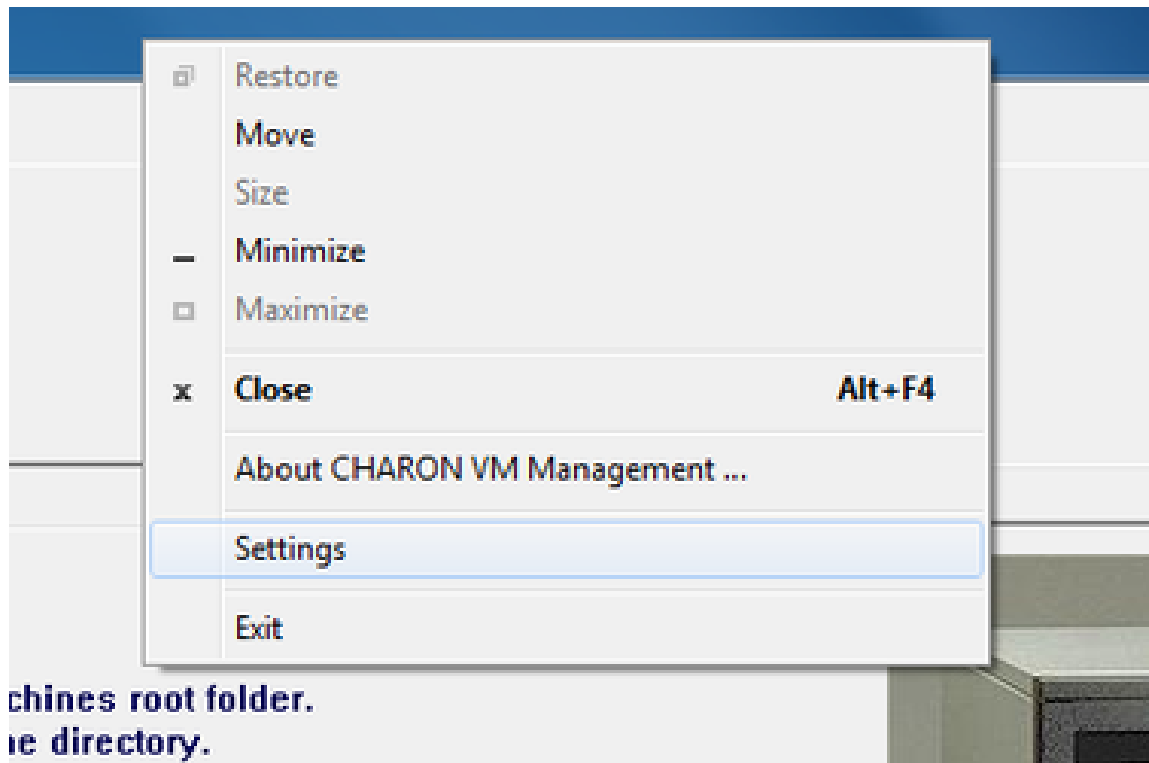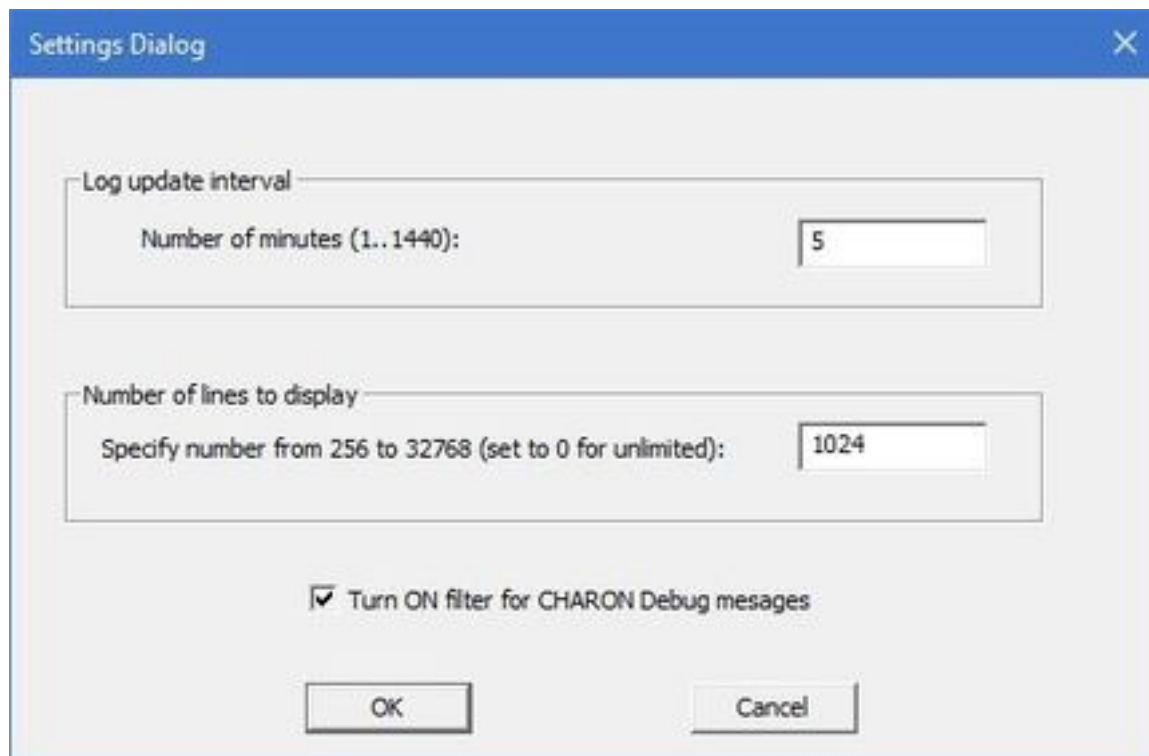
Show Console	Available from the VM control tab, bottom right, and if the VM is running.
	Opens the selected CHARON VM console if it has been closed

Running CHARON utilities

Select the "Host Information & Utilities" tab to get a quick access to the CHARON utilities:
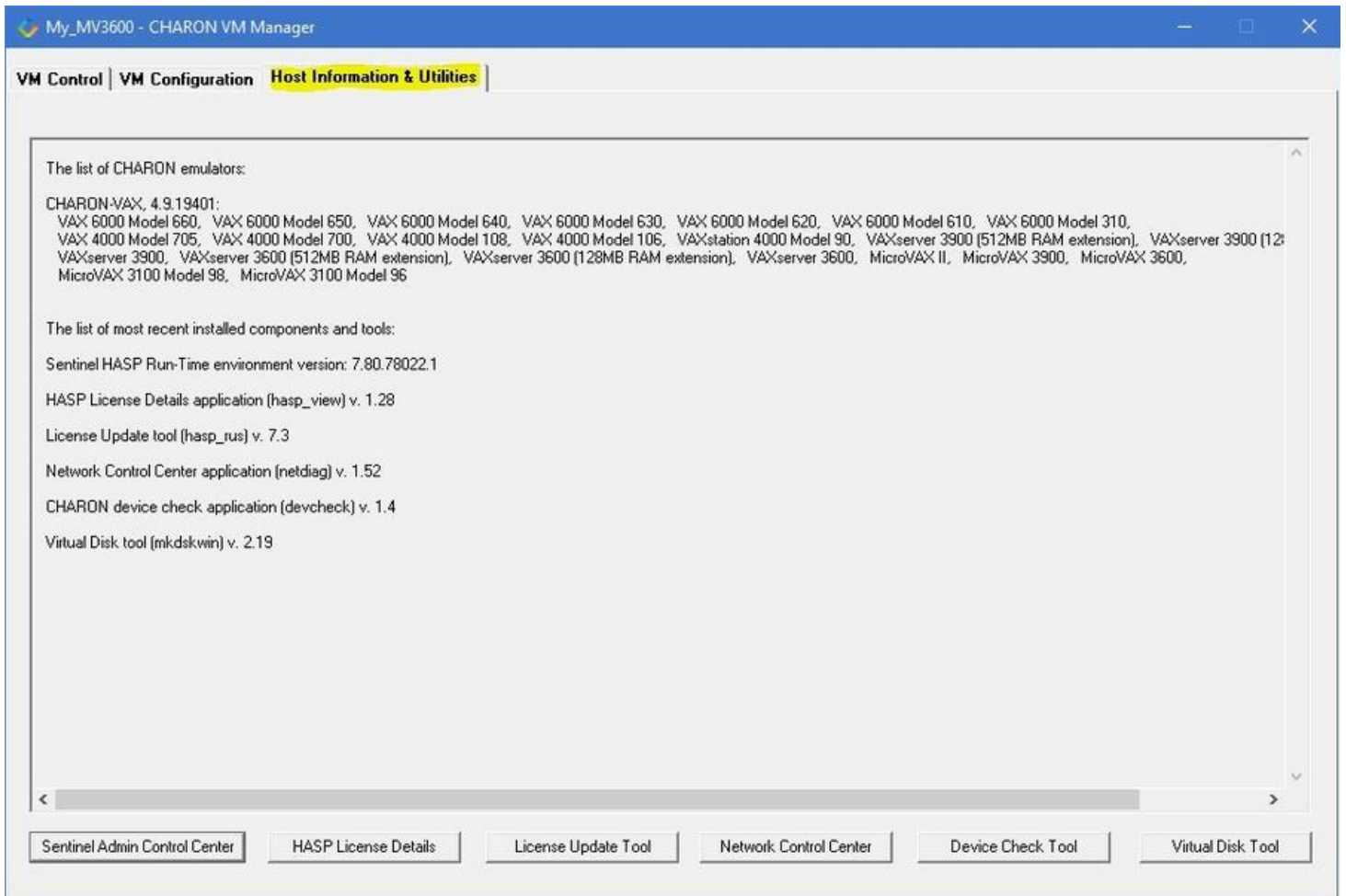
These utilities are described below:

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 Toool	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

When a CHARON Virtual Machine (VM) is created from a template using the CHARON Virtual Machines 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 Machines 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:

Notepad will be used to edit the configuration file:

```
mv3k6.cfg - Notepad
File  Edit  Format  View  Help
#
# Copyright (C) 1999-2016 STROMASYS
# All rights reserved.
#
# The software contained on this media is proprietary to and embodies
# the confidential technology of STROMASYS. Possession, use, duplication,
# or dissemination of the software and media is authorized only pursuant
# to a valid written license from STROMASYS.
#
#=============================================================================
#
# Sample configuration file for MicroVAX 3600.
#
#-----------------------------------------------------------------------------

set session hw_model = MicroVAX_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 = MicroVAX_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).
#
#-----------------------------------------------------------------------------

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

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 Machines Manager level.
- The "configuration_name.icfg" file can be found in the home directiory 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.

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

Here is an example of a CHARON VM log file:

```
20180607:133937:INFO :0:00000249:ethane.cxx(9010): Logging started.
20180607:133937:INFO :0:000003A5:ethane.cxx( 131): session: loading built-in configuration "MicroVAX_3600"...
20180607:133937:INFO :0:000003A6:ethane.cxx( 163): session: ... done loading built-in configuration
"MicroVAX_3600"
20180607:133937:INFO :0:000003A7:ethane.cxx( 321): session: loading service configuration "My_MV3600"...
20180607:133937:INFO :0:000003A8:ethane.cxx( 356): session: ... done loading service configuration "My_MV3600"
20180607:133937: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.
20180607:133937:INFO :0:000003AB:ethane.cxx( 579): session: ... done loading configuration file
"configuration_name.icfg"
The previous message has been repeated 2 times.
20180607:133937:INFO :0:000003F2:sesmgr.cxx(1410): session: default log file size limit is 4194304 bytes
20180607:133937:INFO :0:0000032B:ethane.cxx(2694): Start request received.
20180607:133937:INFO :0:000003AC:ethane.cxx( 805): session: process affinity is 00000000000000FF, system
affinity is 00000000000000FF
20180607:133937:INFO :0:000003D1:ethane.cxx(1463): session: I/O domain affinity is 0000000000000003, CPU domain
affinity is 00000000000000FC
20180607:133937:INFO :0:0000024D:licenseman(1823): Checking the available license key "1422726238".
The previous message has been repeated 26 times.
20180607:133938:INFO :0:00000408:ethane.cxx(2867): CHARON-VAX (MicroVAX 3600), V 4.9 B 19402, May 10 2018 /
000.msc.test.center.nikolaev / 1918154109
20180607:133938: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
20180607:133938:INFO :0:00000097:ethane.cxx(2987): OS Environment: Microsoft Windows 10 Pro, 64-bit (Build
17134).
20180607:133938:INFO :0:00000098:ethane.cxx(2992): Host CPU: Intel(R) Xeon(R) CPU E31275 @ 3.40GHz x8.
20180607:133938:INFO :0:00000099:ethane.cxx(2997): Host Memory: 24320Mb
20180607:133938: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"
20180607:133938:INFO :0:00000418:busemul.cx( 188): SSC: MultiMedia Timer correction is enabled.
20180607:133939:INFO :0:00000001: tpool.cxx(1374): CPU: The ACE option is omitted; enable ACE as license
default.
20180607:133939:INFO :0:00000133: tpool.cxx(1606): Advanced CPU Emulation (ACE) enabled.
20180607:133939:INFO :0:0000032C:ethane.cxx(2730): "My_MV3600" started.
20180607:133939:INFO :0:00000419:winsckio.c(2703): OPA0: Connected. Remote 127.0.0.1:61468.
20180607:134413:INFO :0:0000032D:ethane.cxx(2776): "MicroVAX 3600" stop request received.
```

```
20180607:134413:INFO :0:0000024D:licenseman(1823): Licensing component received stop request.
20180607:134413:INFO :0:0000032E:ethane.cxx(2794): Stopped.
20180607:134413: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 the CHARON utilizes a so-called "rotating log" method. This means that a new default log file is always created each time the CHARON VM starts and if the size of the log file exceeds 64Kb (previous log files are kept).

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

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

ℹ️ 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.

ℹ️ 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 have to be configured. Their presence depends on the VAX/PDP11 model. It is always recommended to enable 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	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 Machines Manager "Show Console" option.It is also possible to start up some chosen application to connect to the CHARON VM console IP port.

Inner table for virtual_serial_line:

Parameter	Description
port	Specifies an IP port on the host to which the CHARON VM console will be mapped
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

Example:

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

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 Machines Manager.

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 pressing F6 button

It is recommended to set a hot key to stop the VM from the console (when the console is accessed remotely for example):

```
set OPA0 stop_on = F6
```

This line allows the CHARON VM to be stopped by pressing the "**F6**" key.

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"
    ```

    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= iqn.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\vms_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_bus=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	Mapping to a file representing VAX/PDP11 disk (disk image) on the host physical disk: • **"<file-name>.vdisk"** 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" > Using compressed folders to store virtual disks and tapes is not supported • **"\\.\PhysicalDrive<N>"** - 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. • **"\\.\PhysicalDrive(DevID =XXXX-XXXX-XXXX-XXXX-XXXX-XXXX-XXXX-XXXX)"** - 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). • **"\\.\PhysicalDrive(iScsiTarget = <iSCSI target>, LUN = <LUN number>)"** - mapping to iSCSI disks. iScsiTarget addresses the disk by its iSCSI target name. LUN specifies LUN on connected iSCSI disk.
physical_scsi_device	Mapping to a host SCSI device: • **"\\.\Tape<N>"** - mapping to a SCSI tape drive known to Windows
virtual_scsi_cdrom	Mapping to a host CD-ROM (not only SCSI) or to ISO image: • **"\\.\CdRom<N>"** - name of host CD-ROM drive • **"<file-name>.iso"** - name of ISO image. It may contain the full path, for example: "C:\My disks\vms_distributive.iso"
virtual_scsi_tape	Mapping to a file representing tape (tape image). It may contain a path, for example: "C:\My tapes\backup.vtape"
floppy_scsi_device	Mapping to a host floppy drive: • **"\\.\A:"** or **"\\.\B:"** - name of host CD-ROM drive. This parameter can be omitted.

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"

ℹ️ **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.

> 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	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.

<table>
<tr><td></td><td colspan="2">

Parameter	Description
port	Specifies an IP port on the host to which the CHARON VM serial line will be mapped
application	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

Example:

```
load virtual_serial_line/chserial TXA0 port=10011 application="putty.exe –load TXA0"
```

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".

> 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.

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

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	Example
affinity	This setting binds the running instance of the emulator CPUs to particular host CPUs.This should be used for soft partitioning host CPU resources or for isolating multiple CHARON instances on the same host from each other. By default the emulator instance allocates as many host CPUs as possible. 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.	`set session affinity = "0, 2, 4, 6"`
n_of_io_cpus	Reserves host CPUs (of those specified by "affinity" parameter, if any) for use by the emulator for I/O handling. By default the emulator instance reserves one third of available host CPUs for I/O processing (round down, at least one). The "n_of_io_cpus" parameter overrides the default by specifying the number of I/O host CPUs explicitly.	`set session n_of_io_cpus = 2`

Migration to CHARON-VAX and CHARON-PDP for Windows

Table of Contents

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 virtul device.

Creating CHARON Virtual Machine

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

Press the "Create VM from Template" button:
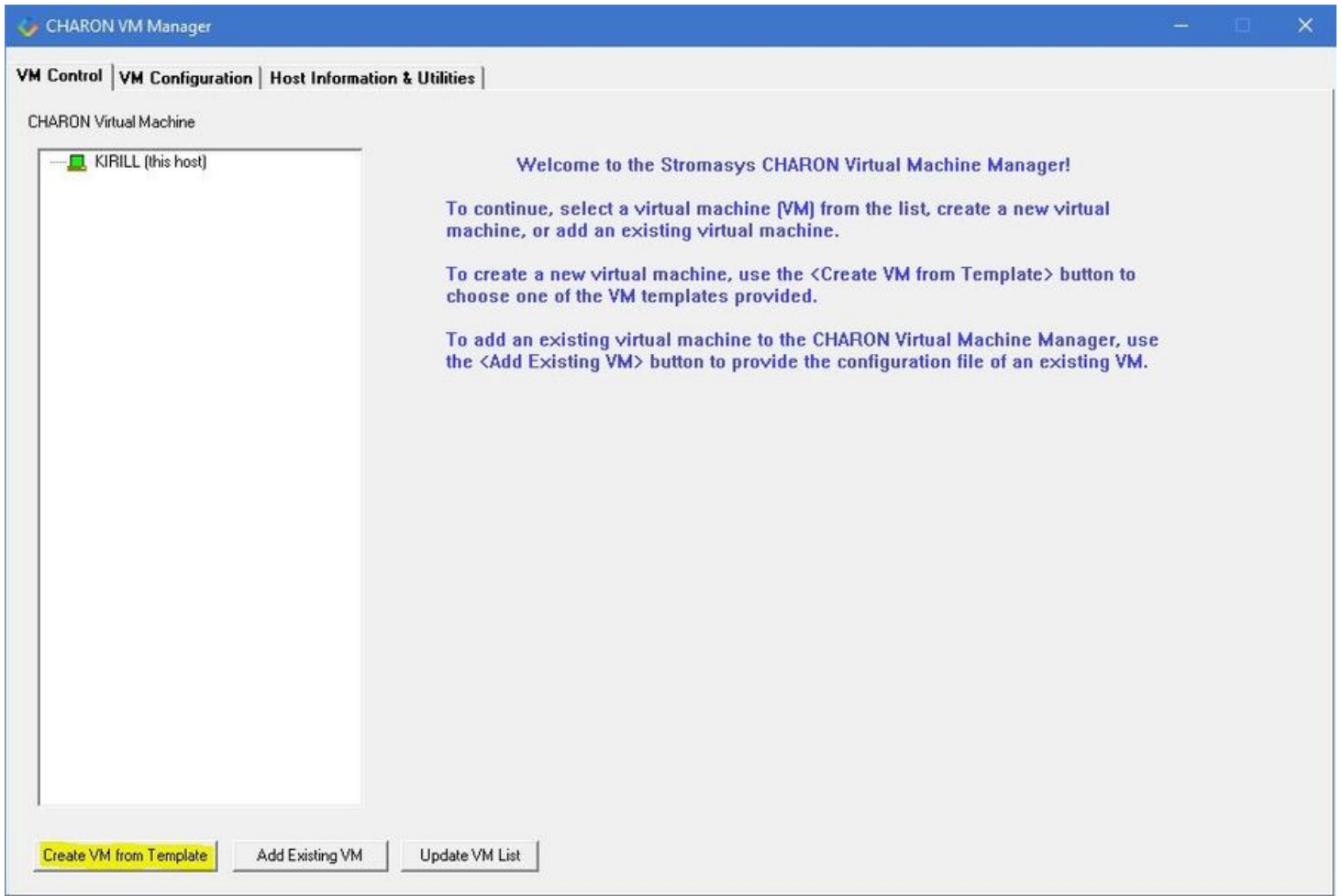
Specify the HP VAX model to be emulated and the name of the VM then press the "Create" button:
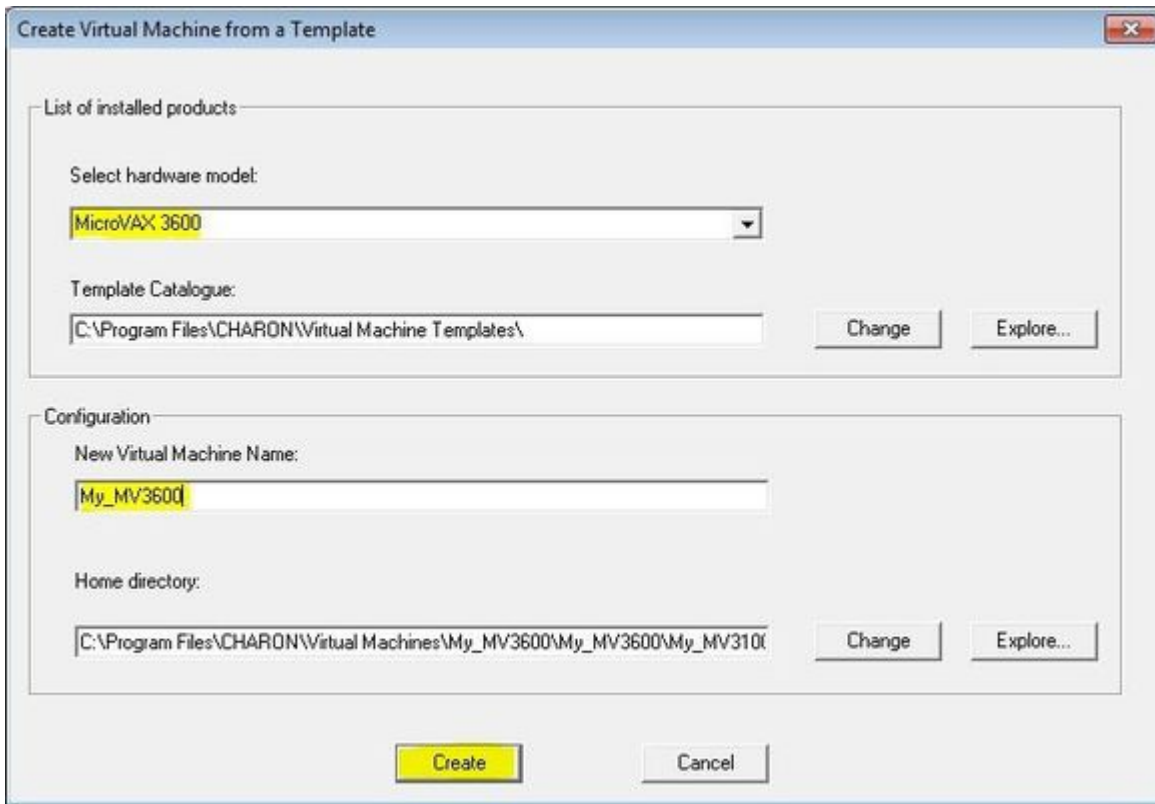
The VM will be created:
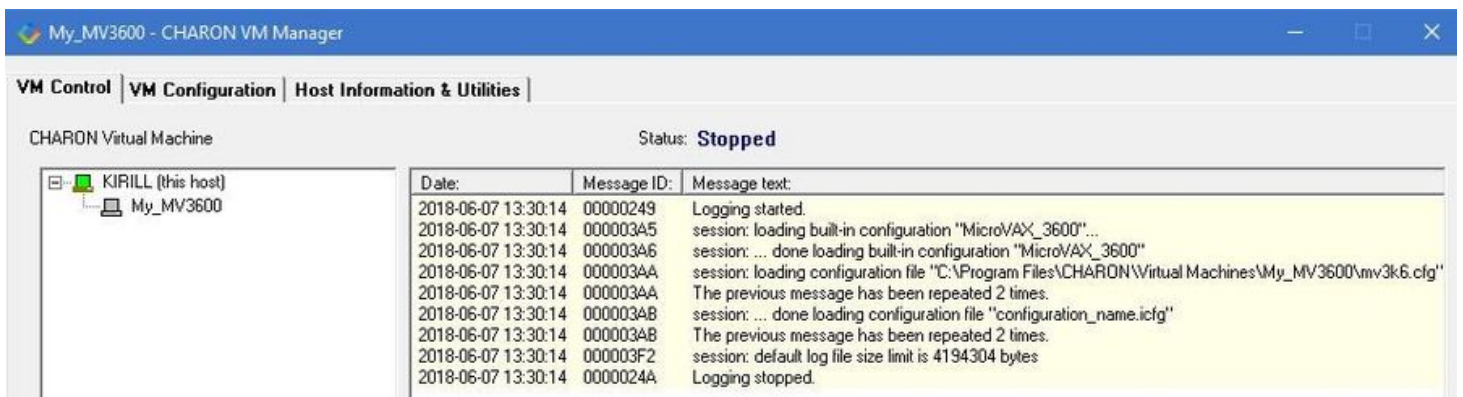
Select the VM from the left pane and press the "Edit Configuration" button:

The 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	
DESQA	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.

ℹ️ 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 Machines Manager:
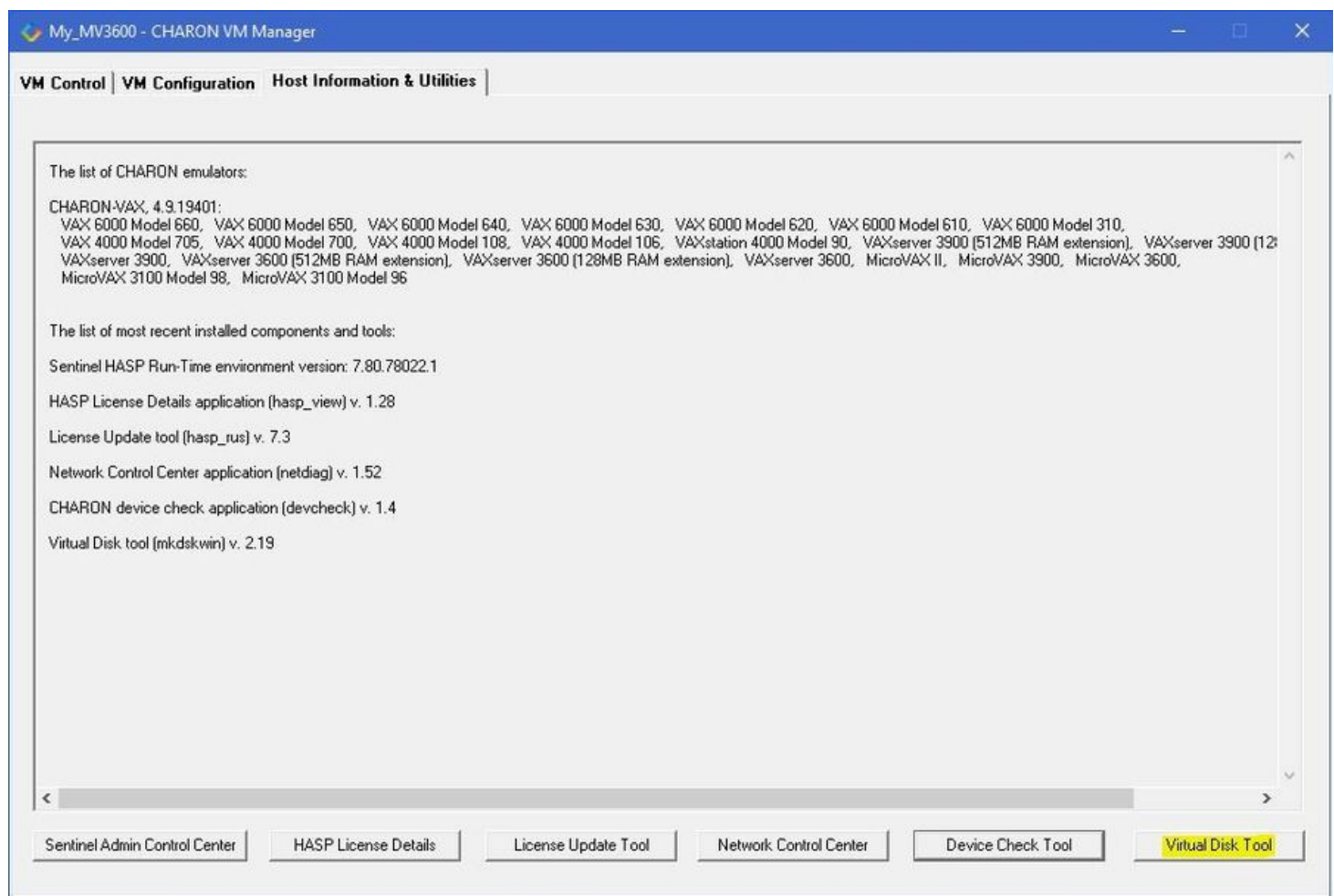
The following dialog will appear:
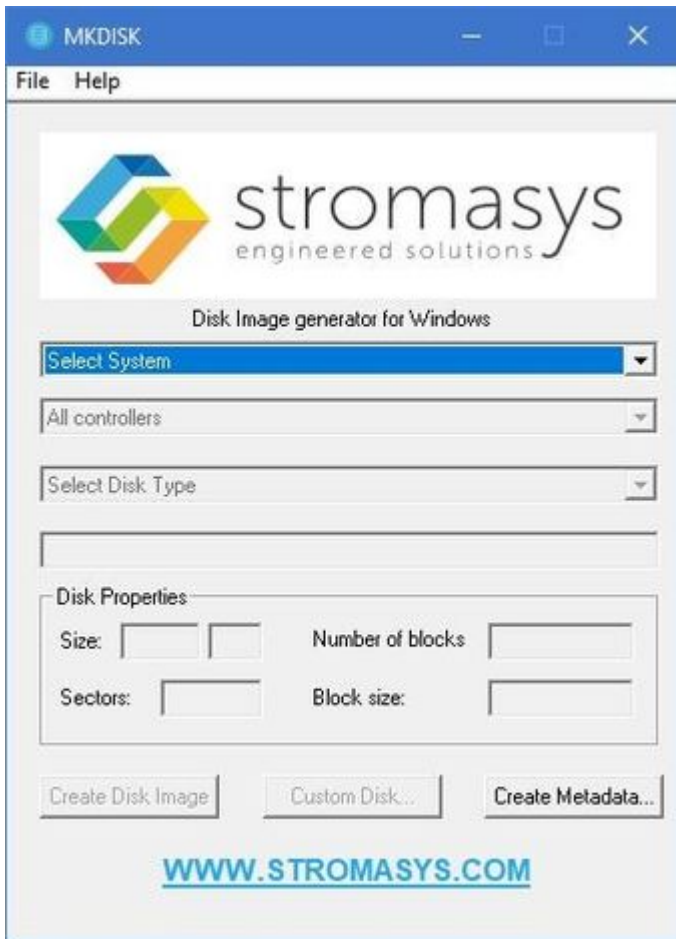
Select "VAX" in the "Select System" drop-down menu, "MSCP" in the "All Controllers" drop-down menu and "RZ24" (for example) in the "Select Disk Type" drop-down menu.

Press then the "Create Disk Image" button:
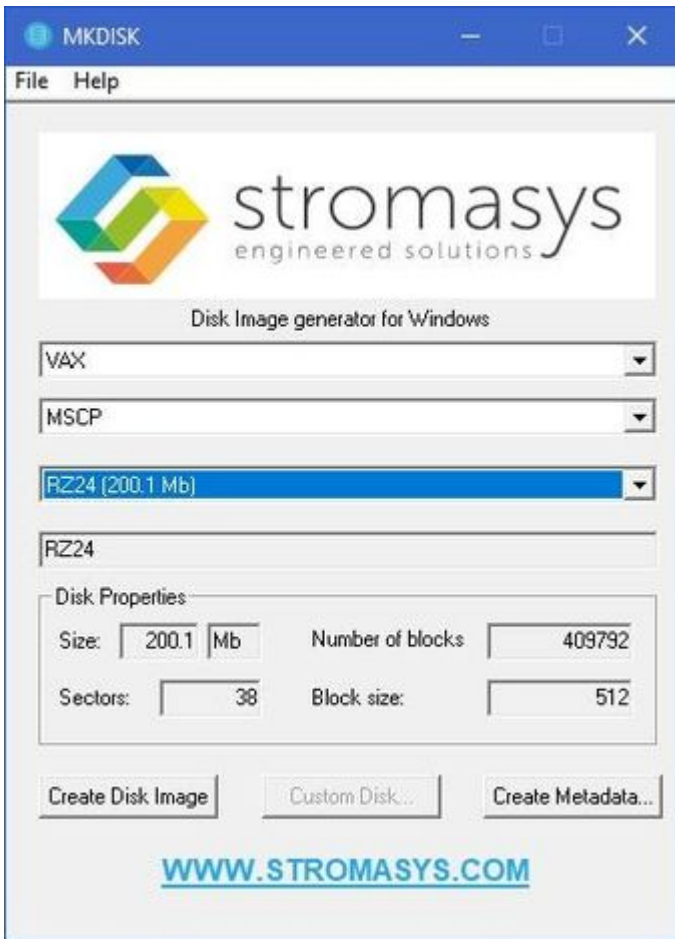
The utility will ask you to specify the output file name. Select the "C:\My disks" directory that has been created earlier, or create it directly in the dialog, and press the "Save" button:
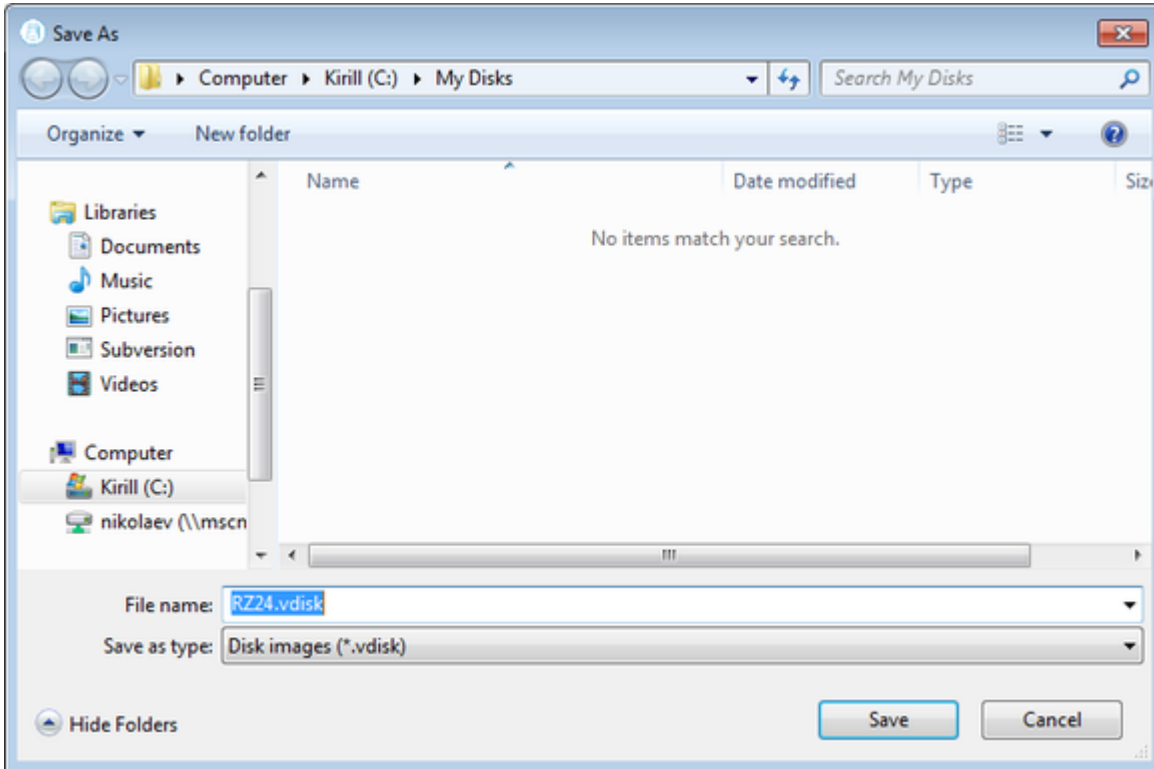
The "MkDisk" utility will create the requested disk image:
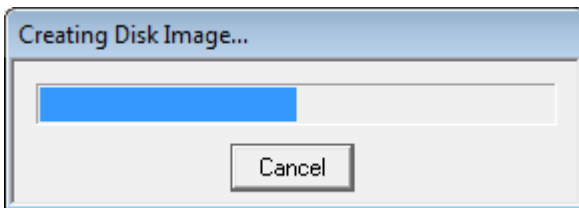
Repeat this sequence for the disks "RZ28.vdisk", "RZ22.vdisk" and "RZ23.vdisk". Place them in the same folder.

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.

ℹ️ 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/RSX11/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/RSX11/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

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

Configuration	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: ``` >>>sho dssi_id DSSI_ID Bus 0/A = 1 DSSI_ID Bus 1/B = 1 >>>set dssi_id 0 2 >>>set dssi_id 1 2 ```
Description	⚠ 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 HDS50 controller instance cannot use ID 0 and 1.

2. Configure preloaded SHAC adapters PAA

Configuration	``` set PAA port[<ID of the node to connect to>]=<port to receive connection from the node specified by ID> host[<ID of the node to connect to>]="<connected node host>:<connected node port>" ``` **Example**: ``` set PAA port[2]=11012 host[2]="pollux:11021" ``` 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.
Description	The second step is to interconnect SHAC adapters of different VMs. Use ID of the nodes as a reference for interconnections as it is shown in the example at left. Detailed meaning of the parameters is explained in "example_1" section below

3. Load HSD50 adapter

Configuration	``` load HSD50 <instance name> dssi_host=<SHAC instance> dssi_node_id=<HSD50 node ID> ``` **Example**: ``` load HSD50 DISKS dssi_host=PAA dssi_node_id=3 ``` In this example HSD50 controller instance "DISKS" is loaded, connected to "PAA" SHAC controller and assigned with ID 3.
Description	Load HSD50 instance, assign ID for it and connect it to "PAA" SHAC adapter. 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. ⚠ Note that <HSD50 node ID> 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 HDS50 instance cannot use 0 or 1, but it can be set to 3, for example.

4. Set SCS system ID and allocation class

Configuration	`set <HSD50 instance name> scs_system_id=<SCS ID> mscp_allocation_class=<allocation class>`
	
	Example:
	
	`set DISKS scs_system_id=3238746238 mscp_allocation_class=1`
	
	In this example HSD50 instance "DISKS" (configured at the previous step) is assigned with SCS ID "3238746238" and MSCP allocation class 1.

Description	Set SCS system ID and the MSCP allocation class. ⚠ These setting must be the same for all HSD50 controllers loaded! > Note that the MSCP allocation class must be different from the allocation class of the VAX node (excuding some special cases when they can be the same - please refer to OpenVMS Guides on Clusters configuration for more details). > > For nodes it is set using MODPARAMS.DAT and AUTOGEN > > Refer to OpenVMS User's Guides for details

5. Configure mapping to the system resources

Configuration	`set <HSD50 instance name> container[<unit number>] = <mapping>`
	
	Example:
	
	`set DISKS container[0]="\\DiskServer\Share\dua0-rz24-vms-v6.2.vdisk"`
	
	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.

Description	The final step is specifying HSD50 mapping to the system resources. 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<N>"). See the sections below for different examples of the mapping.

> 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 differents ports of "PAA" the HDS50 instance cannot use 0 or 1. But it can use 3 for example.
scs_system_id	A string value that specifies the *SCSNODENAME* 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 *ALLOCLASS* 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 abobe.

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

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 *ALLOCLASS* 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

Table of Contents

General description

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

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

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

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

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

> The CHARON-VAX / CHARON-PDP software license is not distributed for Proof-of-Concept and evaluation installations. Only hardware dongles are used in these cases.
>
> It is important to connect the HASP license keys to the computer powered on even if the CHARON VM is not running because the keys contain a built-in battery that needs to be charged. If the battery is completely discharged, the license key can be irreparably damaged.

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

The following sections list all the main parameters of the CHARON-VAX / CHARON-PDP licensing mechanism.

Parameters defined by CHARON VM license

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

General	Products relevant	Optional
- Physical key ID - License Number - End user name - Master key ID - License release date and time - Update Number - Purchasing Company name. In most cases the company to which the key was issued originally	- Commercial product name - Commercial product code - Commercial product version and range of build numbers suitable for running - Range of CHARON VM virtual models available for running - Type of host CPU required - Host operating system required - Number of virtual CPUs enabled for virtual SMP systems - Minimum number of host CPU cores required - Minimum host memory required - Maximum memory emulated. If not present the value defaults to the maximum memory possible for the particular virtual system. Note that the maximum memory may not be available to the virtual system if the host computer has insufficient physical memory. - Maximum number of CHARON VM instances that can be run concurrently - Whether or not CHAPI (CHARON API) can be used with this product - Product and Field Test expiration dates (if any) - Product and Field Test executions counter (if any) - Maximum number of hosts that may run CHARON VM concurrently (in the case of a networking license) - Level of support (if any), end date of any support contract, the "First Line" Service Provider - Frequency of CHARON VM license checking during CHARON VM execution	- Possibility to attach hardware QBUS/UNIBUS hardware via adapter - Parameter that reduces the maximum speed of the program - Parameter that enables the product to support additional serial lines through an option board from a company such as DIGI - Parameter that prohibits use of Advanced CPU Emulation. If not present the Advanced CPU Emulation is enabled - Parameter that enables the emulation of the IEQ11-A IEEE488 Controller (on top of the DCI-3100 IEEE488 Controller) - Parameter that enables the emulation of the DRV11-WA I/O controller (on top of the DCI-1100 I/O controller)

CHARON-VAX / CHARON-PDP licensing models

CHARON-VAX / CHARON-PDP licensing models are divided into 3 groups:

Regular Sentinel HASP keys

This is the most common way of CHARON-VAX / CHARON-PDP licensing.

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

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

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

Network Sentinel HASP keys

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

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

The Sentinel run-time driver is distributed as a part of the CHARON-VAX / CHARON-PDP kit. Please see the "License installation" section of this chapter for details.

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

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.

Software licenses

The CHARON-VAX / CHARON-PDP Software License (SL) is a "virtual" key with exactly the same functionality as the hardware dongle. It does not require any hardware but the installation of the Sentinel run-time environment is required.

The Software Licenses (SL) are best suited for stable environments because their correct function depends on certain characteristics of the host system. Changing any of these characteristics will invalidate the license.

- If the CHARON host runs on real hardware, the software licenses are by default **tightly bound to the hardware** for which they were issued. If major hardware characteristics of the system are changed, the license will be disabled.
- If the CHARON host runs in a **virtual environment** (e.g. VMware), 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 always network-wide on Windows, so they behave the same way as Network HASP keys.

Multiple licenses configuration and backup license

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

The "HASP License details" utility (it can be started from CHARON Virtual Machines Manager) displays a range of available licenses but note that, by default, a CHARON VM may use any of them as the "active" one - unless it is directly specified by the "license_key_id" parameter (see below). The utility provides the license numbers and ID / IP addresses of the hosts where the licenses are installed.

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

When needed, it is possible to use a special parameter in the CHARON VM configuration files to specify exactly which license must be used by each particular CHARON VM:

Parameter	license_key_id
Type	Text string
Value	A set of Sentinel Key IDs that specifies the license keys to be used by CHARON. It is also possibly to use a keyword "**any**" to force CHARON to look for a suitable license in all available keys if the license is not found in the specified keys. **Example**: `set session license_key_id = "1877752571,354850588,any"` Based on the presence of this parameter in the configuration file, CHARON behaves as follows: 1. **No keys are specified (the parameter is absent)** CHARON performs an unqualified search for any suitable key in unspecified order. If no key is found, CHARON exits. 2. **One or many keys are specified** 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). 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. 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.

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

License installation

Installation from scratch

> **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-VAX / CHARON-PDP regular or network license consists of:

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

1. Login as Administrator
2. Open the CHARON-VAX/PDP 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 x64 host) or "`C:\Program Files\Common Files\Aladdin Shared\HASP`" (in case of x86 host)

> Note that the following operations have to be performed on installation phase for network licenses:
>
> - *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 by CHARON-VAX / CHARON-PDP).

<div style="background-color:#4CAF50; color:white; padding:8px;">

Replacement of currently installed Sentinel run-time

</div>

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 other 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 x64 host) or "`C:\Program Files\Common Files\Aladdin Shared\HASP`" (in case of x86 host)

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

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

The CHARON-VAX / CHARON-PDP software licenses can be installed / updated according to the procedure described below:

- Install CHARON-VAX / CHARON-PDP together with Sentinel run-time (Sentinel run-time is an essential part of CHARON-VAX / CHARON-PDP for Windows distribution)
- Reboot the host system
- Connect the HASP dongle to the host system (in case of update of a license located on a dongle)
- Collect the CHARON-VAX / CHARON-PDP host fingerprint file ("*.c2v")

Open the CHARON Virtual Machines Manager, switch to the "Host Information & Utilities" tab and press the "License Update tool" button to envoke the "License Update Service" utility:
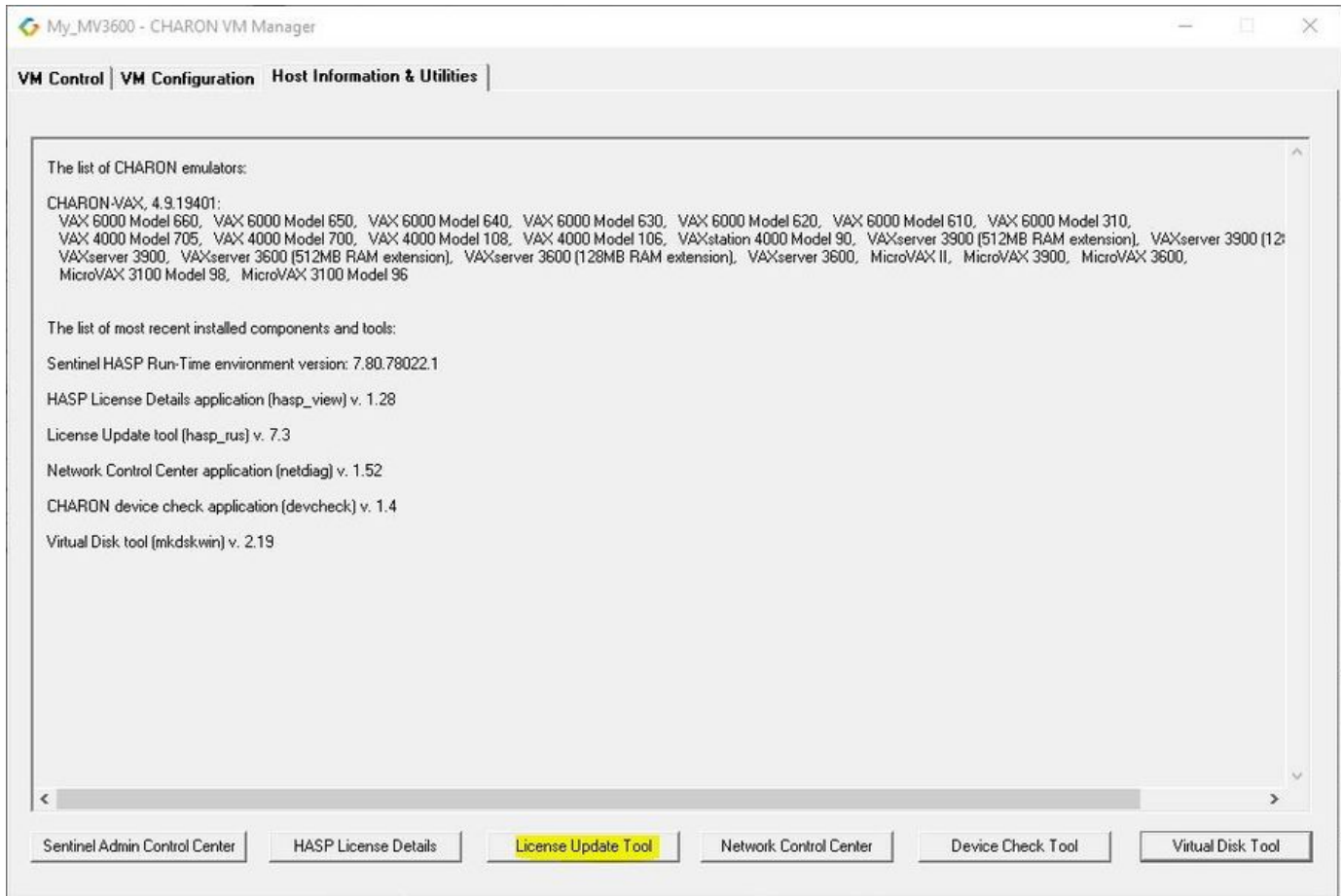
- In the utility dialog popup switch to the "Collect Status Information" tab (opened by default) then select:
  - "Installation of new protection key" in case of Software License if no Software License has been already installed on the host.

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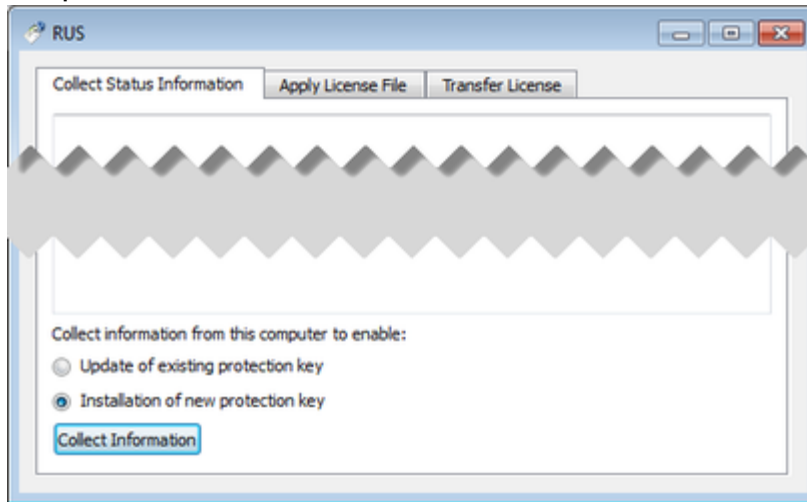
- "Update of existing protection key" in case of HL/HASP dongle or Software License that has been already installed on the host and needs updating.

Example:
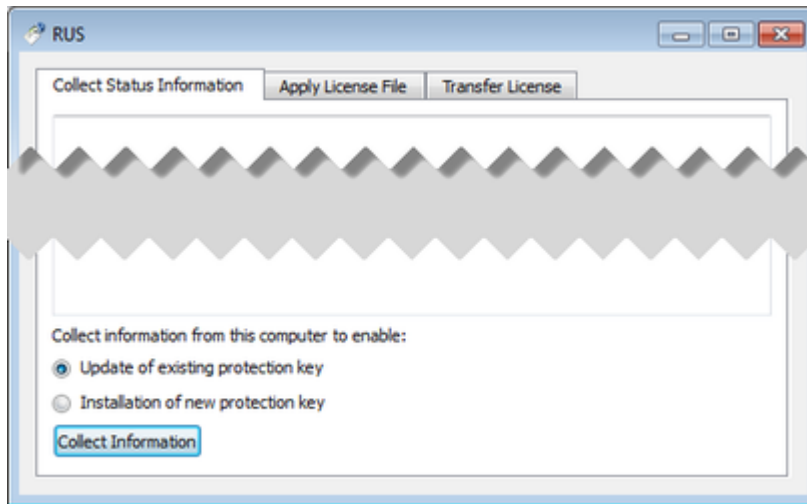
> When using HASP dongles, select only the "Update of existing protection key" option.
>
> In case of Software License use the "Installation of new protection key" option if the host does not have any Software License installed and the "Update of existing protection key" option if an already installed Software License has to be updated.

- Press the "Collect Information" button. In the popup dialog choose the place to store the "Fingerprint.c2v" file and press the "Save" button:
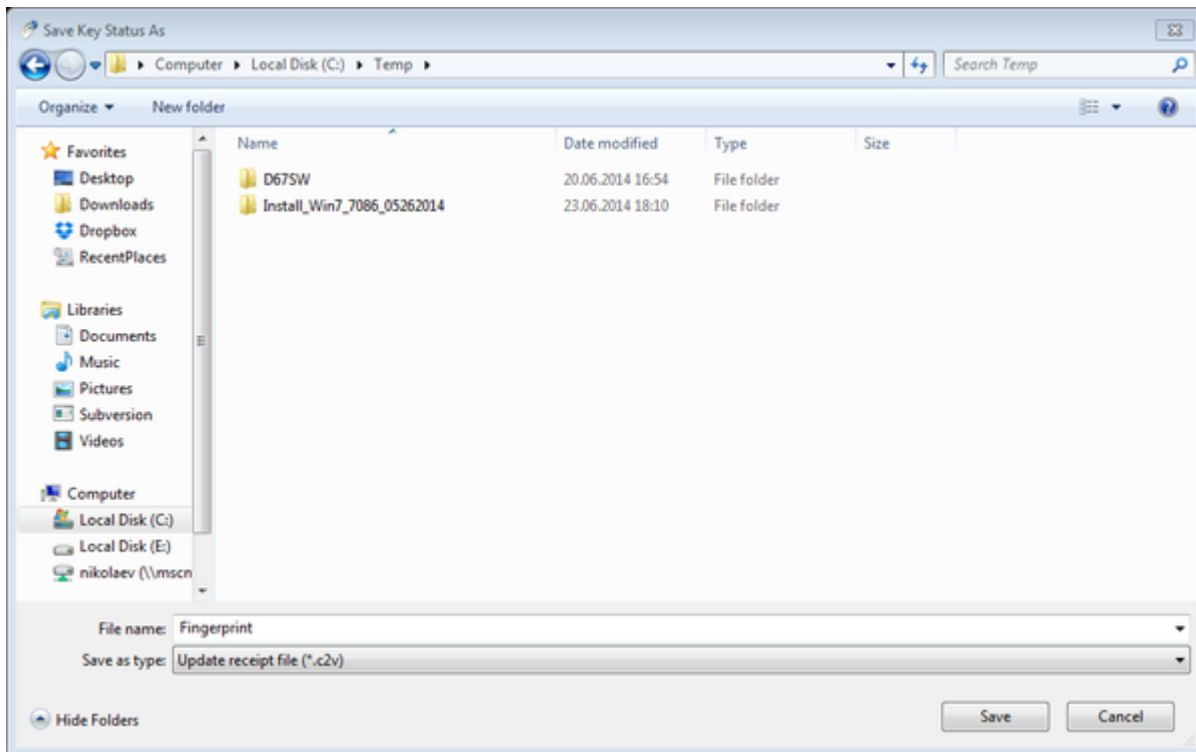
- A message should appear similar to this example, confirming the fingerprint has been collected successfully.
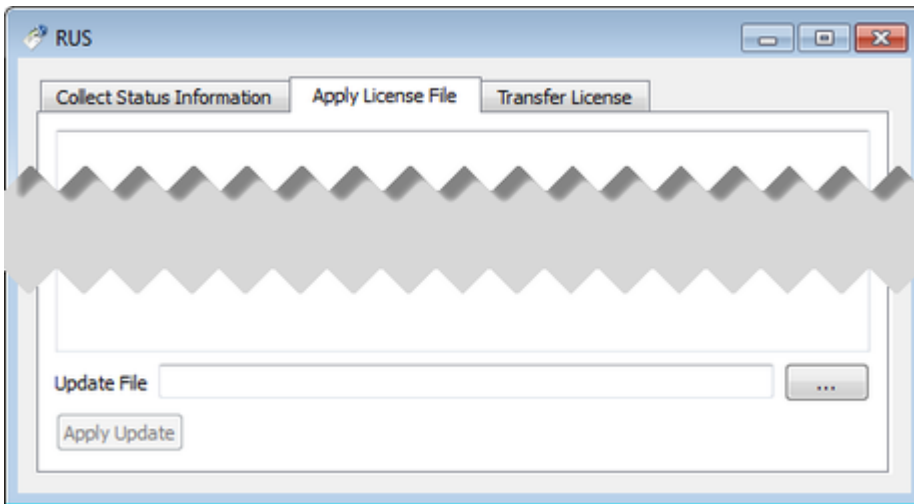
- Send the ".c2v" file ("*Fingerprint.c2v*" in the example above) to STROMASYS

- STOMASYS will send you a ".v2c" file in return. Put it somewhere on the CHARON-VAX / CHARON-PDP host.

- Open up the "License Update Service" utility the way described above and open the "Apply License File" tab:
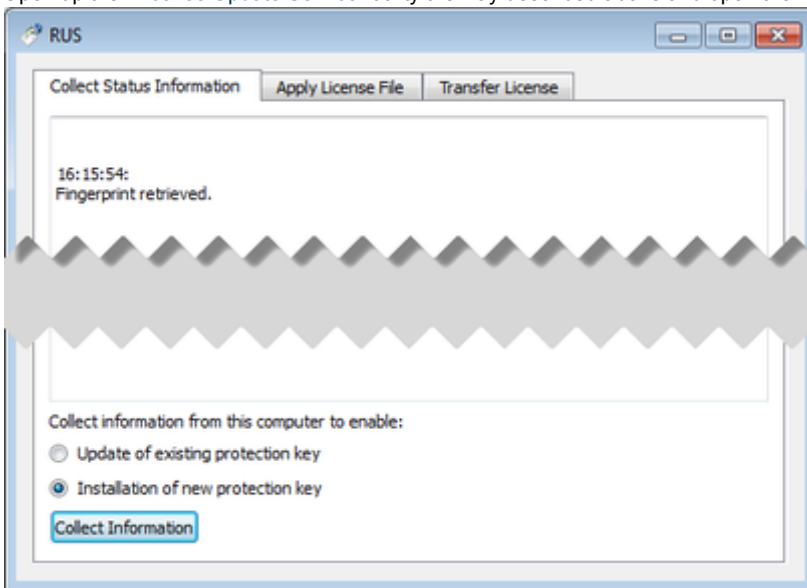
Press the "..." button beside the "Update File" edit-box. In the popup select the license file received from STROMASYS:
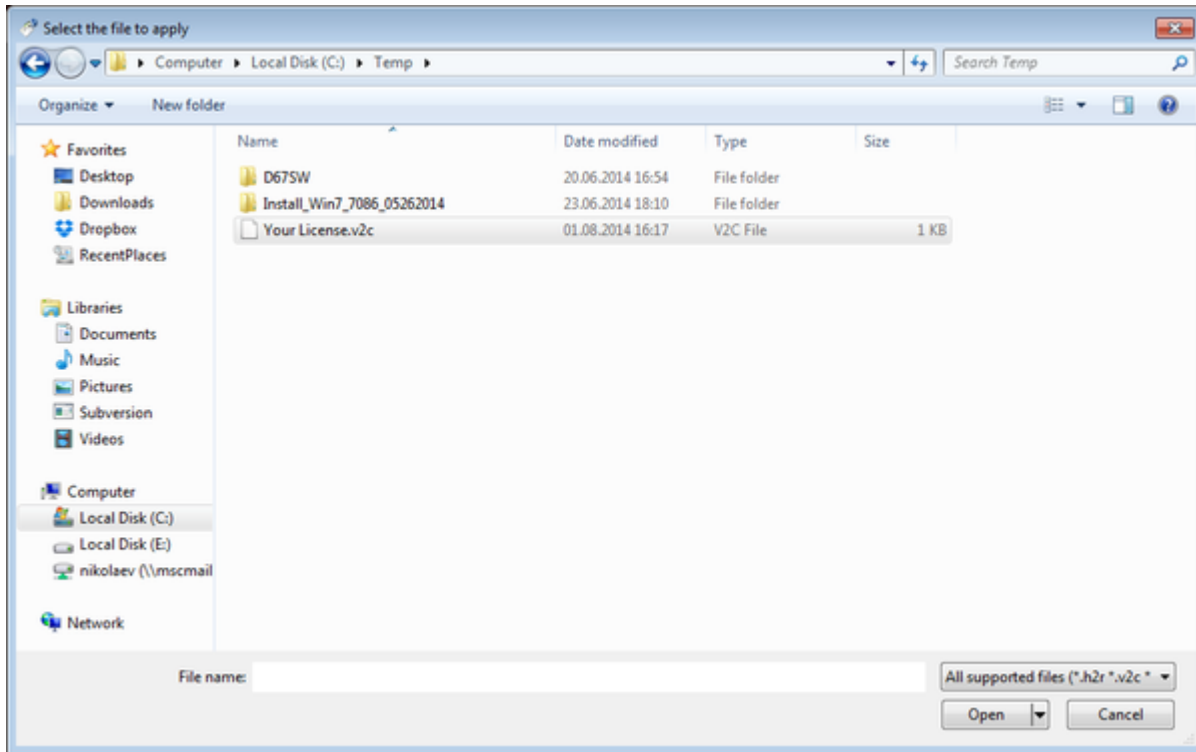
Press the "Open" button and apply the license.

- Open the CHARON Virtual Machines Manager, switch to the "Host Information & Utilities" tab and press the "Sentinel Admin Control Center" button:
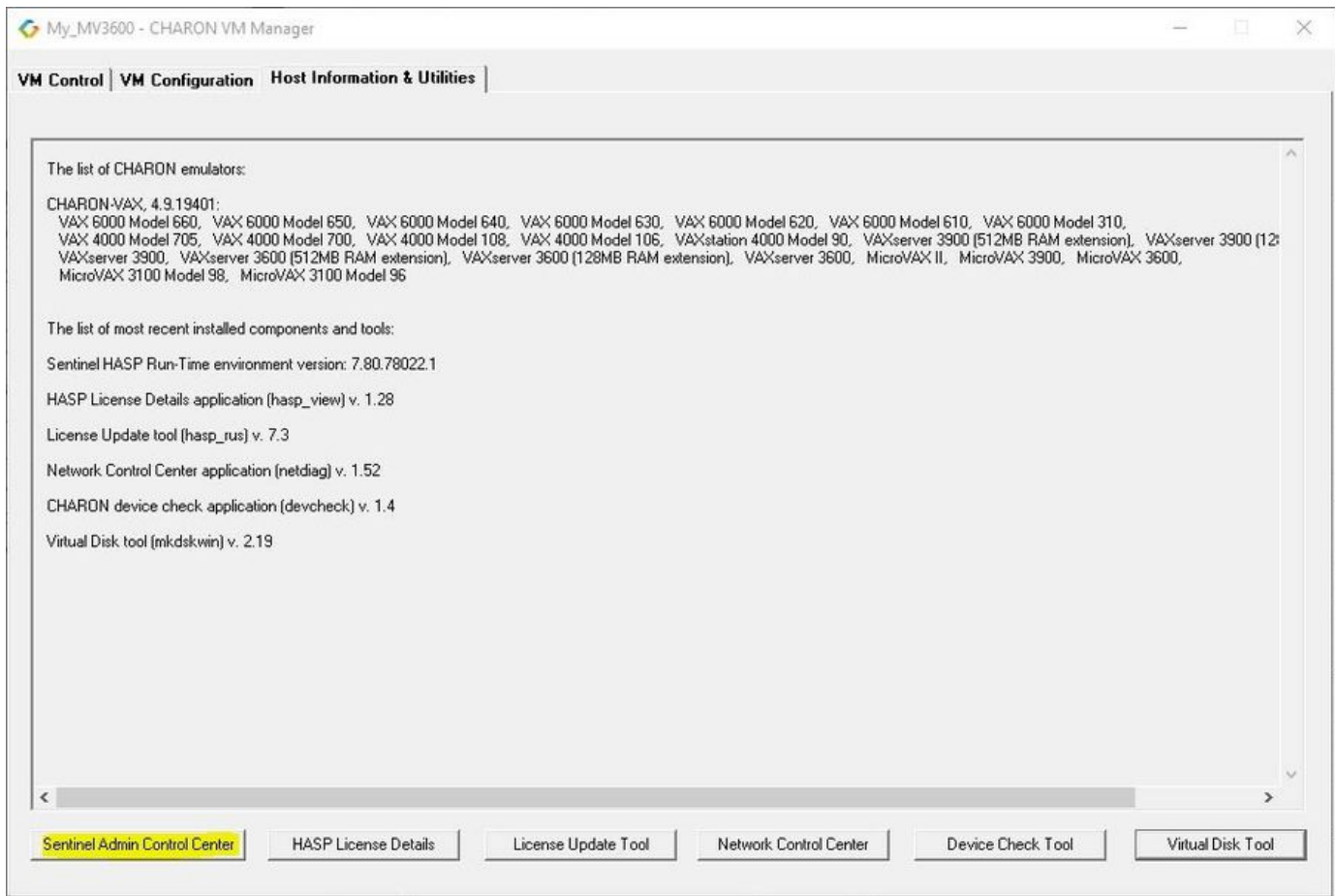
or - alternatively - start any web browser on this system and go to the http://localhost:1947 page to access the "Sentinel HASP Admin Control Center" (ACC).

- Ensure that the license appears in the "Sentinel Keys" menu.

As the content of the installed software or HL/HASP license is not shown by the Sentinel HASP Admin Control Center, press the "HASP License Details" button in the "Host Information & Utilities" section of the CHARON Virtual Machines Manager (see above) to display it:
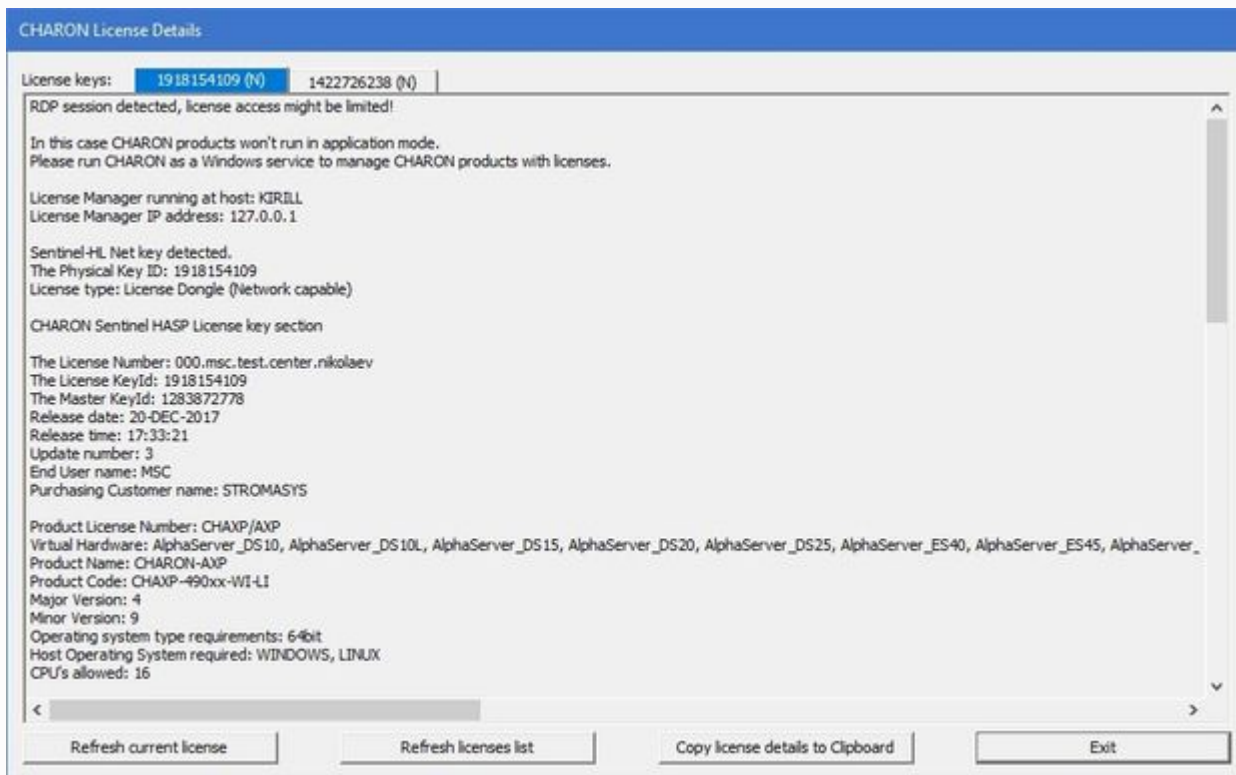
Note that the following operations have to be performed on installation phase for network-wide software licenses:

- *On the server side (where 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 by CHARON-VAX / CHARON-PDP).

License management

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

Sentinel Admin Control Center

General Description

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

> Sentinel Admin Control Center is not able to display CHARON-VAX / CHARON-PDP licenses. To do this operation, open the CHARON Virtual Machines Manager, switch to the "Host Information & Utilities" tab and press the "HASP License Details" button (see below).

To access the Sentinel Admin Control Center, open the tCHARON Virtual Machines Manager, switch to the "Host Information & Utilities" tab and press the "Sentinel Admin Control Center" button:

or - alternatively - start any web browser on this system and go to the http://localhost:1947 page.

Example:

This example demonstrates that 4 license keys are available:
1. A network key ("*HASP-HL NetTime*") on the host "XEON4WAYW7"
2. A network key installed locally
3. An HASP-HL installed locally
4. A network-wide software license on the host "RH64"

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

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

Disable remote keys access

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

To disable access to remote keys, switch to the "Access to Remote License Managers" tab, uncheck the "Allow Access to Remote Licenses" check box then press the "Submit" button to apply this change:

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

License management utility

CHARON-VAX / CHARON-PDP for Windows provides two specific utilities for license management:

- "HASP View" - This utility is used to display the license(s) content.
- "License Update Service" - This utility is used to collect key status information and host fingerprint (C2V) files and to apply updates (".v2c" files).

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

Removing CHARON-VAX / CHARON-PDP software licenses

The following procedure must be applied to remove the software license (SL):

1. Open the CHARON Virtual Machines Manager, switch to the "Host Information & Utilities" tab and press the "Sentinel Admin Control Center" button:
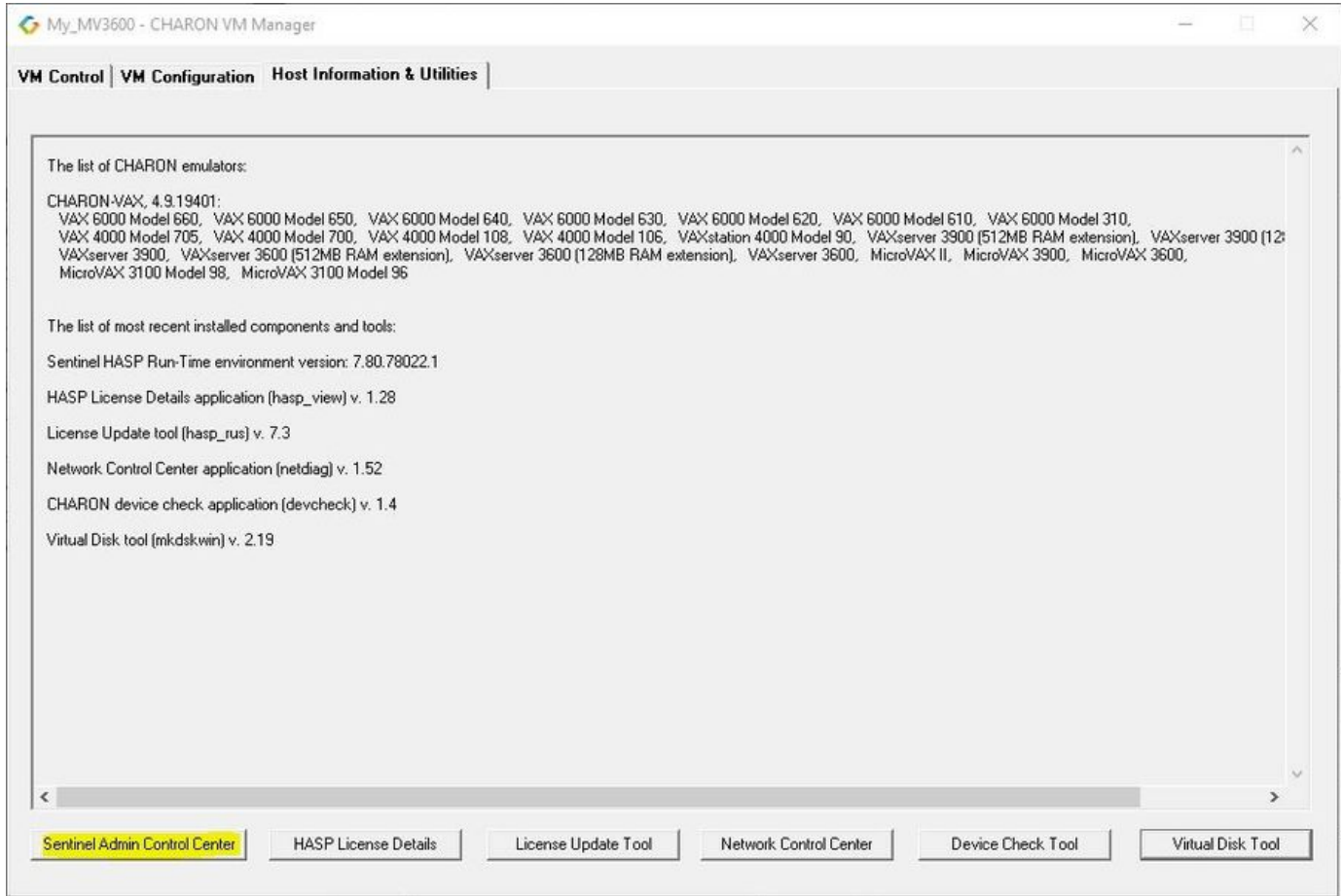
or - alternatively - start any web browser on this system and go to the http://localhost:1947 page.

2. In the "Sentinel HASP Admin Control Center" (ACC), select the "Sentinel Keys" option in the left pane and locate the target "Sentinel SL AdminMode" license.
3. Press the "Certificates" button at the right side of the SL description:
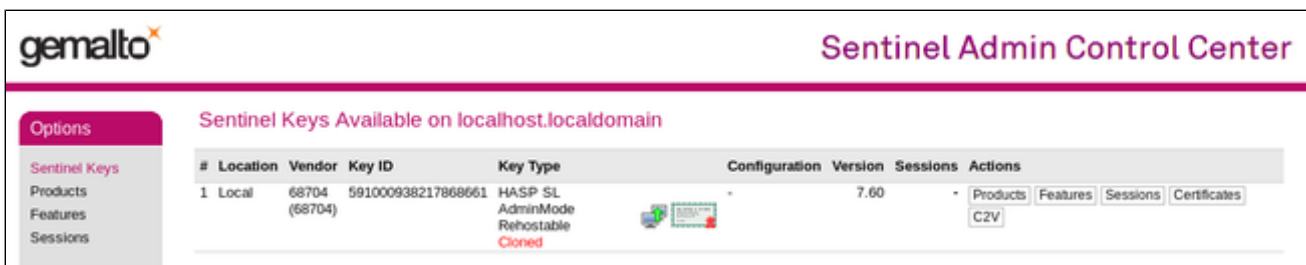
4. Note the name of the correspondent certificate in the "Certificates" section. The name is in the form of `<KeyID>_base.v2c`
5. Remove the corresponding certificate file in the "`C:\Program Files\Common Files\SafeNet Sentinel\Sentinel LDK\installed\68704`" folder.
6. Reboot the CHARON host.
7. Start the "Sentinel HASP Admin Control Center" (ACC) again to ensure the SL has been removed.

License Deinstallation

To completely remove a CHARON-VAX / CHARON-PDP license from a host, remove the Sentinel run-time driver using the following procedure:

- Login as Administrator
- Open the CHARON CHARON-VAX / CHARON-PDP kit folder
- Switch to the "hasp_install" subfolder
- Unzip the archive located in this folder
- Open "cmd.exe" and switch to the folder where the files were unzipped
- Execute the following command:

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

Unplug the license dongle. Note that the Sentinel run-time driver is uninstalled automatically when a complete uninstallation of CHARON-VAX / CHARON-PDP is performed.

Special "backup" license keys

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

- The run time is typically limited to 720 hours (30 days). This is the time alloted to get a replacement dongle from STROMASYS.
- A backup license may be valid only until a certain date.

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

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

Open the CHARON Virtual Machines Manager, switch to the "Host Information & Utilities" tab and press the "Device Check Tool" button:
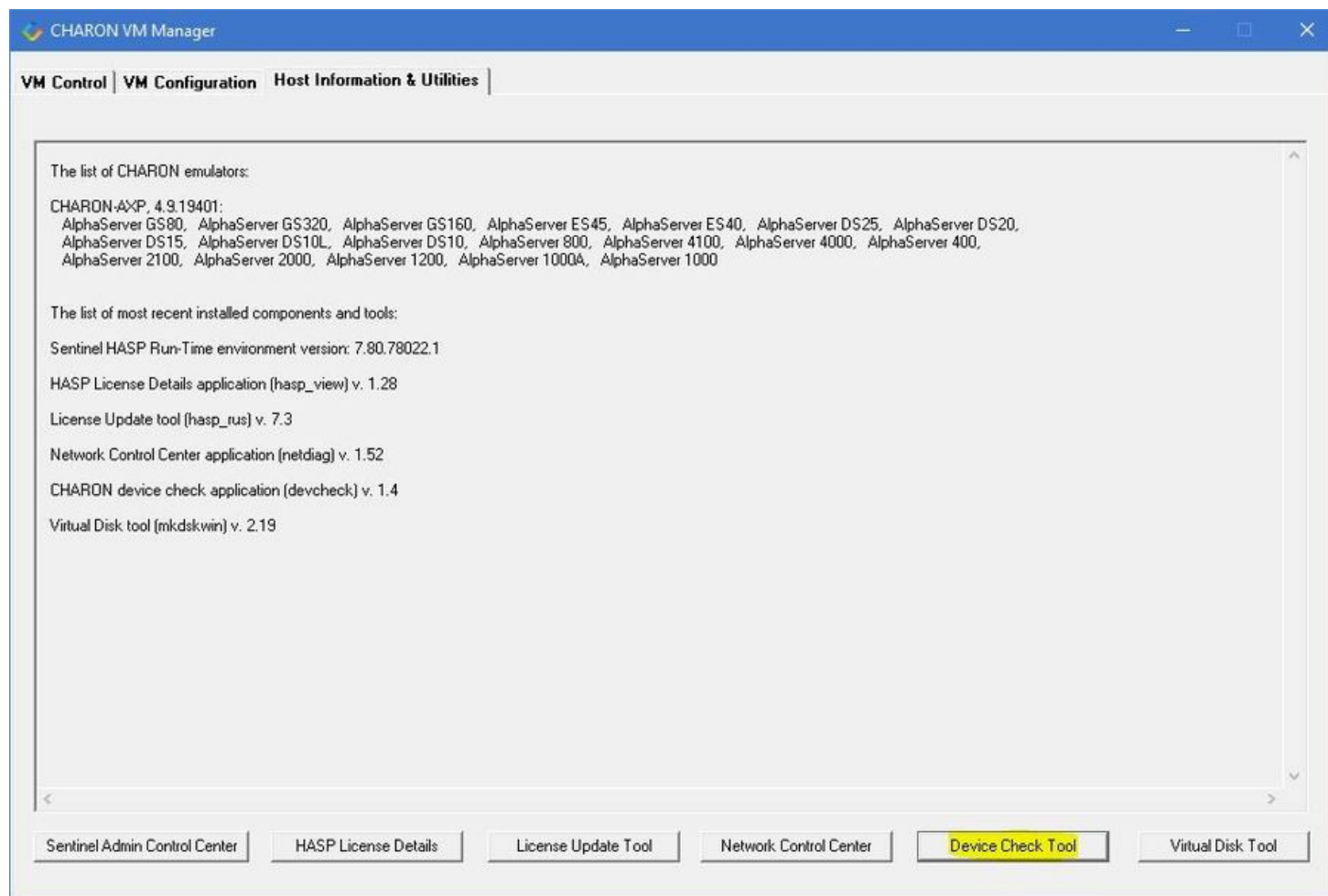
### Reviewing available physical disks

By default the "Host Device Check" utility reports the available physical disks:
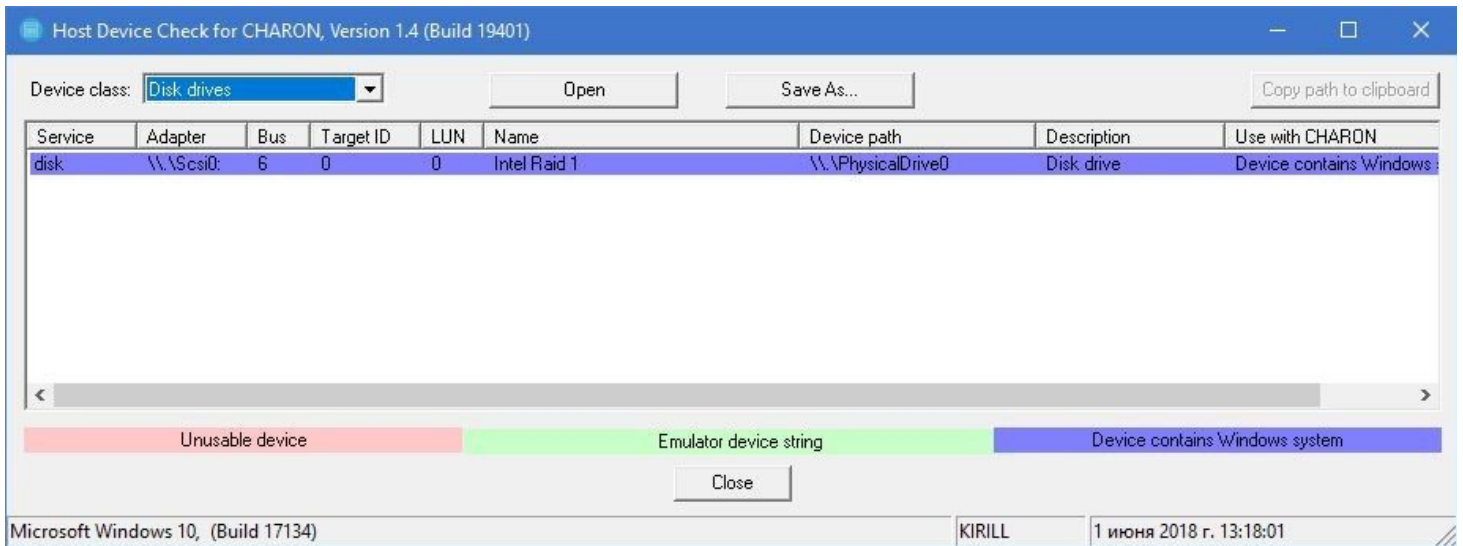
⚠ Please note that the unusable devices and the devices containing Windows system (see corresponding colors above) **must not** be mapped to CHARON!

Reviewing all the available host resources

Select "All drives" in the "Device class" drop-down list:
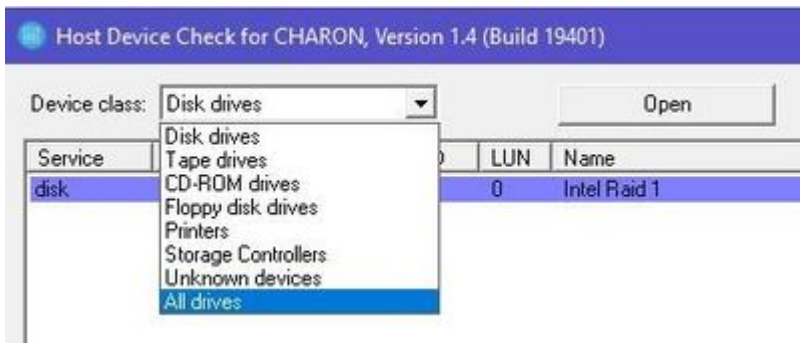
The "Host Device Check" utility will display all the host resources:
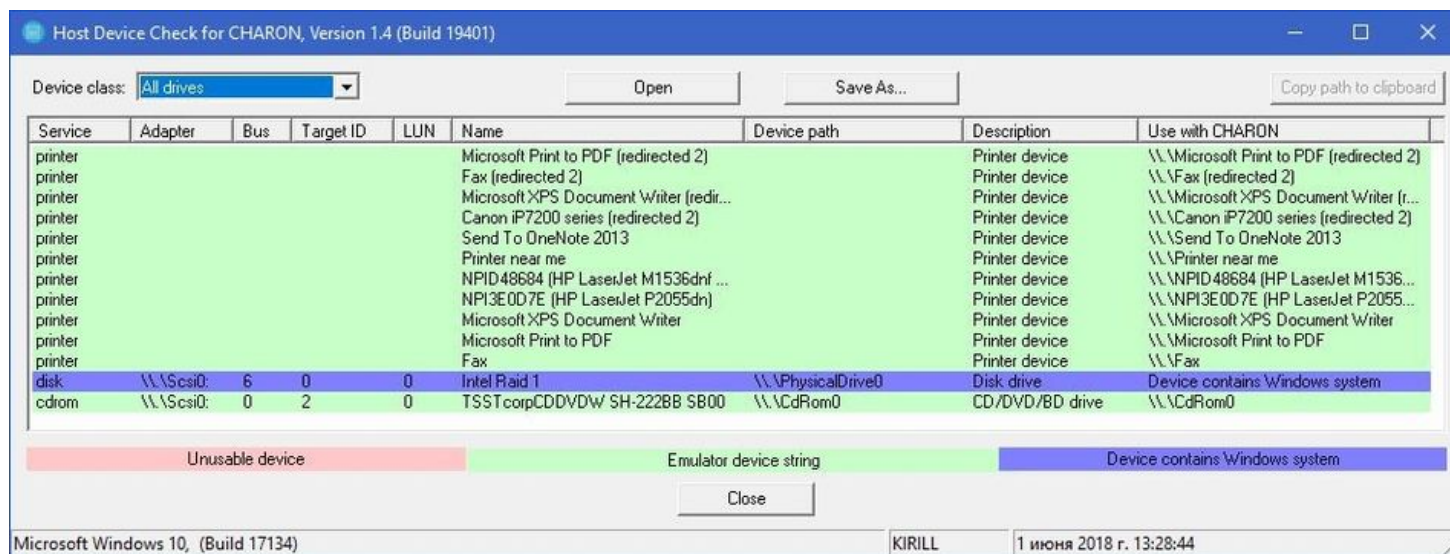
It is also possible to choose all the other categories to narrow the possible mapping options list.

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:
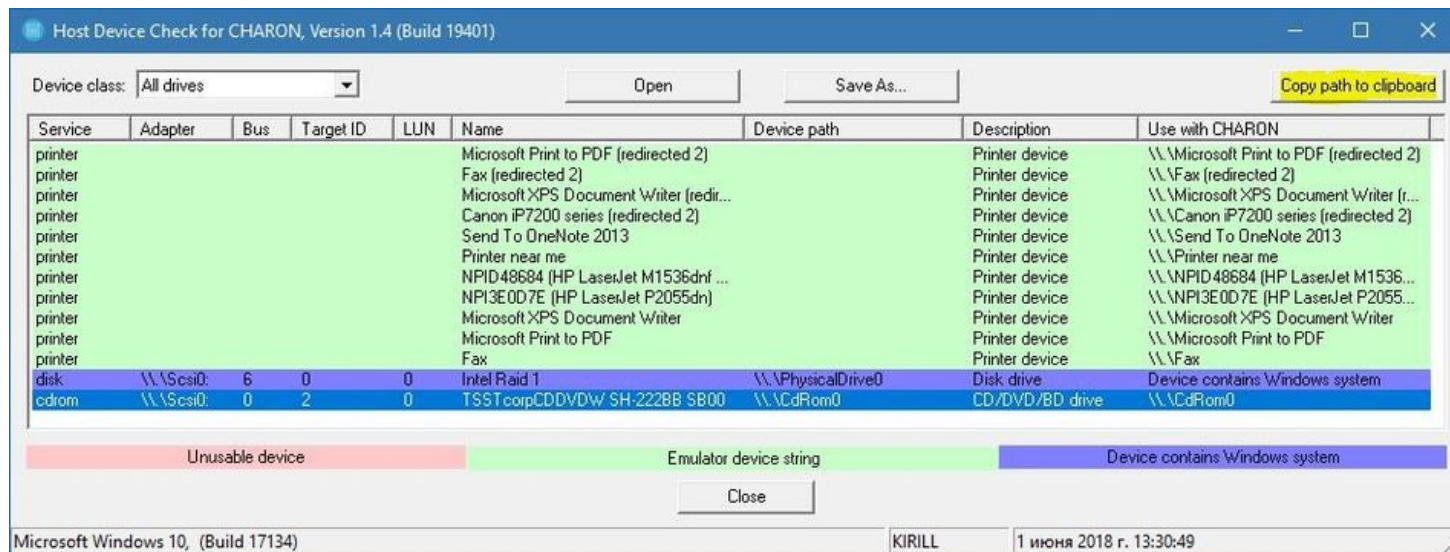
The selected configuration string will be copied to the clipboard, it can then be pasted to the CHARON configuration file (using the "CTRL-V" keys combination for example).

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

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 Machines Manager, switch to the "Host Information & Utilities" tab and press the "Network Control Center" button:

CHARON VM Manager

VM Control | VM Configuration | **Host Information & Utilities**

The list of CHARON emulators:

CHARON-VAX, 4.9.19401:
VAX 6000 Model 660, VAX 6000 Model 650, VAX 6000 Model 640, VAX 6000 Model 630, VAX 6000 Model 620, VAX 6000 Model 610, VAX 6000 Model 310,
VAX 4000 Model 705, VAX 4000 Model 700, VAX 4000 Model 108, VAX 4000 Model 106, VAXstation 4000 Model 90, VAXserver 3900 (512MB RAM extension), VAXserver 3900 (12...
VAXserver 3900, VAXserver 3600 (512MB RAM extension), VAXserver 3600 (128MB RAM extension), VAXserver 3600, MicroVAX II, MicroVAX 3900, MicroVAX 3600,
MicroVAX 3100 Model 98, MicroVAX 3100 Model 96

The list of most recent installed components and tools:

Sentinel HASP Run-Time environment version: 7.80.78022.1

HASP License Details application (hasp_view) v. 1.28

License Update tool (hasp_rus) v. 7.3

Network Control Center application (netdiag) v. 1.52

CHARON device check application (devcheck) v. 1.4

Virtual Disk tool (mkdskwin) v. 2.19

Sentinel Admin Control Center | HASP License Details | License Update Tool | Network Control Center | Device Check Tool | Virtual Disk Tool

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:

CHARON Network Control Center

CHARON Network Control Center.
Introduction page.

The CHARON Network Control Center designed to manage CHARON networking including:

- Installation/de-installation of the CHARON network drivers;
- Configuring physical adapters to be used by the CHARON/Host system;
- Troubleshooting the physical adapter setup;
- Checking the connection reliability.
- Checking that adapter supports MAC address change.
- Checking for the MAC address duplication on the LAN;
- Checking for the IP address duplication on the LAN;
- Suggesting the configuration file settings;
- Disable TCP chimney offload for CHARON emulators;
- Monitoring NIC activity
    (NDIS driver v. 5.3.0 or later should be installed
    and CHARON running to enable Monitor feature);

Select the desired function and click Next button for continue.

Please select the action to continue:

⦿ Install/Upgrade
○ Configure NIC
○ Troubleshoot NIC dedicated for CHARON
○ Disable TCP chimney offload for CHARON
○ Monitor CHARON NIC activity

< Back   Next >   Cancel   Help

Select "Install/Upgrade CHARON NDIS driver" checkbox and press the "Select" button:
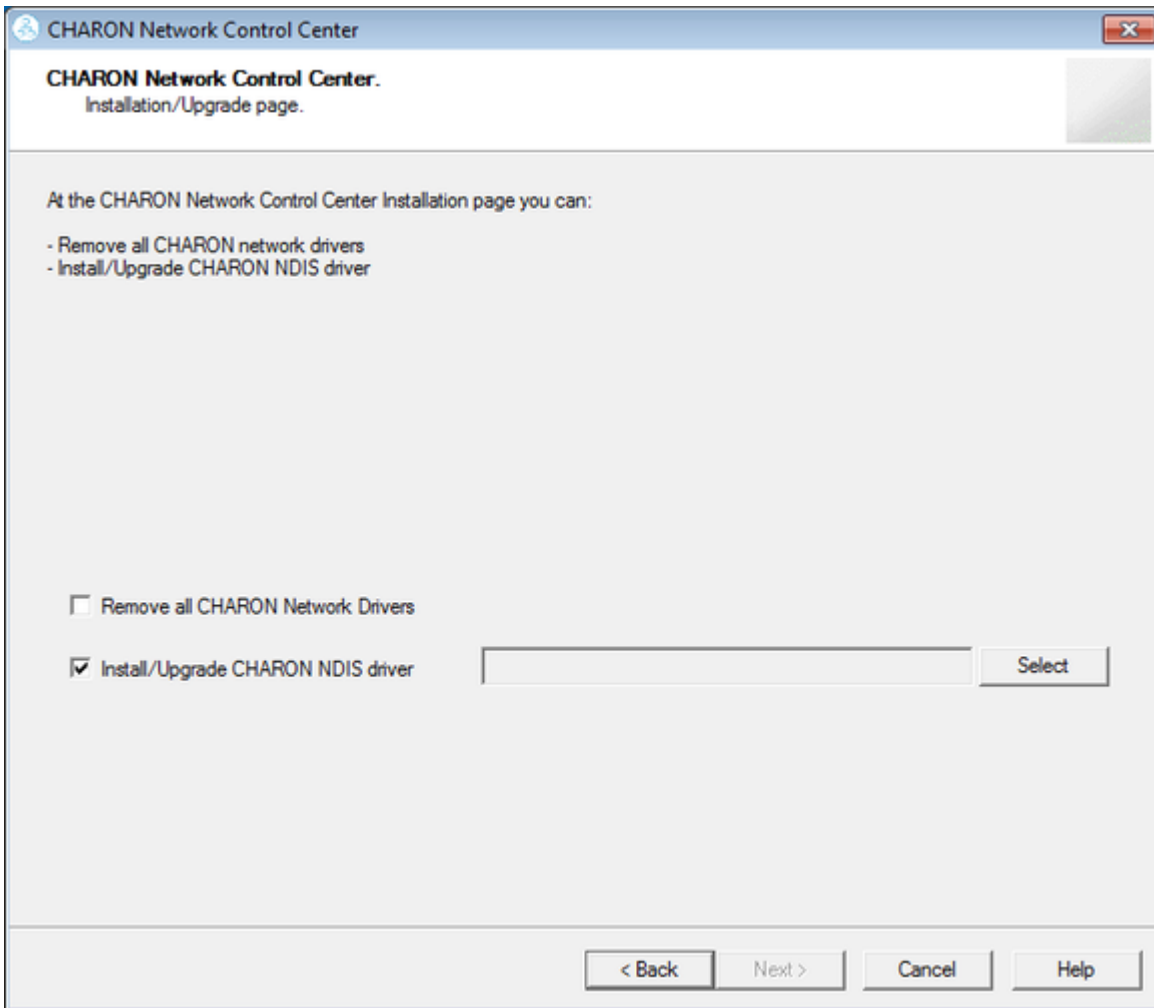
Press the "Browse" button:
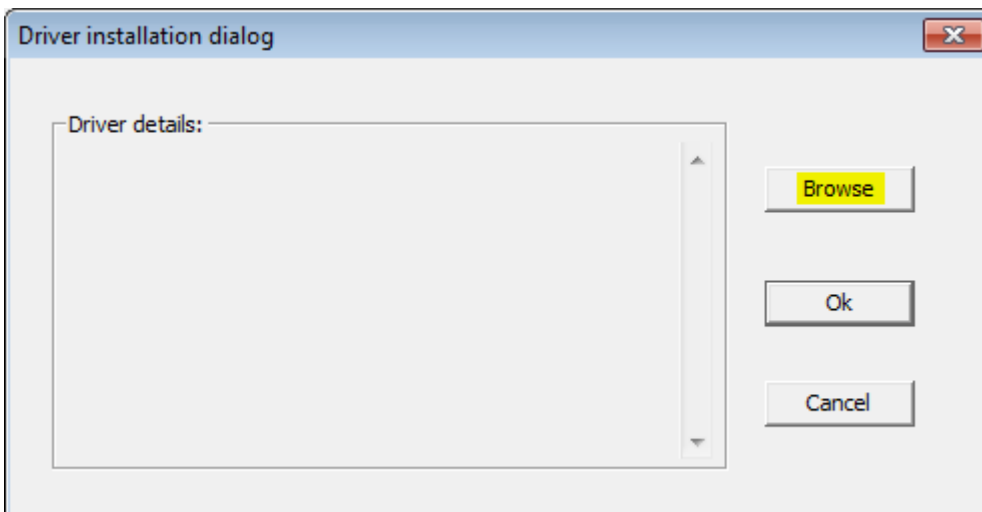
Browse for the target "`sripacket.inf`" file (by default it is located in the "`C:\Program Files\CHARON\Drivers\NDIS6_X.X.X.XXXXX`" directory), select it and press the "Open" button:
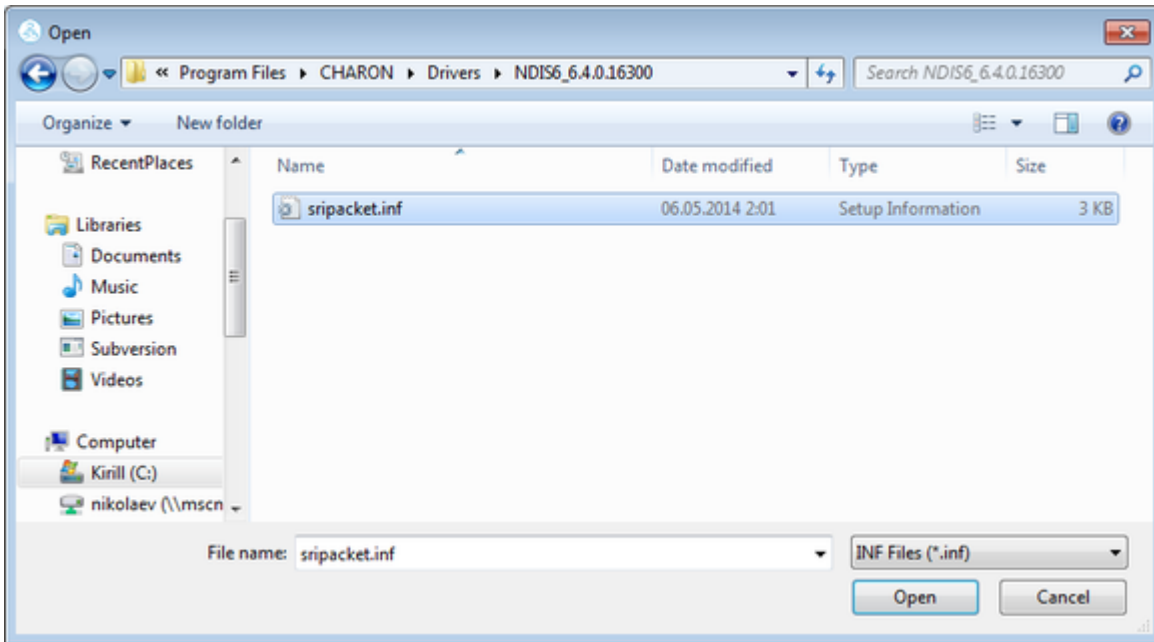
If the CHARON network driver has been acquired directly from STROMASYS, put it in a temporary directory and choose the "`sripacket.inf`" file from this directory in the dialog above.

Review the version of the driver and, if it is correct, press the "Ok" button:
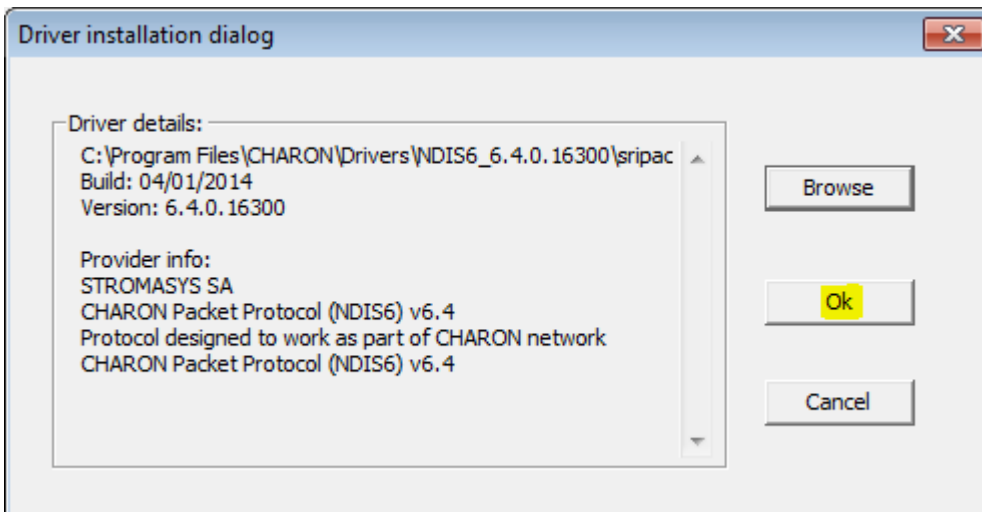
The Network Control Center will display the previous dialog with the path to the CHARON network driver. Press the "Next" button in the popup below to continue:
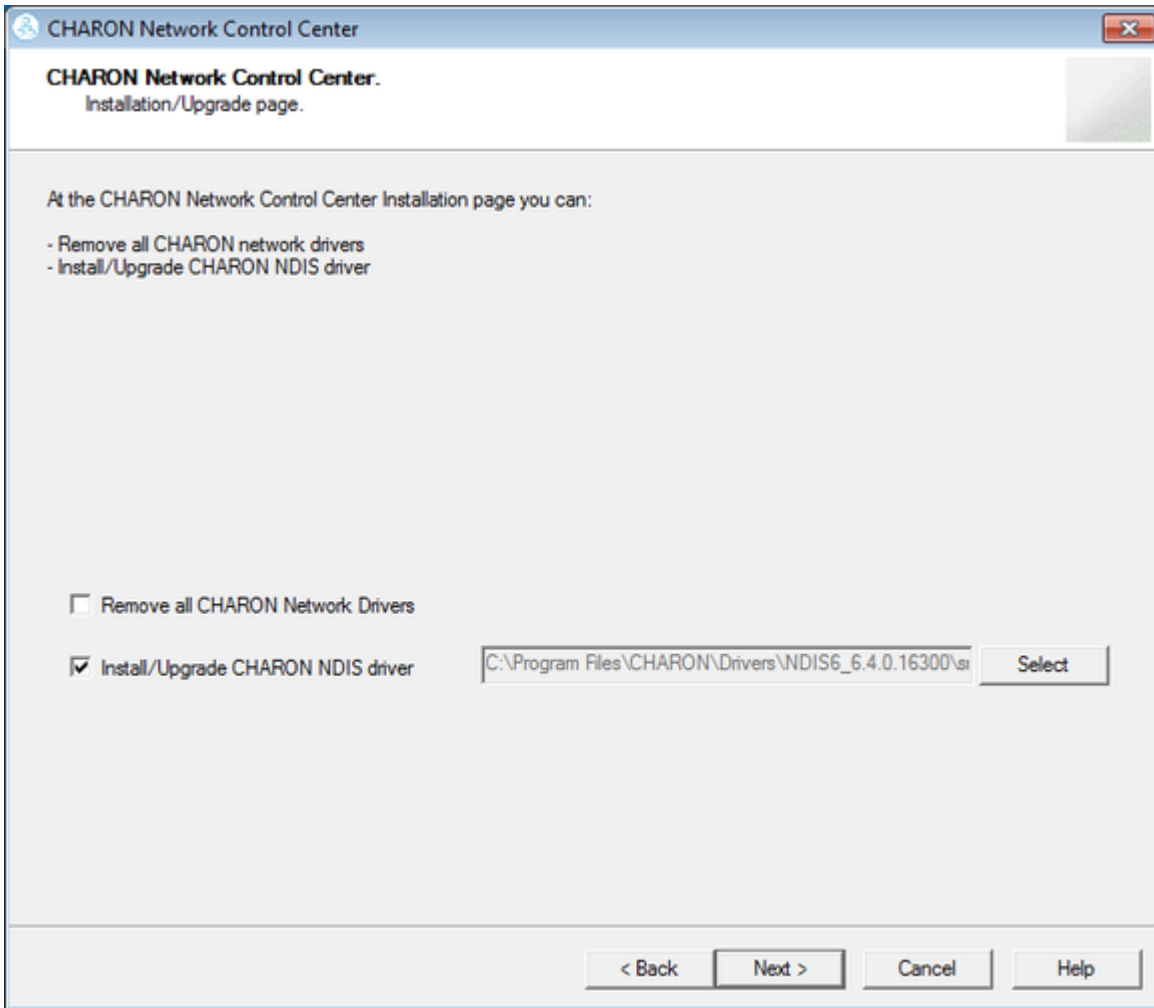
The utility will begin the CHARON network driver installation. If Windows Security asks you to confirm the driver installation, press the "Install" button:

The driver installation will resume:
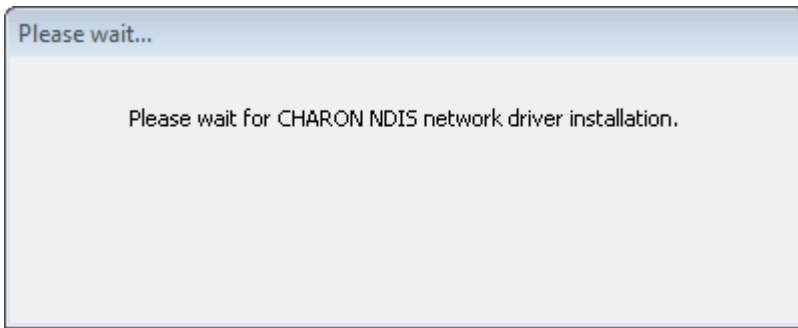
At the completion of the network installation procedure, the utility will display the following log:
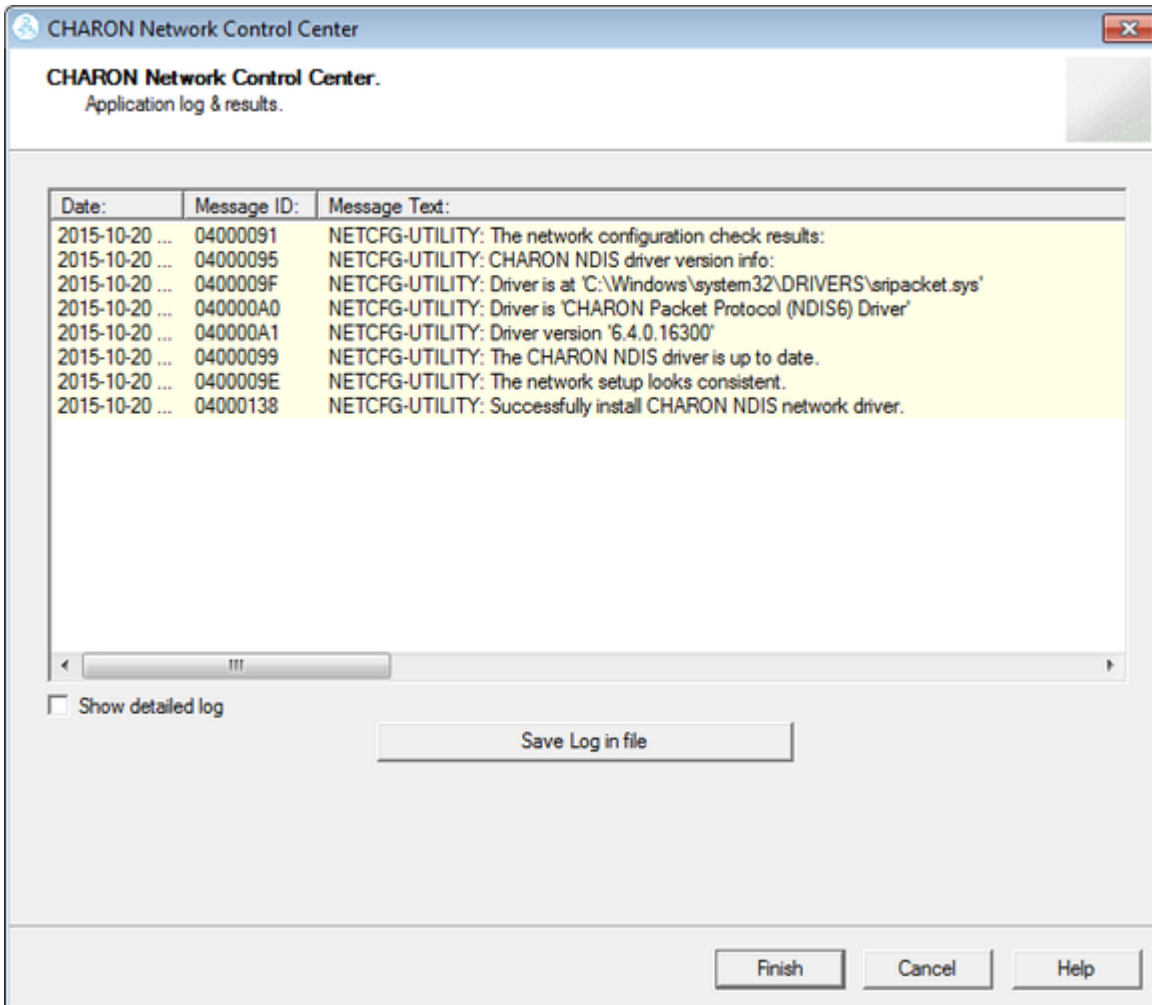
Review the log, make sure it is correct. You can save the log to a file by pressing the "Save Log in file" btton in case the log file has to be sent to Stromasys Customer Support.

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:

```
CHARON Network Control Center                                              [×]

  CHARON Network Control Center.
      Introduction page.


          The CHARON Network Control Center designed to manage CHARON networking including:

      - Installation/de-installation of the CHARON network drivers;
      - Configuring physical adapters to be used by the CHARON/Host system;
      - Troubleshooting the physical adapter setup;
      - Checking the connection reliability.
      - Checking that adapter supports MAC address change.
      - Checking for the MAC address duplication on the LAN;
      - Checking for the IP address duplication on the LAN;
      - Suggesting the configuration file settings;
      - Disable TCP chimney offload for CHARON emulators;
      - Monitoring NIC activity
            (NDIS driver v. 5.3.0 or later should be installed
             and CHARON running to enable Monitor feature);

      Select the desired function and click Next button for continue.


    ┌ Please select the action to continue: ──────────────────────────────┐
    │                                                                      │
    │        ⦿ Install/Upgrade                                             │
    │        ○ Configure NIC                                               │
    │        ○ Troubleshoot NIC dedicated for CHARON                       │
    │        ○ Disable TCP chimney offload for CHARON                      │
    │        ○ Monitor CHARON NIC activity                                 │
    │                                                                      │
    └──────────────────────────────────────────────────────────────────────┘


                        [ < Back ]  [  Next >  ]  [ Cancel ]  [  Help  ]
```

Select "Remove all CHARON Network Drivers" and press the "Next" button:
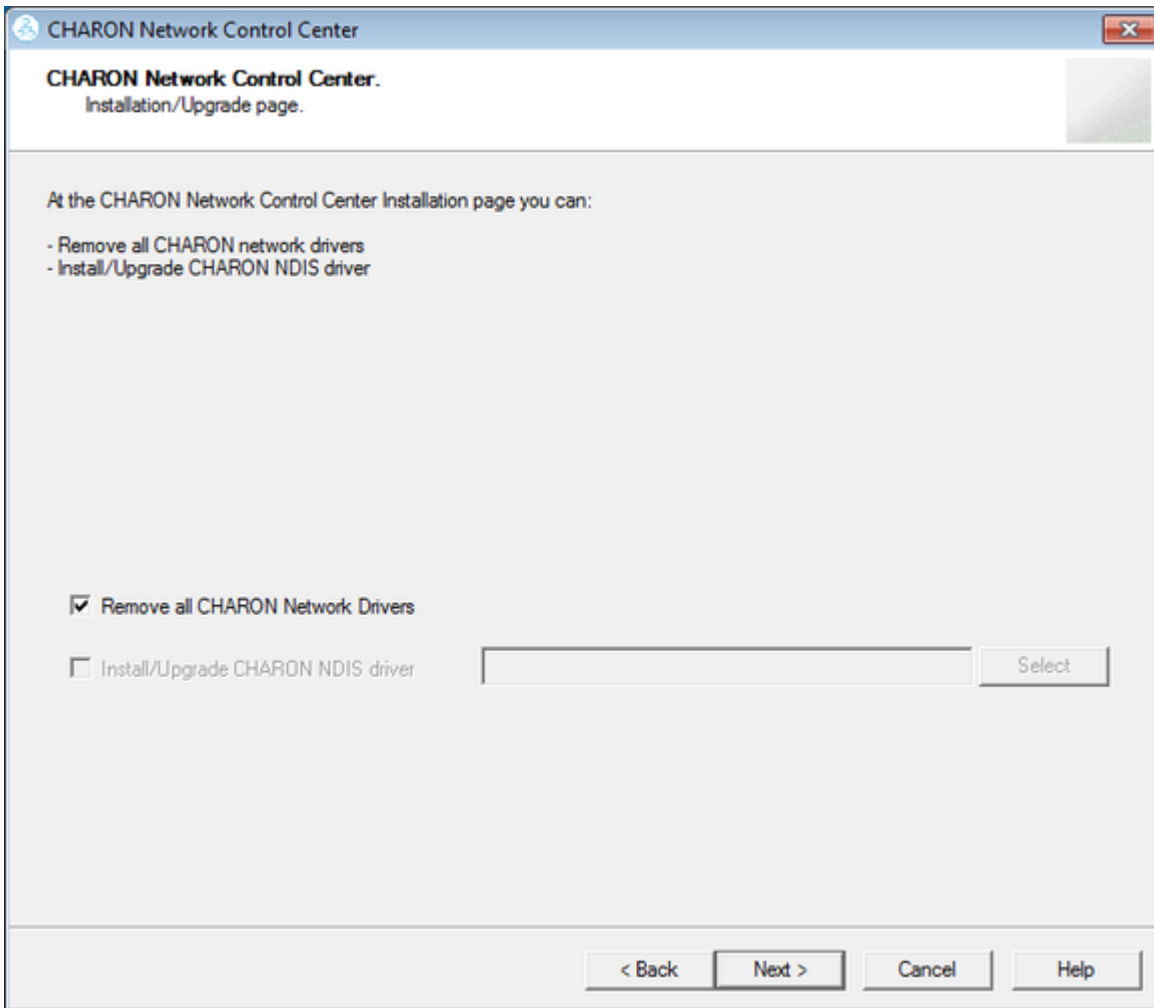
The utility will report that the CHARON drivers have been removed:
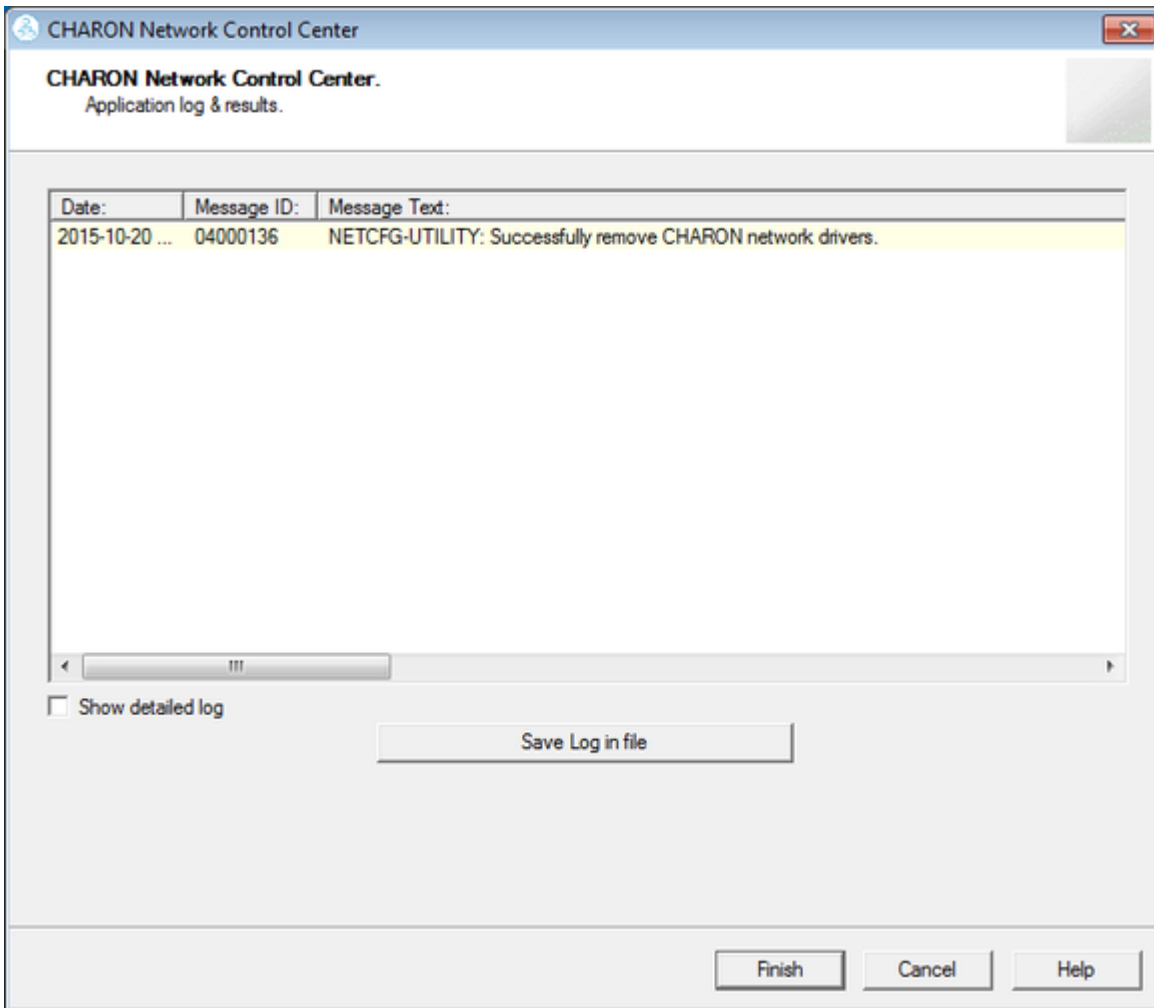
Review the log for errors. You can save the log to a file by pressing the "Save Log in file" button in case the log file has to be sent to Stromasys Customer Support.

Select "Show detailed log" to display more detail.
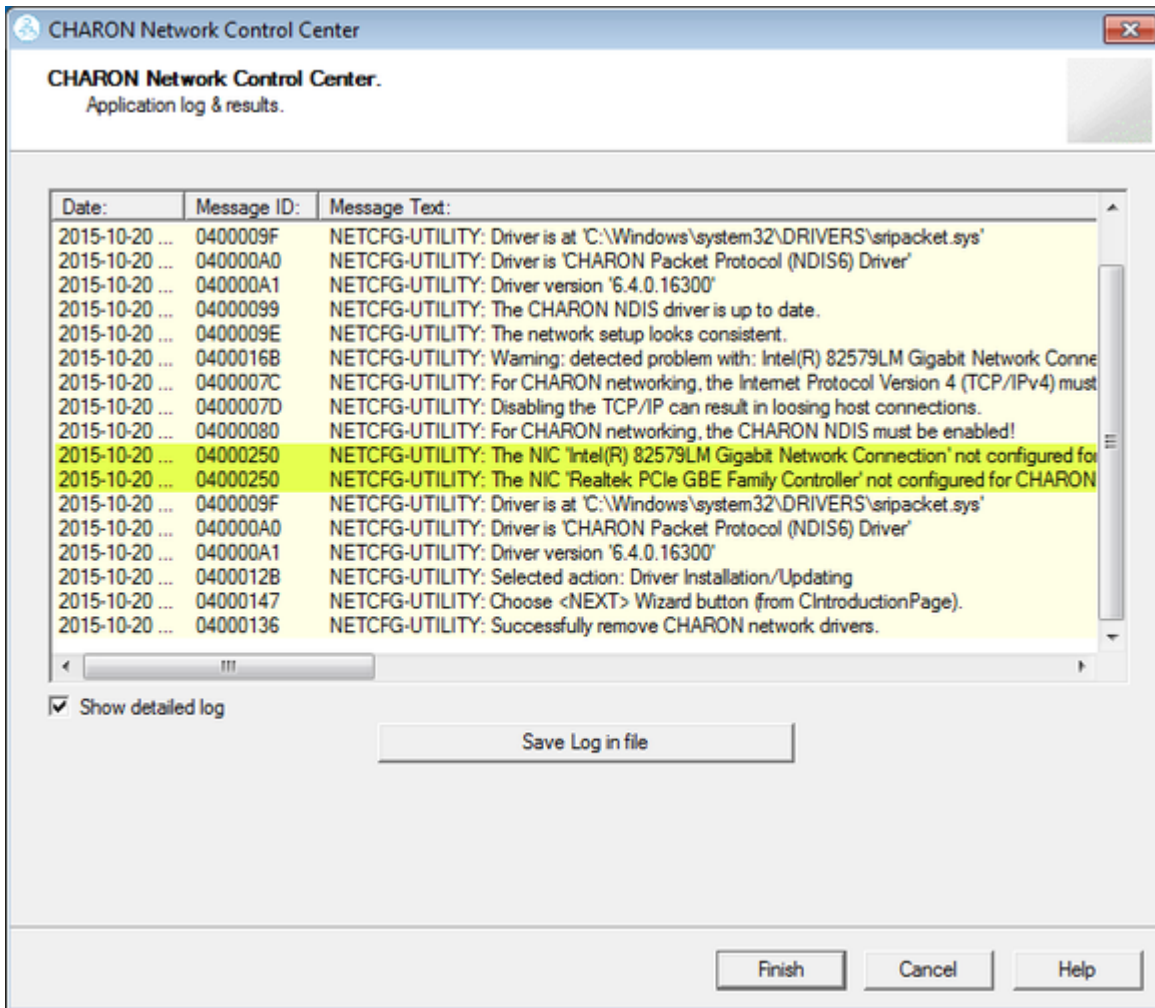
Press the "Finish" button to exit.

Configuring the host network interfaces for CHARON

Start the utility, select "Configure NIC" and press the "Next" button:
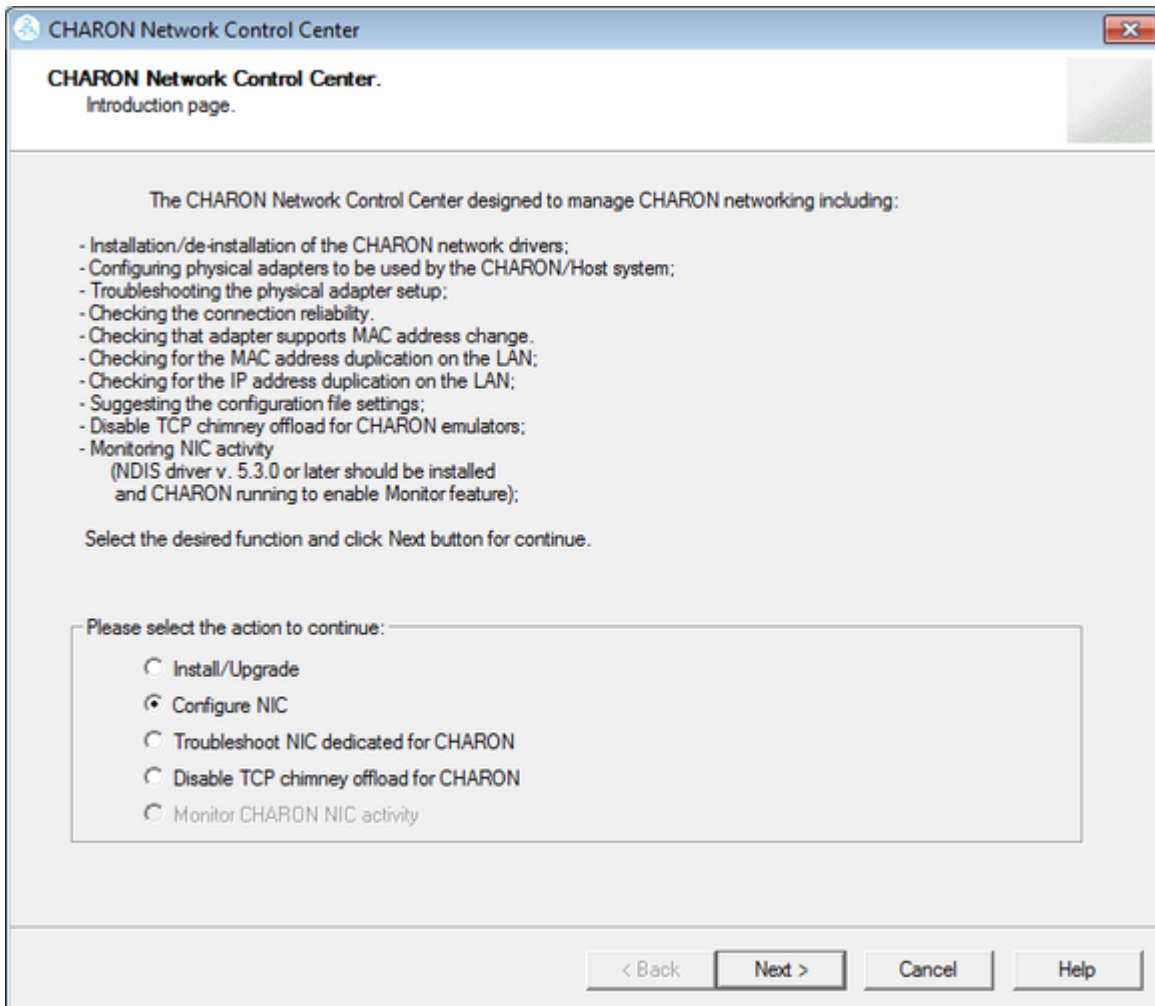
Select the host interface to be dedicated to CHARON (in the example below it is "Charon"), press "Dedicate adapter to CHARON":
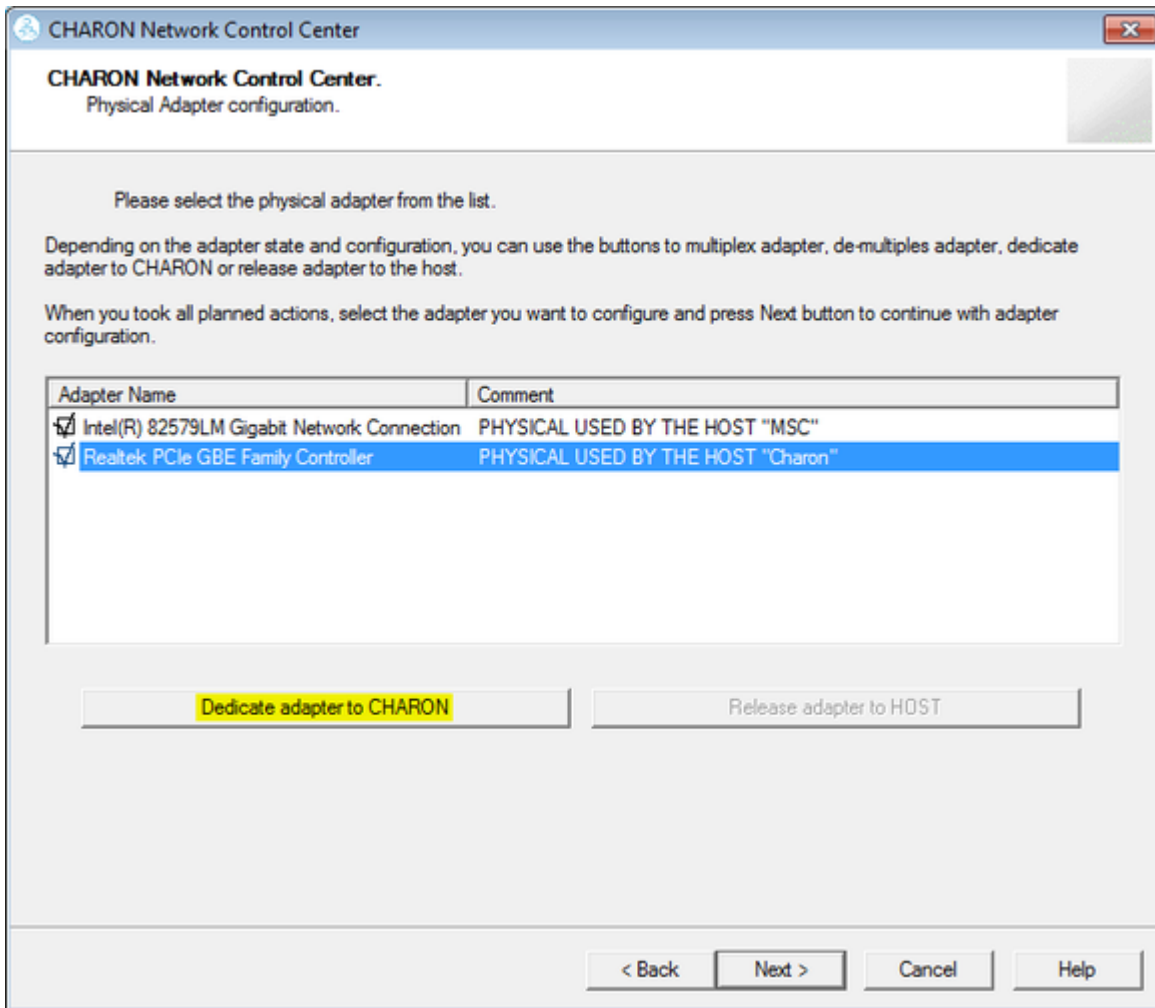
Install and configure VLAN adapters, according to the vendor's User's Guide, if required. Select the VLAN adapter in the dialog box in the above example. A VLAN adapter is not configured differently, the same procedure should be followed.

The Network Control Center will dedicate the selected adapter to CHARON. Press the "Next" button.
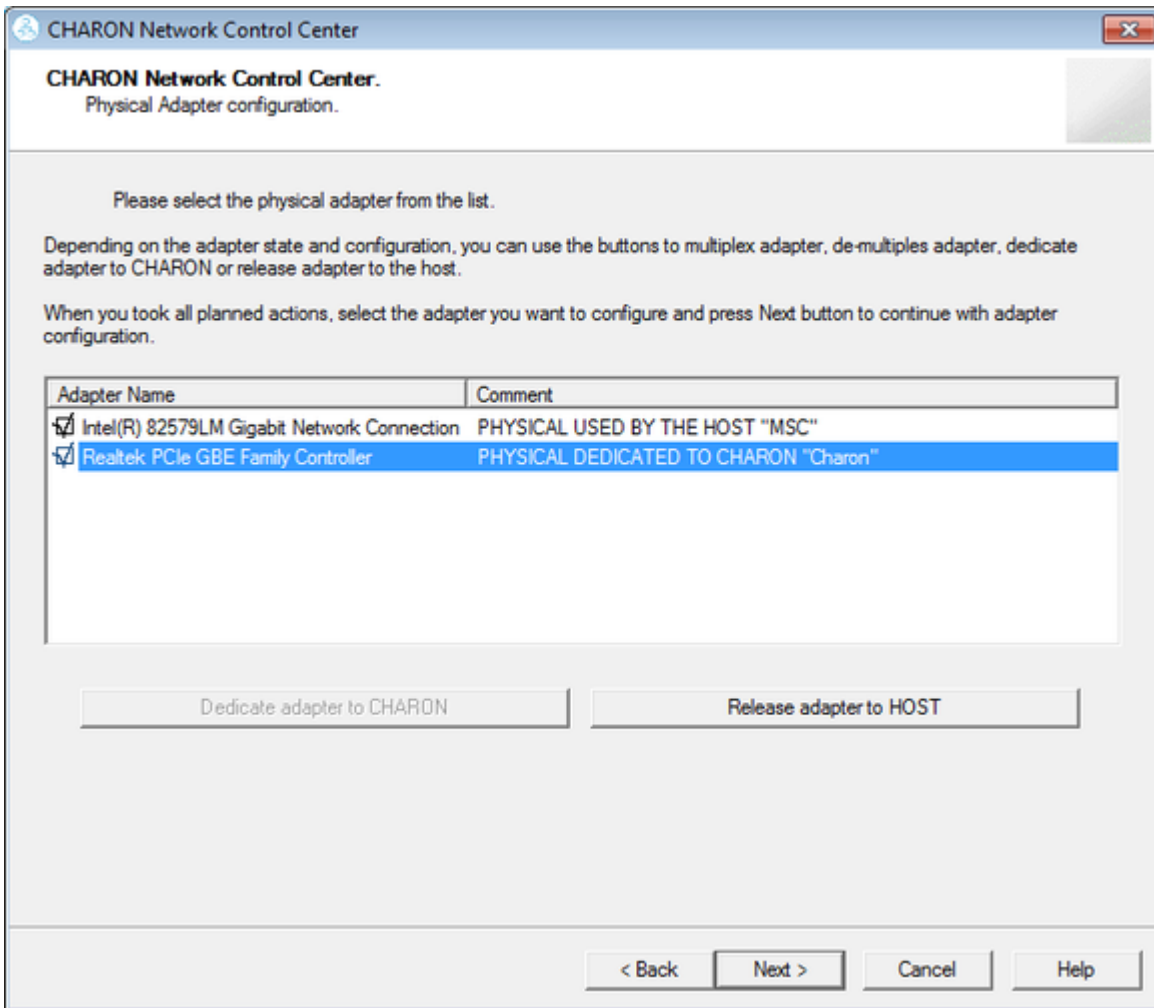
The next dialog helps to define the correct lines for the CHARON configuration file. Select the target emulated network adapter and press the "Copy suggestions to clipboard" button:

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/chnetwrk XQA_0 interface="connection:Charon"
...
```

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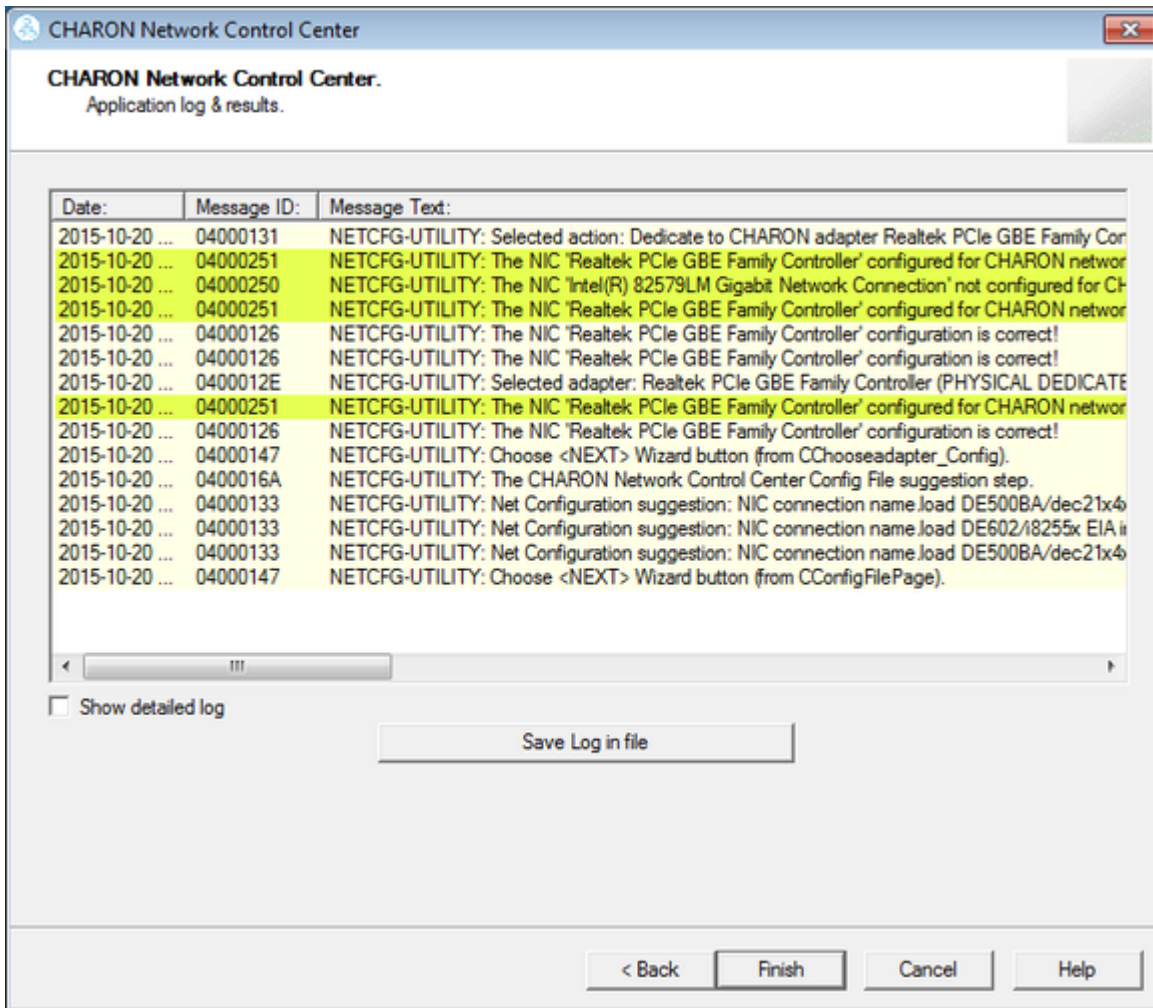
Review the log for errors. Save the log to a file by pressing the "Save Log in file" button in case the log file has to be sent to Stromasys Customer Support.

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:
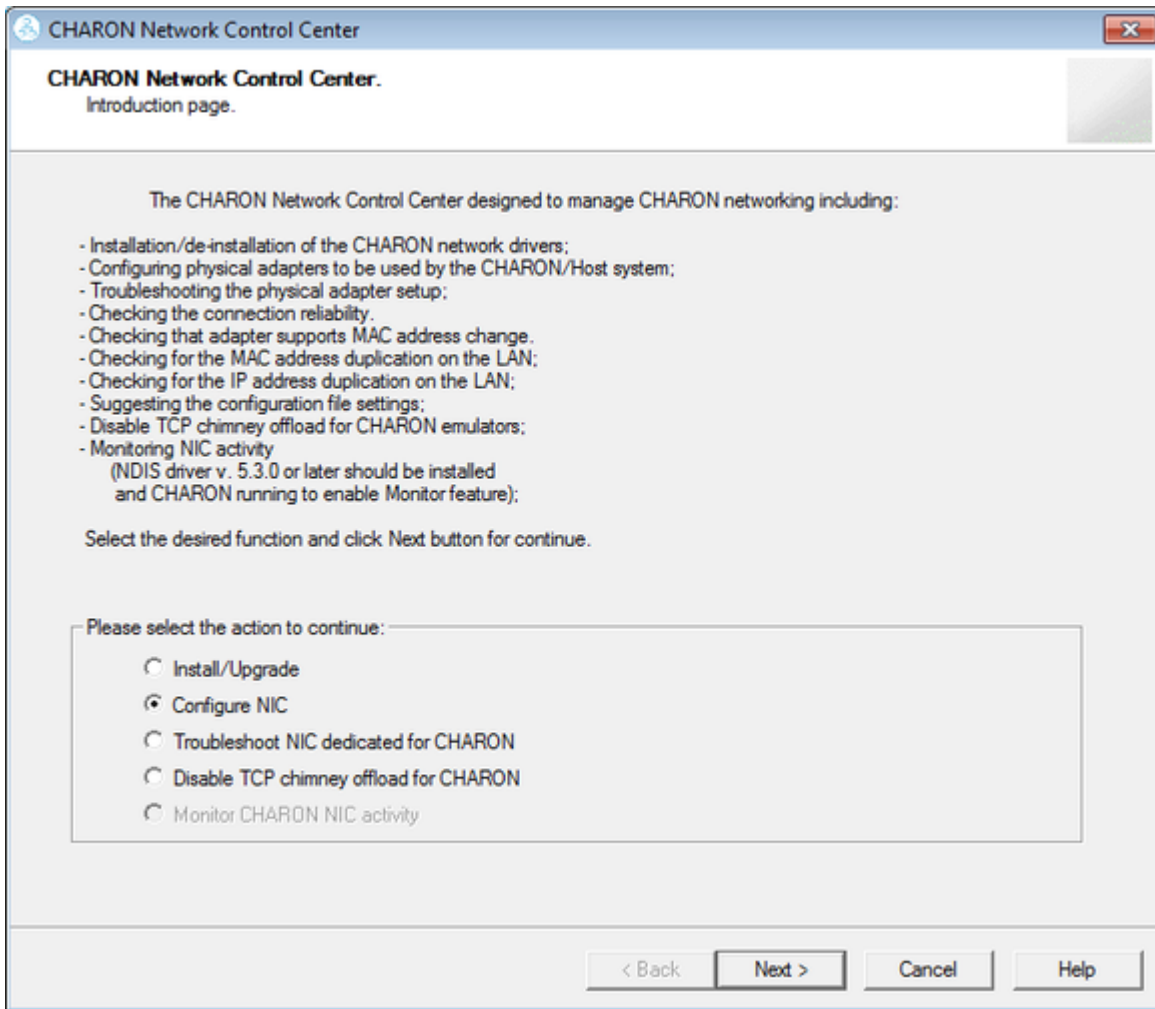
Select the interface to be released back to the host (in the example below its name is "Charon"), press the "Release adapter to HOST" button:
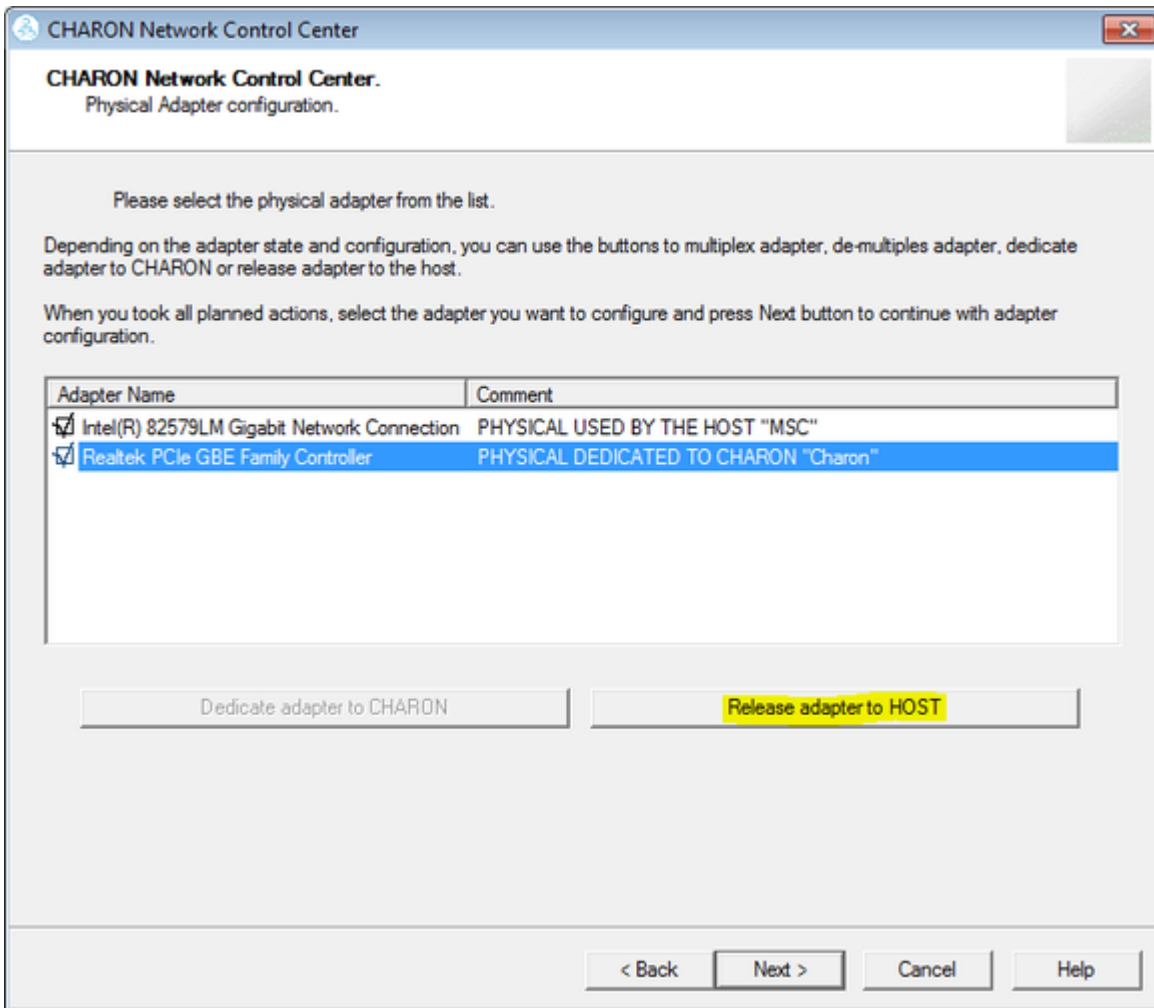
Press the "Next" button and review the log of this operation:
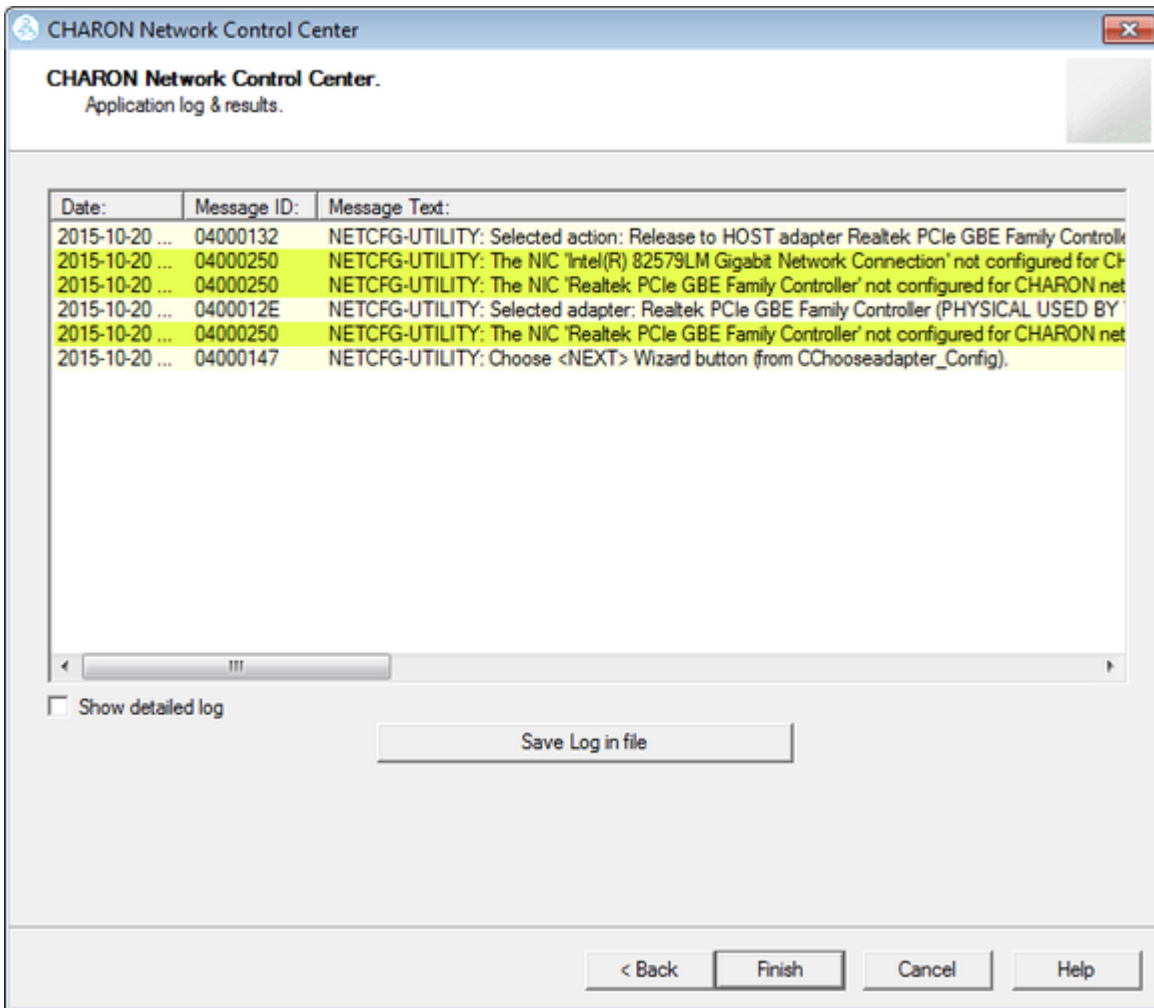
Select "Show detailed log" to display more detail.

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:
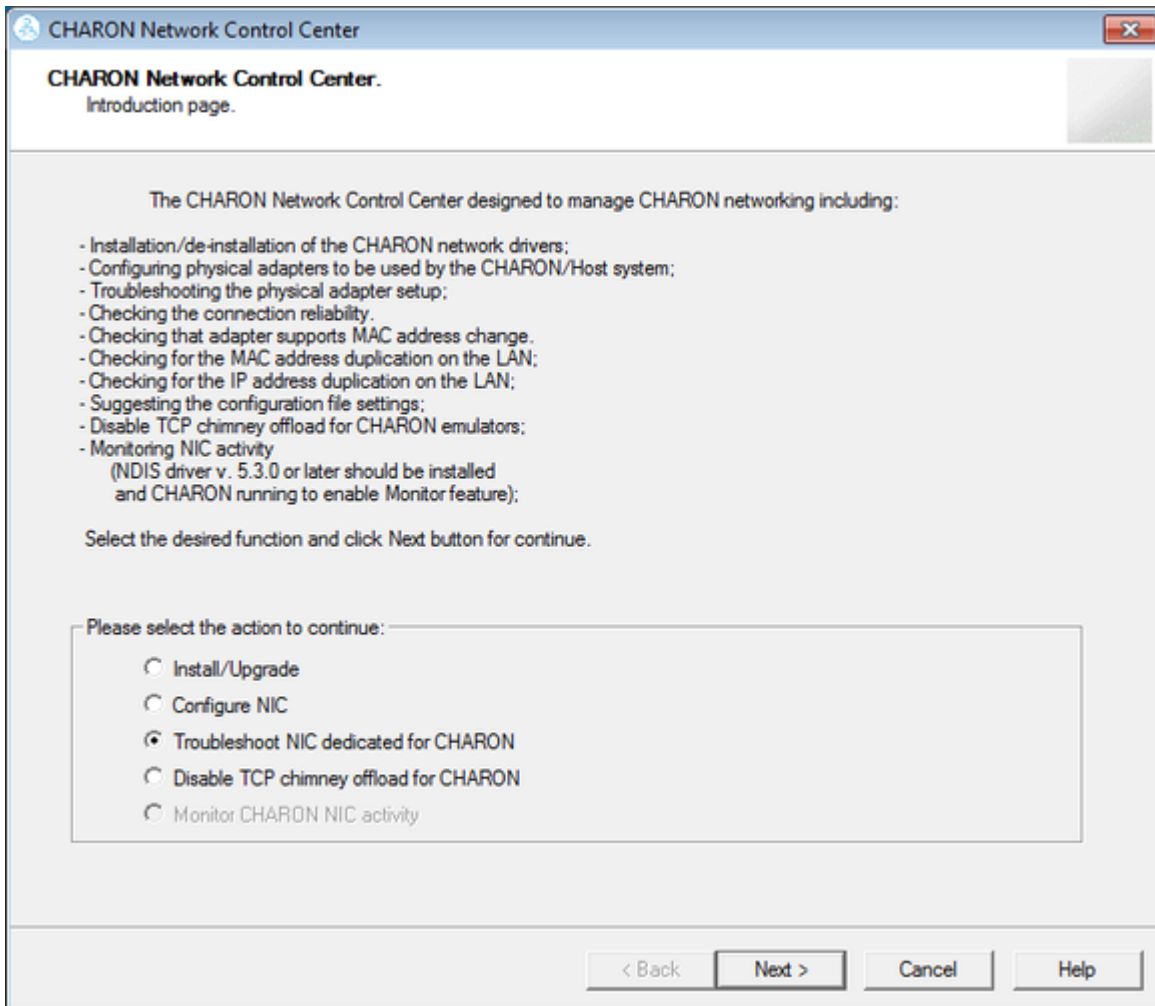
Select the target interface and press the "Next" button:
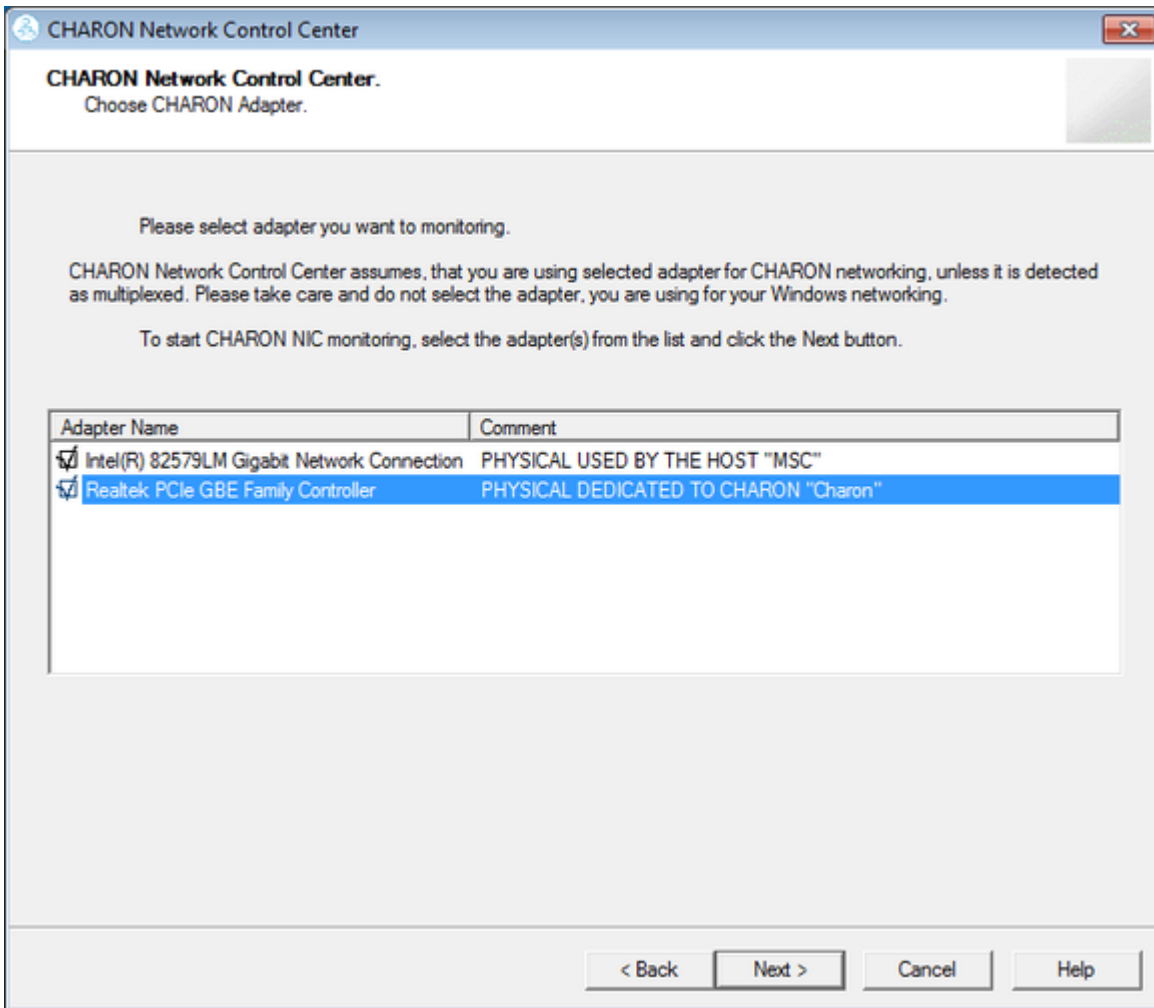
Review the status of the interface:
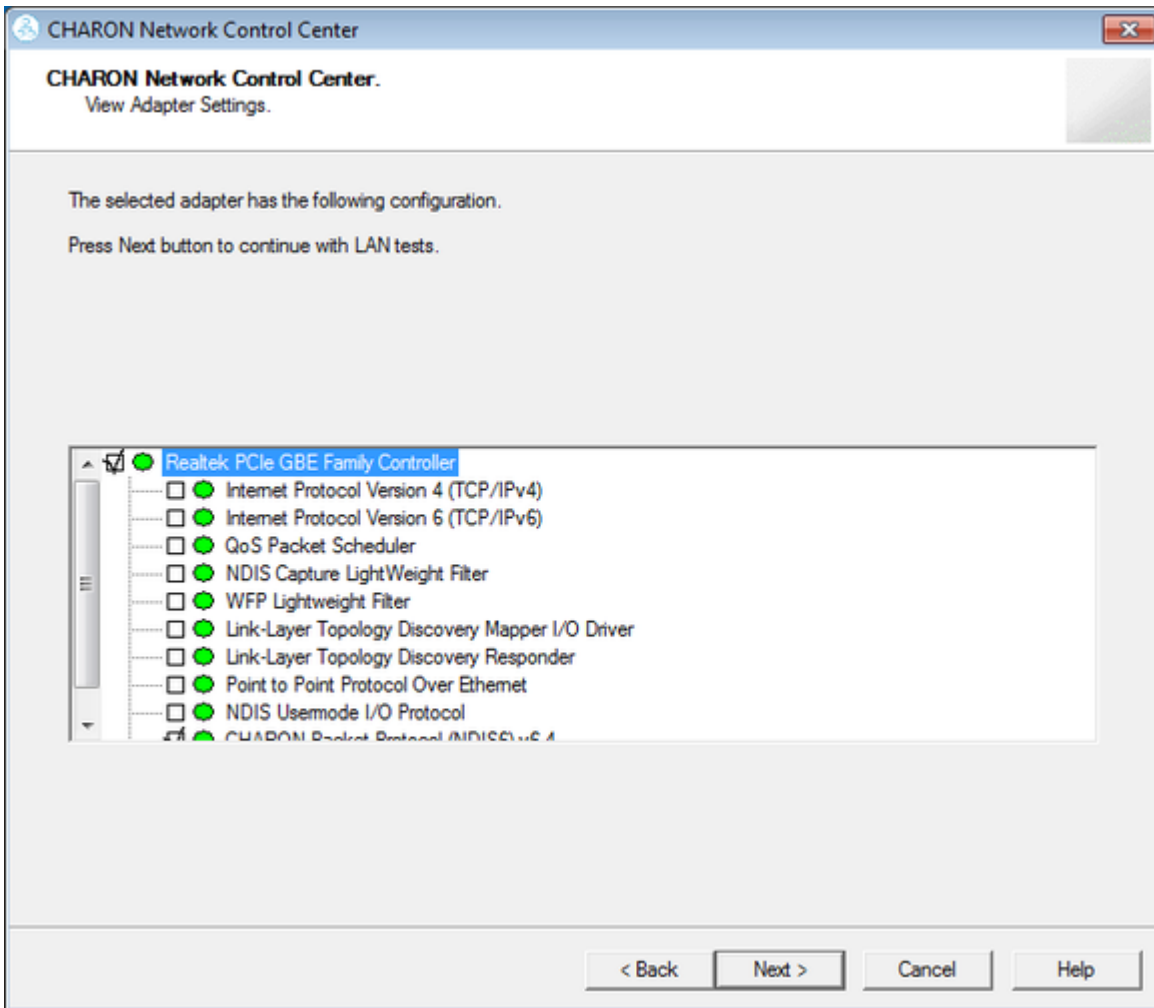
In the example below, there is an issue with the host TCP/IP being enabled on the CHARON interface. Review the problem description, select "Fix the adapter configuration" and press the "Next" button.
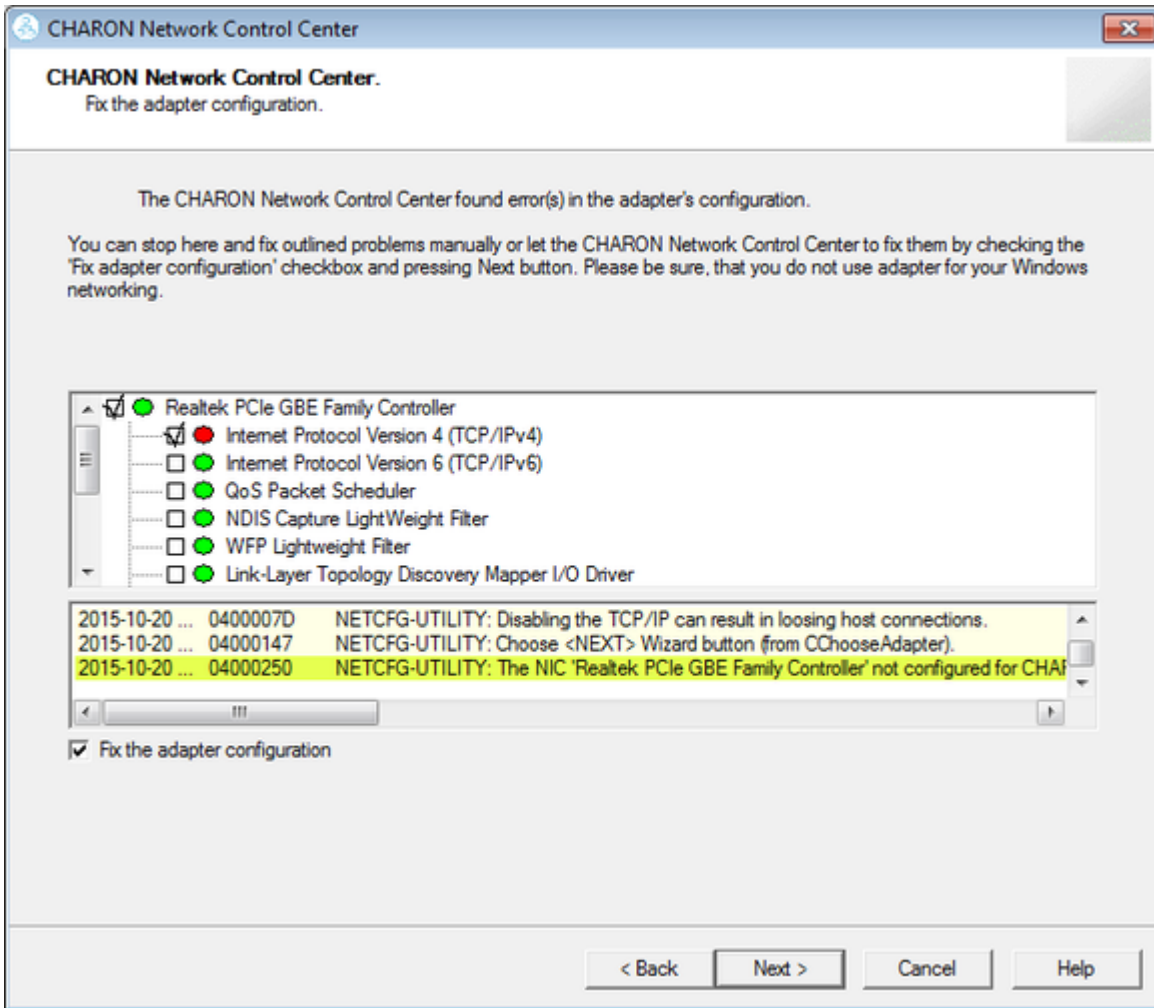
The utility will fix issues and report a good status for the interface. Press the "Next" button, the following dialog will appear:
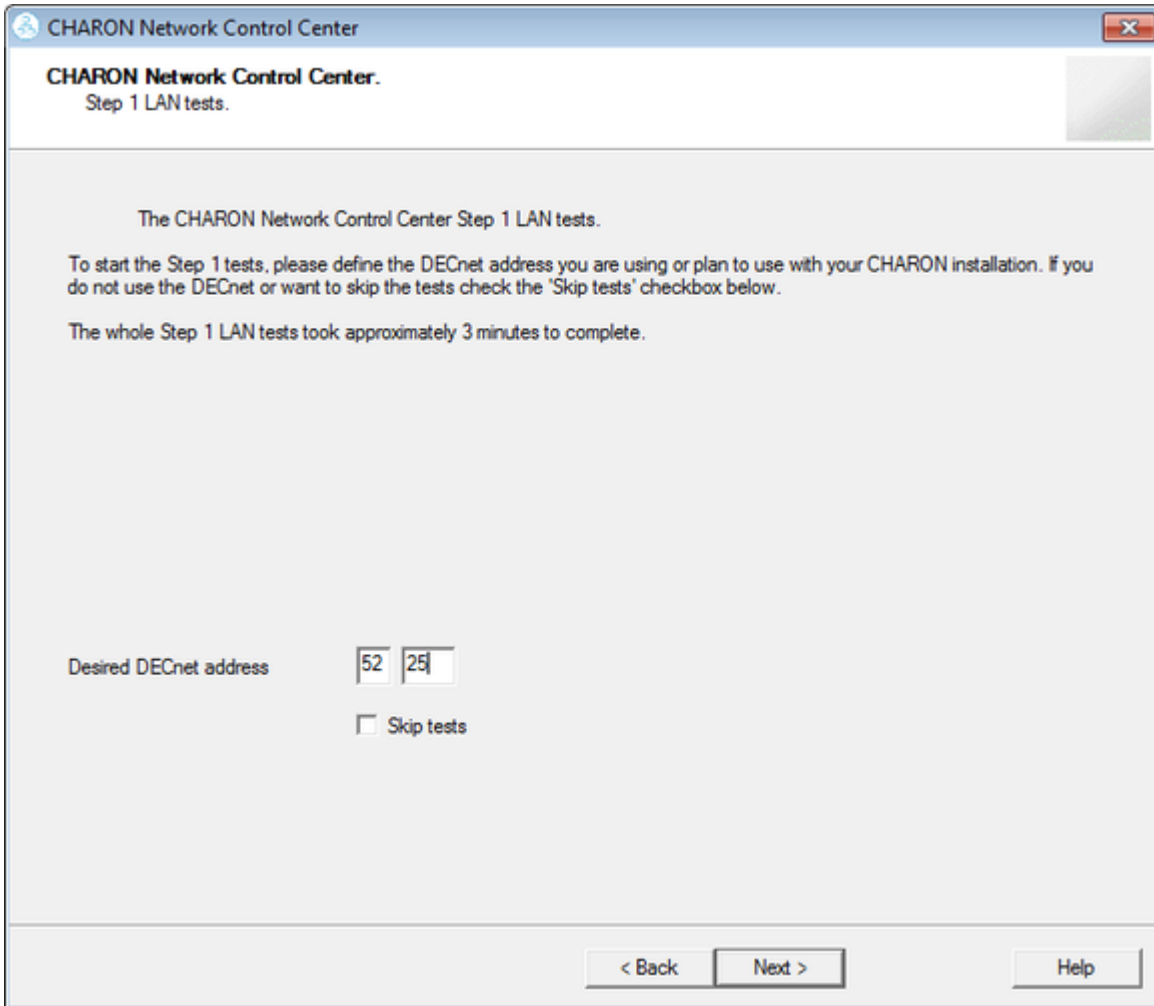
The Network Control Center offers to check whether the DECnet address, to be used by CHARON, is unique on the network in this step. Enter the desired DECnet address, for the CHARON guest, and press the "Next" button:
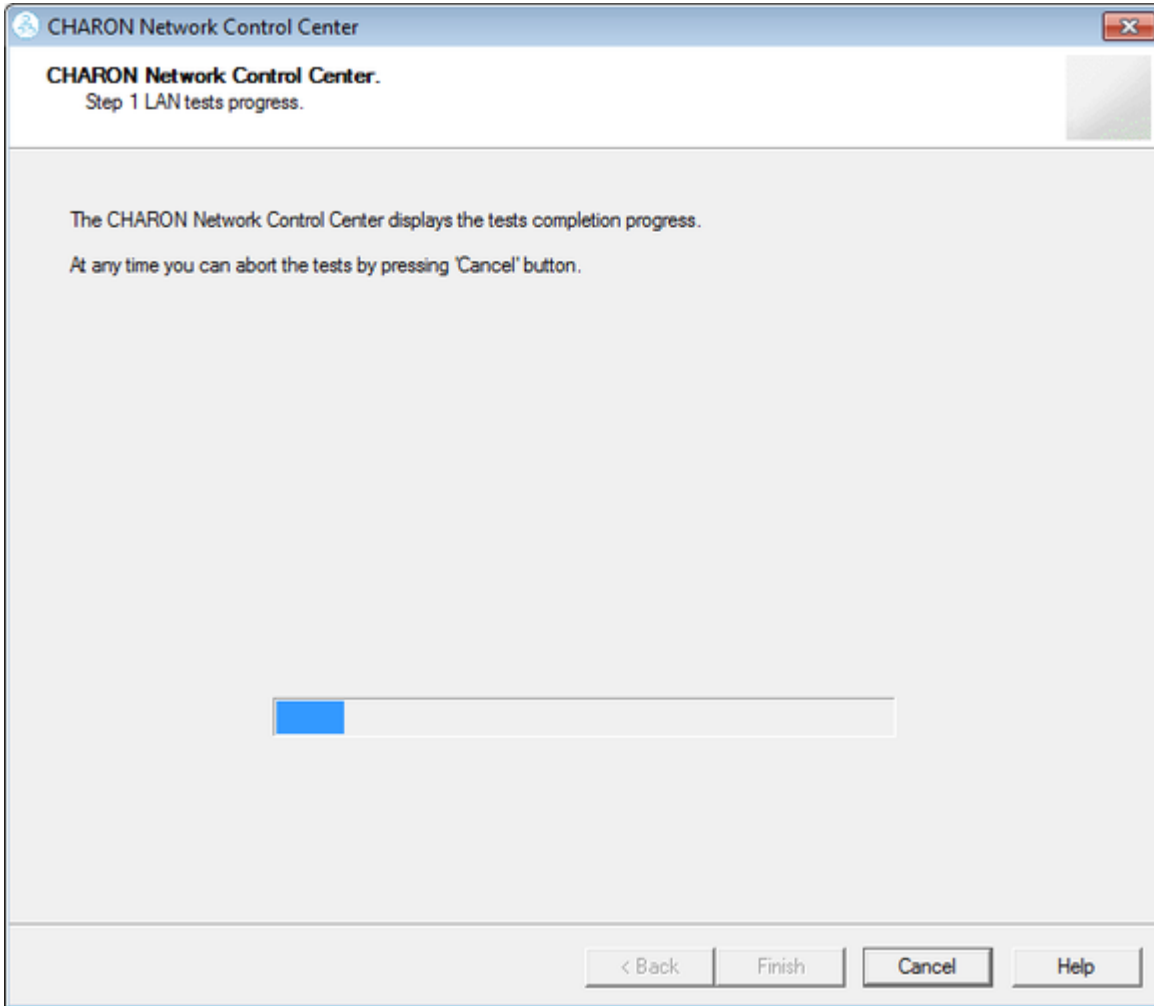
Once the checking is completed, if there are issues, the utility will abort the test and display a log containing information about the issues found:
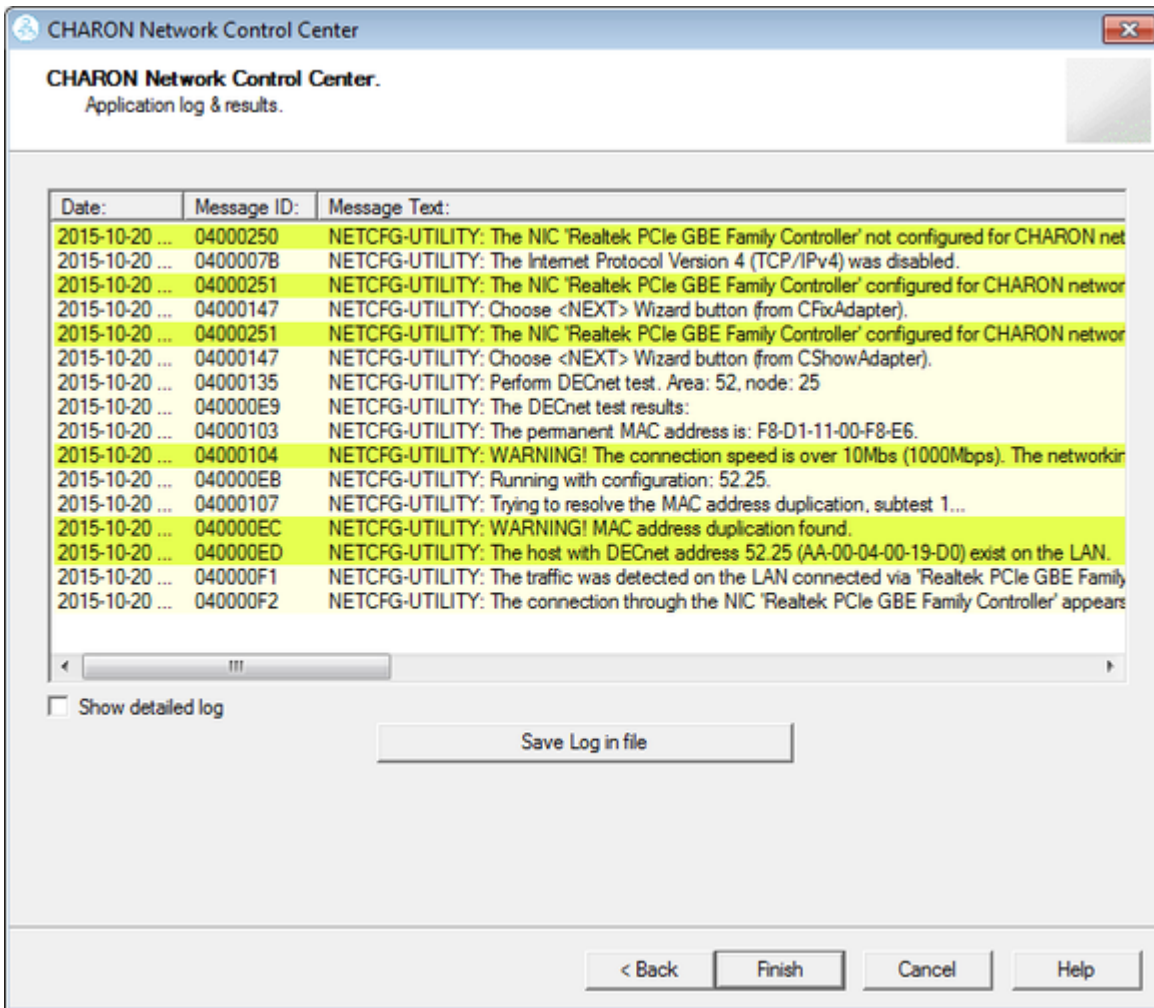
If no issues are found, the utility will offer to check the CHARON TCP/IP networking the same way:

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

The Network Control Center will display the log of the performed operations:
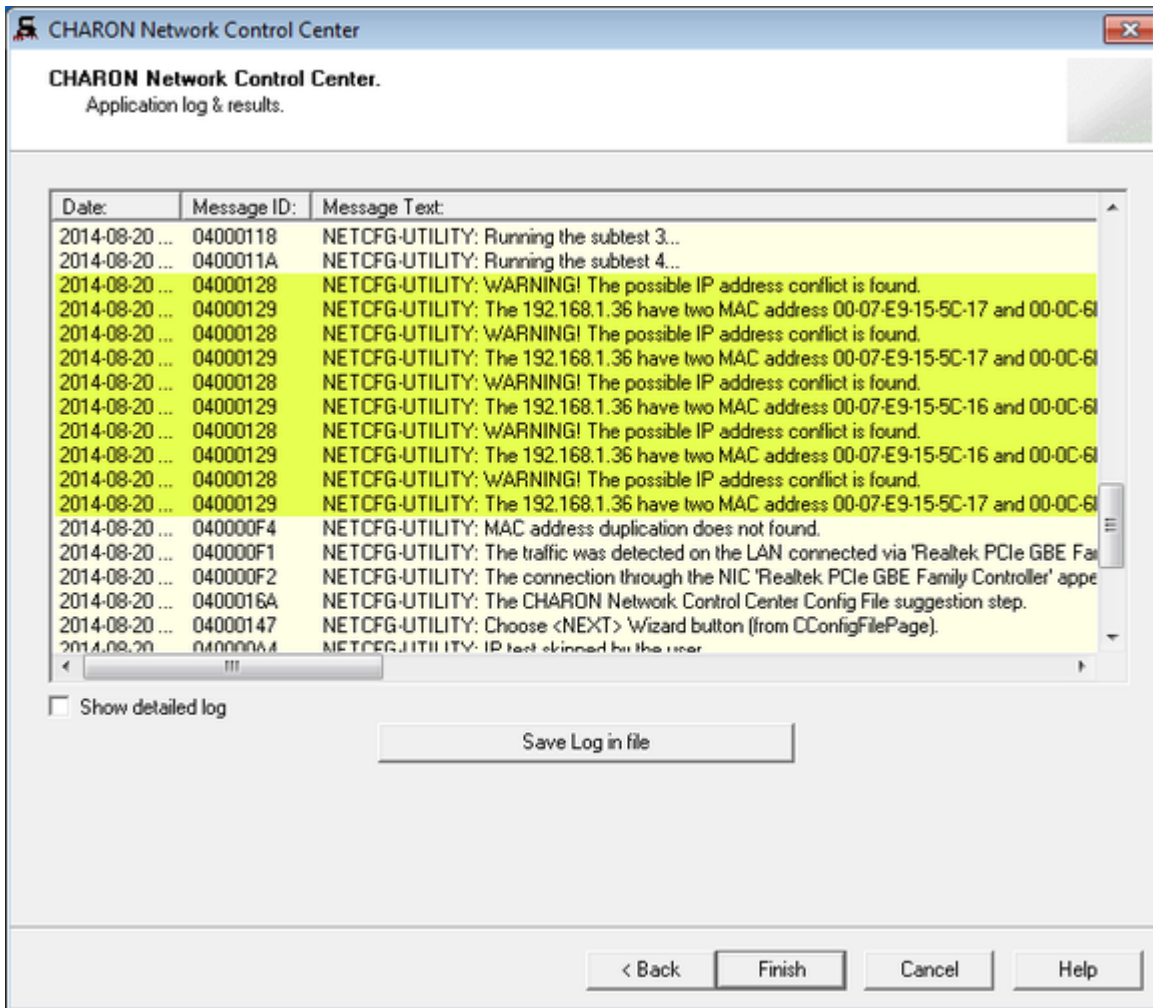
Select "Show detailed log" for more detail.

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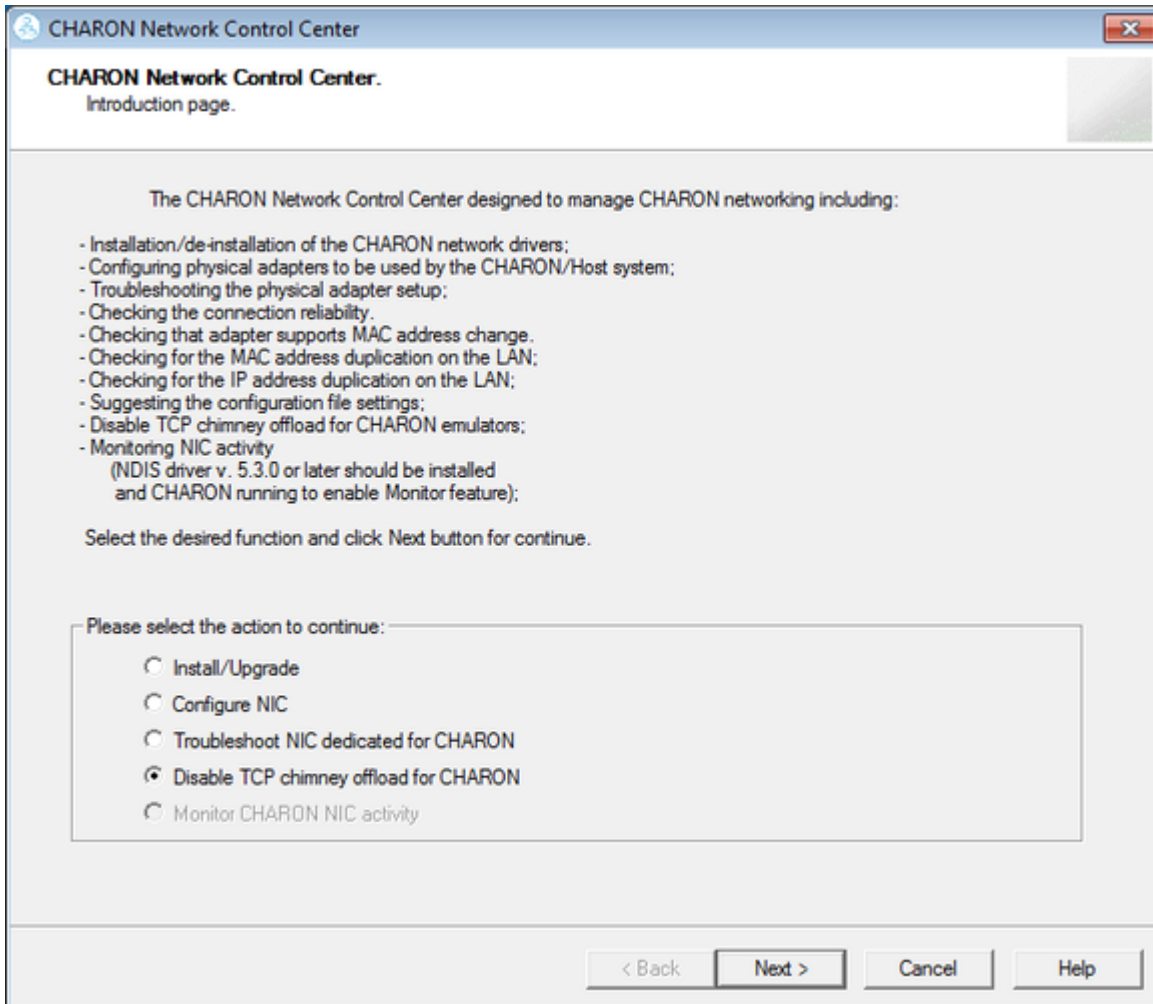
The "offload" parameters will be reset for all CHARON emulators (executable files).

Press the "Finish" button to exit.

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

⌄ More information on details of disabling TCP chimney offload is available

Operating system	Notes	Charon version	Configuration steps

Windows 2008 R2
 	
Windows 2012 R2
 	
Windows 2016
 	
Windows 7
 	
Windows 8 & 8.1
 	
Windows 10
	TCP Chimney offload must be disabled.
	
	Note: TCP Chimney Offload is a networking technology that helps transfer the workload from the CPU to a network adapter during network data transfer.
	
	⚠️ **Warning**
	The commands to enable and disable TCP Chimney Offload for specific applications and ports require that the Windows Firewall service and Base Filtering Engine (BFE) services are running. Before using these commands or the Network Control Center Utility, ensure that the Windows Firewall service and BFE service are running.
	
	⚠️ Running guests will have to be powered off/on for the new settings to be taken into account.
	
	💡 To check TCP Chimney offload is disabled, use the following commands:
	
	`C:\Windows\system32>`**`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`
	`...`
	
	💡 To revert back to the original settings manually, use the following command:
	
	`C:\Windows\system32>`**`netsh int tcp set global chimney=automatic`**
		• **CHARON-VAX V4.7 and CHARON-AXP V4.7 Build 171-10 and above (patched)**
		• **Charon-AXP/PDP/VAX V4.8 Build 183-02 and above (patched)**
			Configuration type 1

		Older versions of Charon-AXP/PDP/VAX ⚠️ Do not use the network control center utility to disable TCP chimney offload. Please use only the manual settings described in this chapter	Configuration type 2
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:
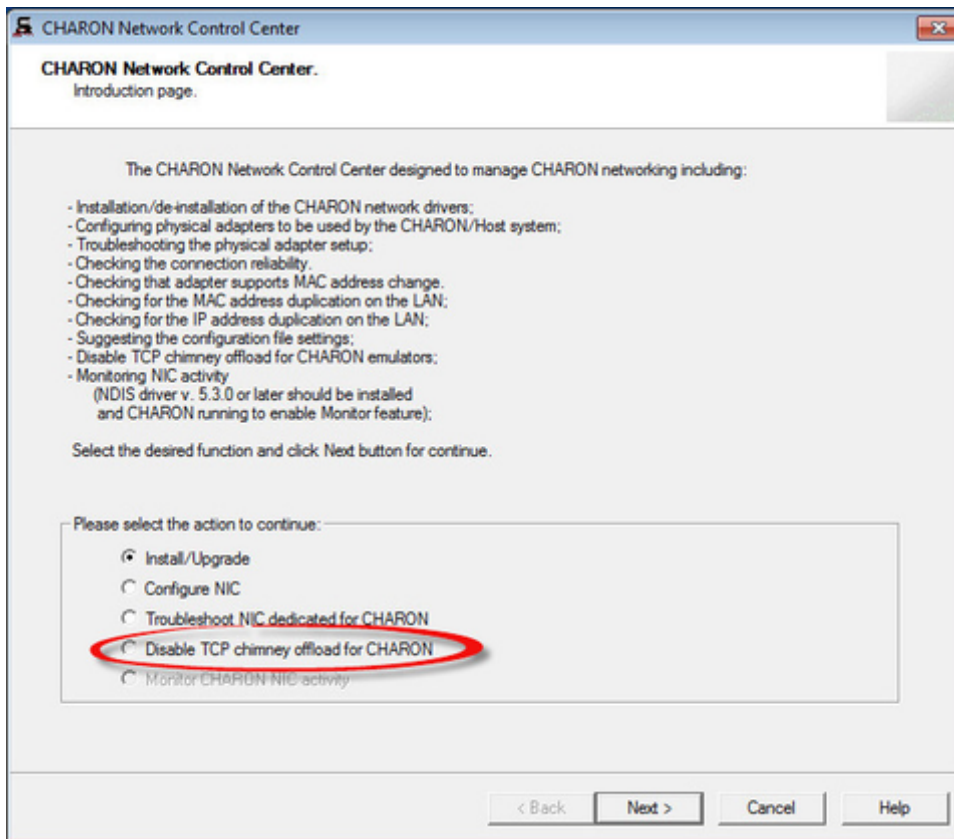
- Click on the **"Next"** button.

- The TCP Chimney offloading will be disabled:
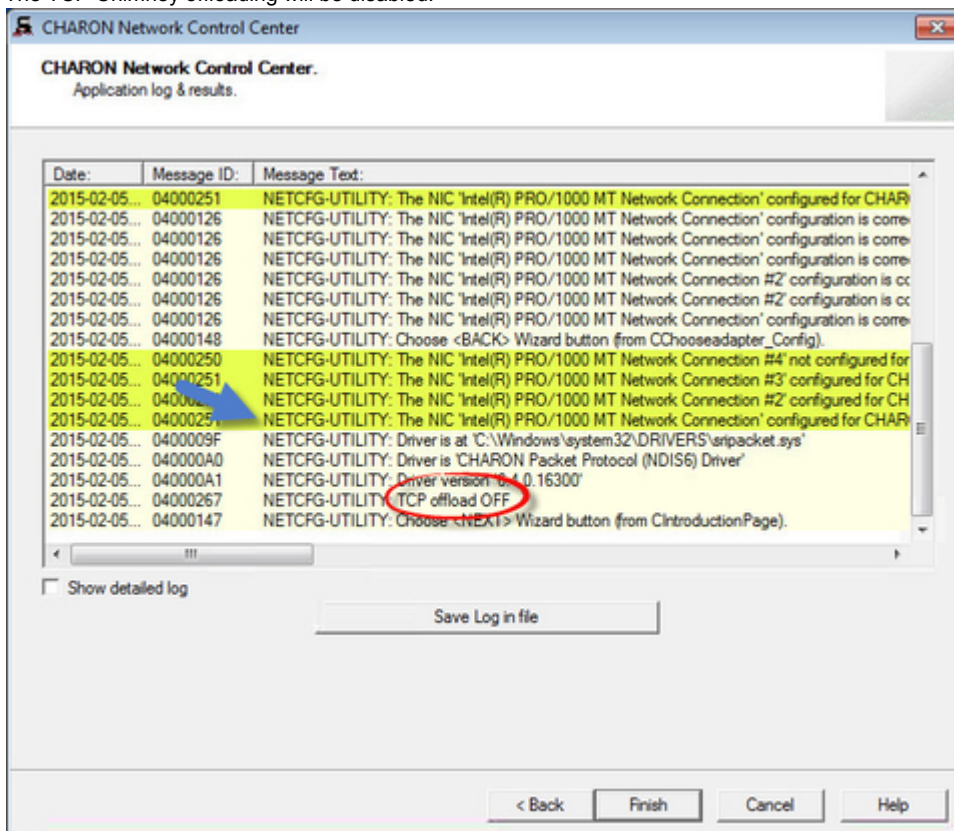
- Click on the "**Finish**" button to exit.

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 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):

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

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

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

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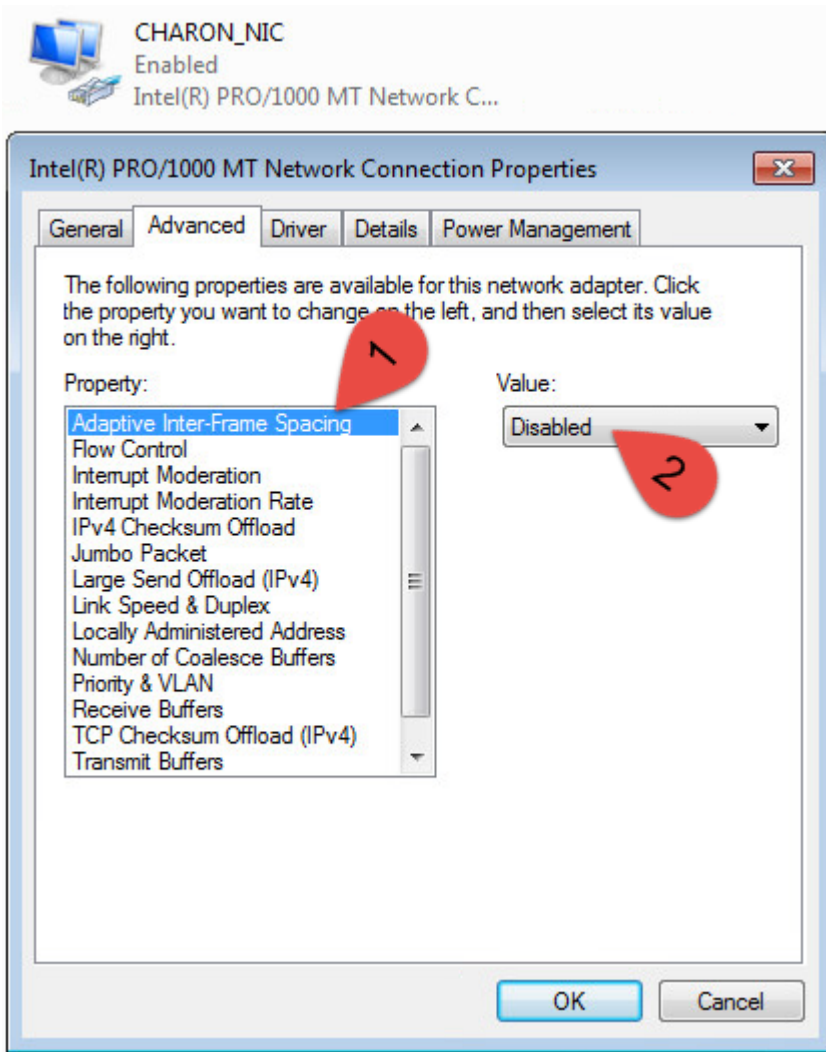
Once all parameters have been turned off, click on the "**OK**" button.

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:
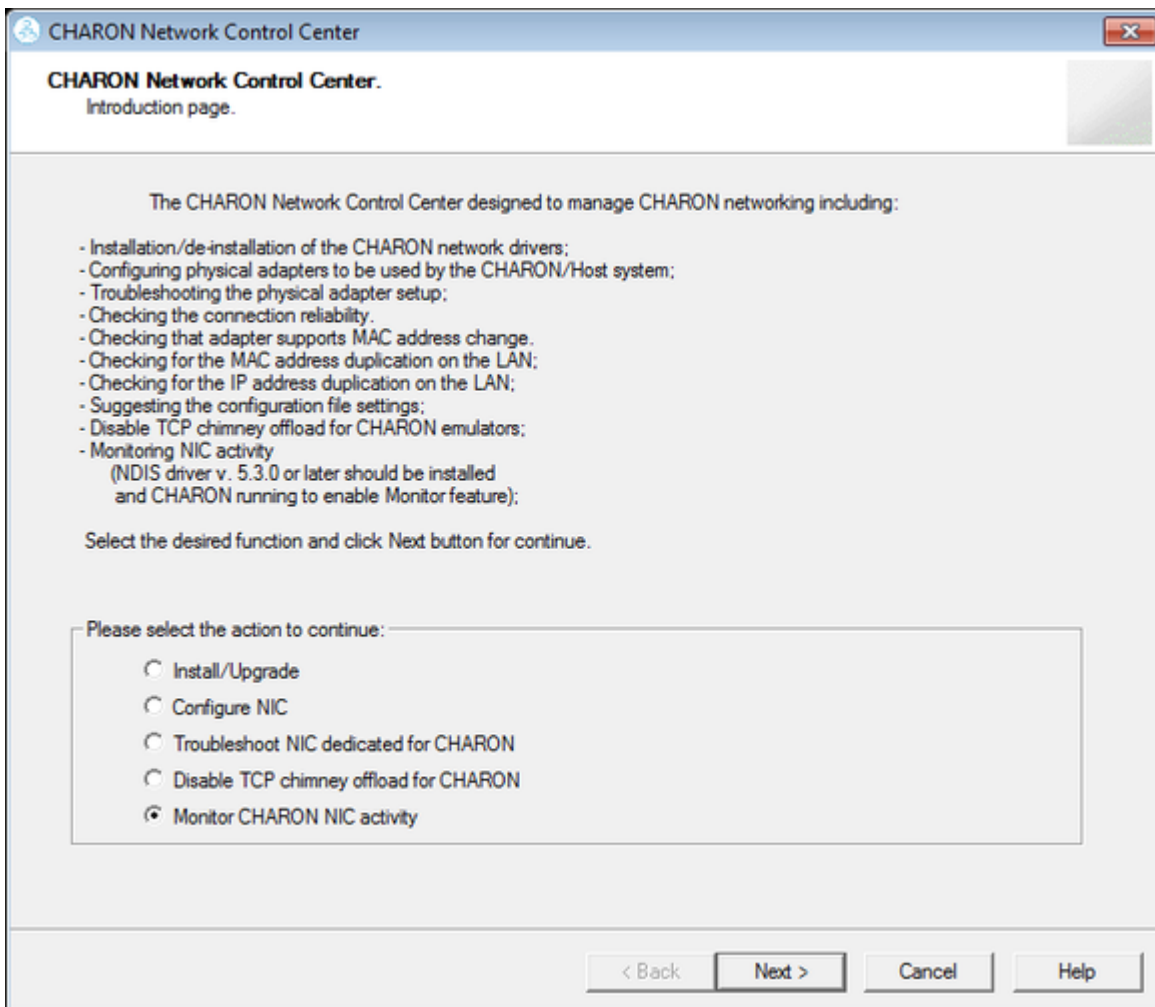
Select the network interface to monitor (it must be dedicated to CHARON) and press the "Next" button:
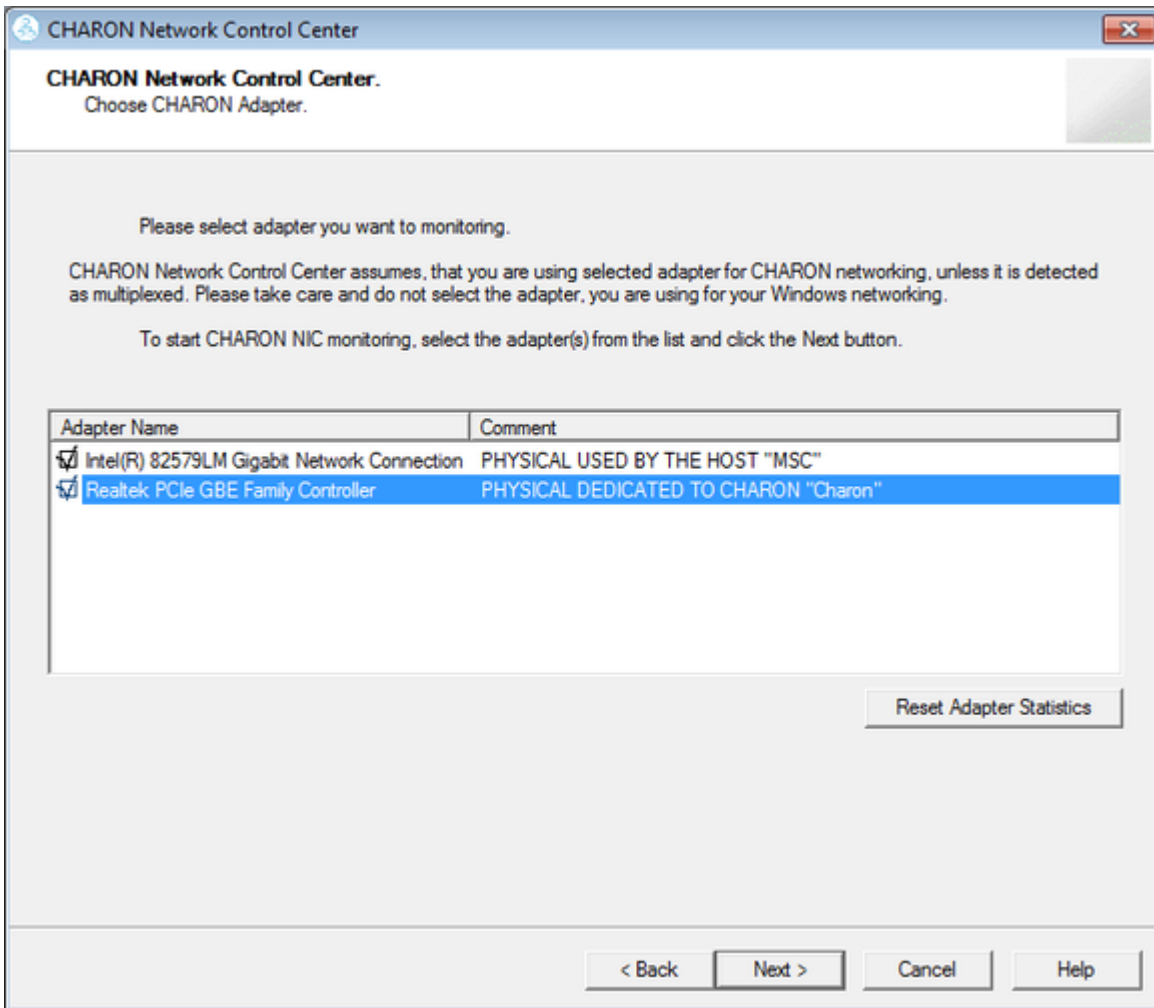
The utility will display statistics updated in a real time. Note, it is possible to record the statistics in a log file with a selected write interval.
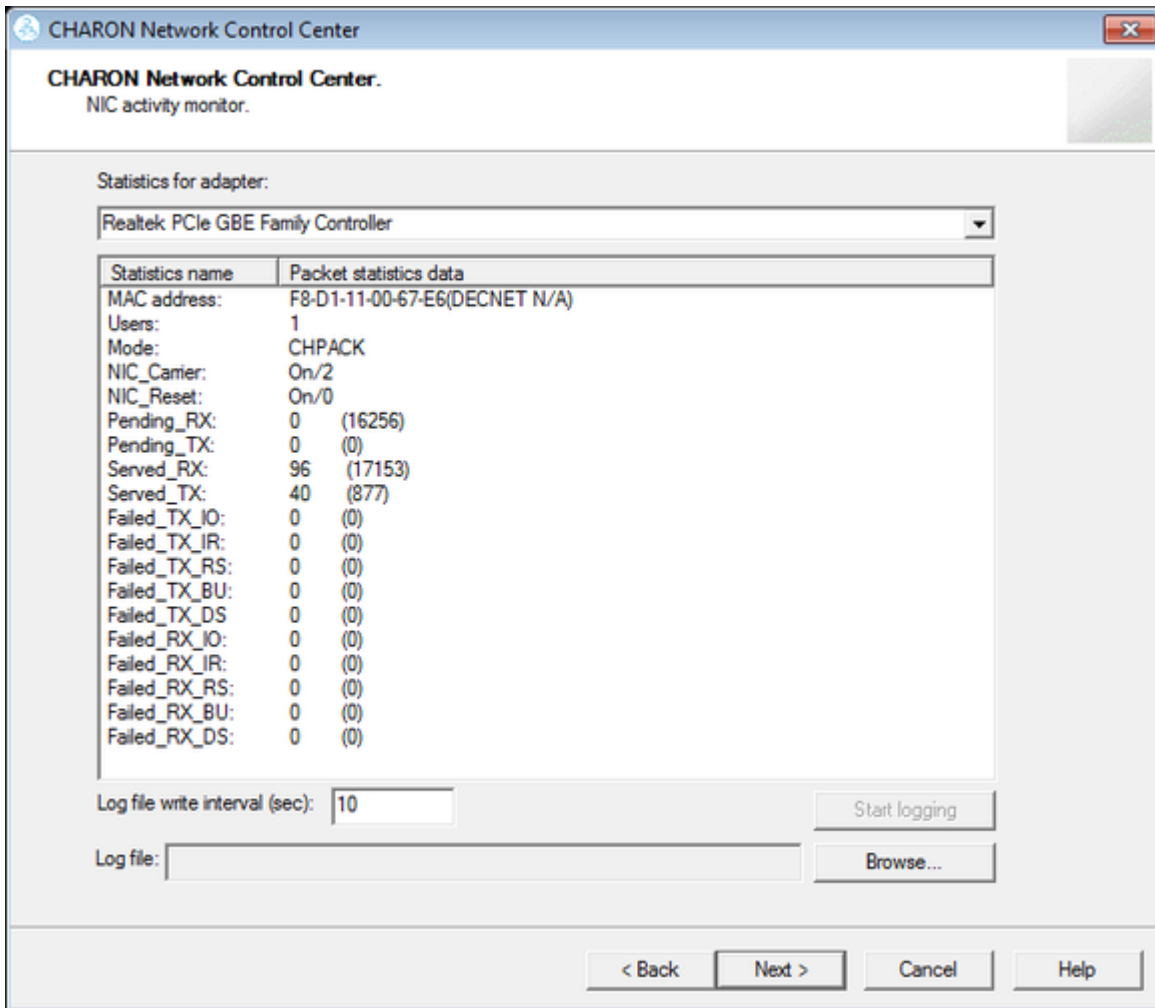
Press the "Next" button to stop the recording and to see the log of this operation:
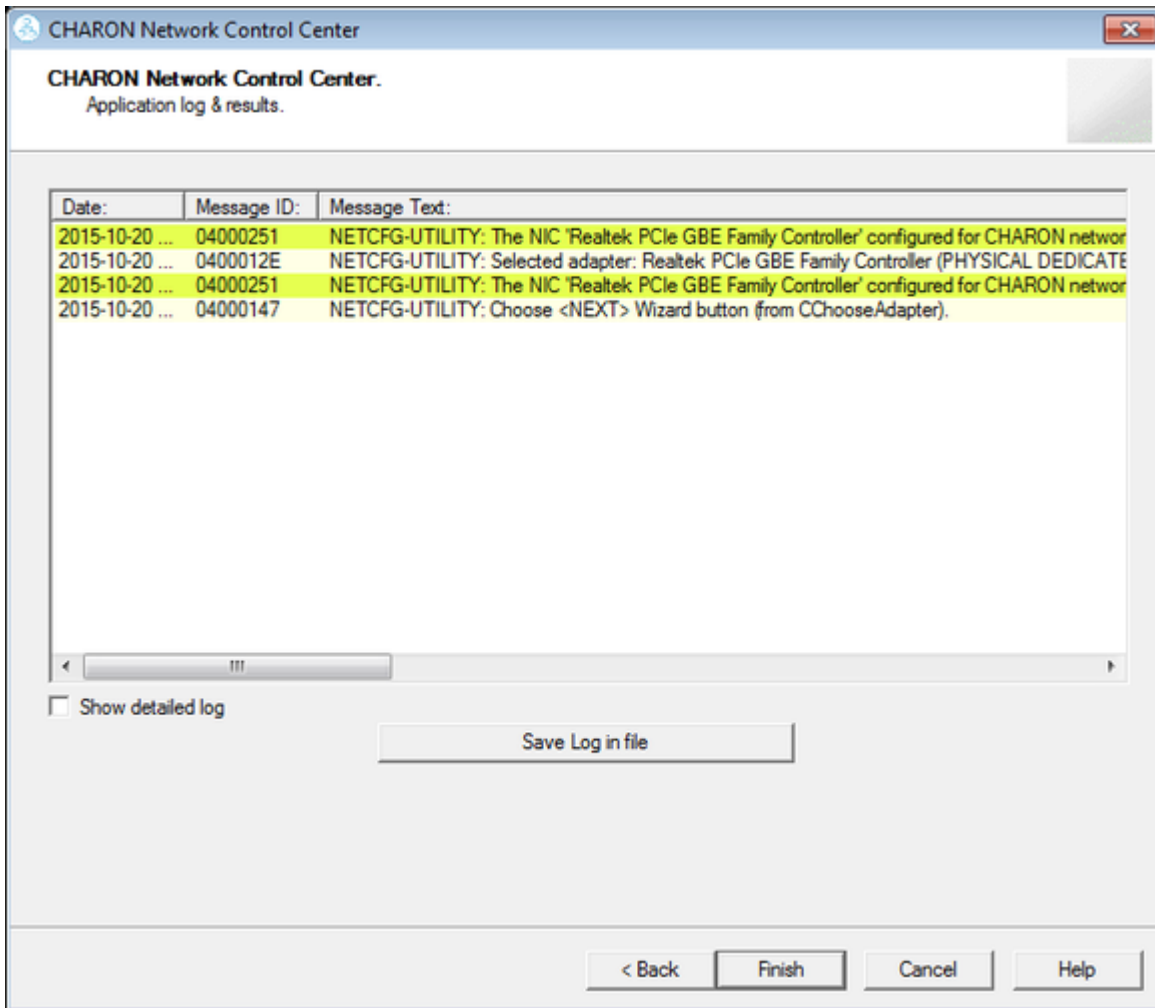
Select "Show detailed log" for more detail.

Press the "Finish" button to exit.

MkDisk

Table of Contents

General Description

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

Open the CHARON Virtual Machines Manager, switch to the "Host Information & Utilities" tab and press the "Virtual Disk Tool" button:
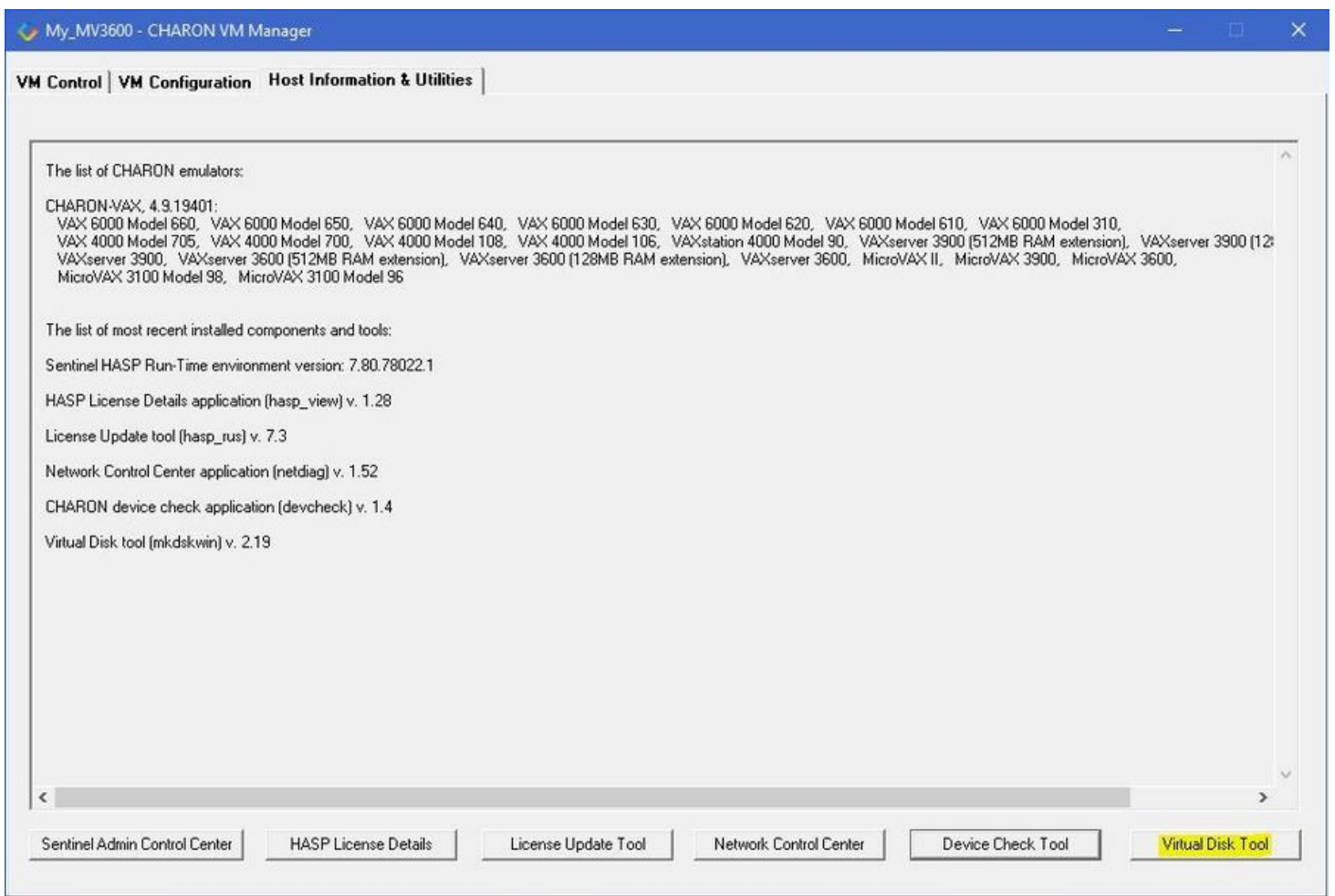
The main dialog of the utility will appear:

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:
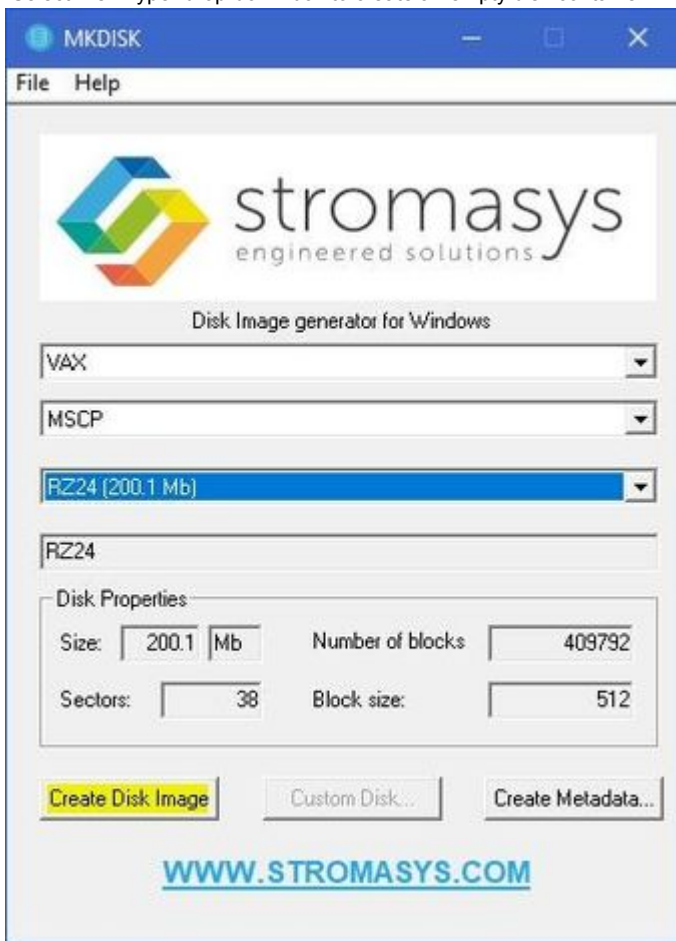
Press the "Create Disk Image" button to proceed. A dialog asking to specify the name of the disk image will appear:
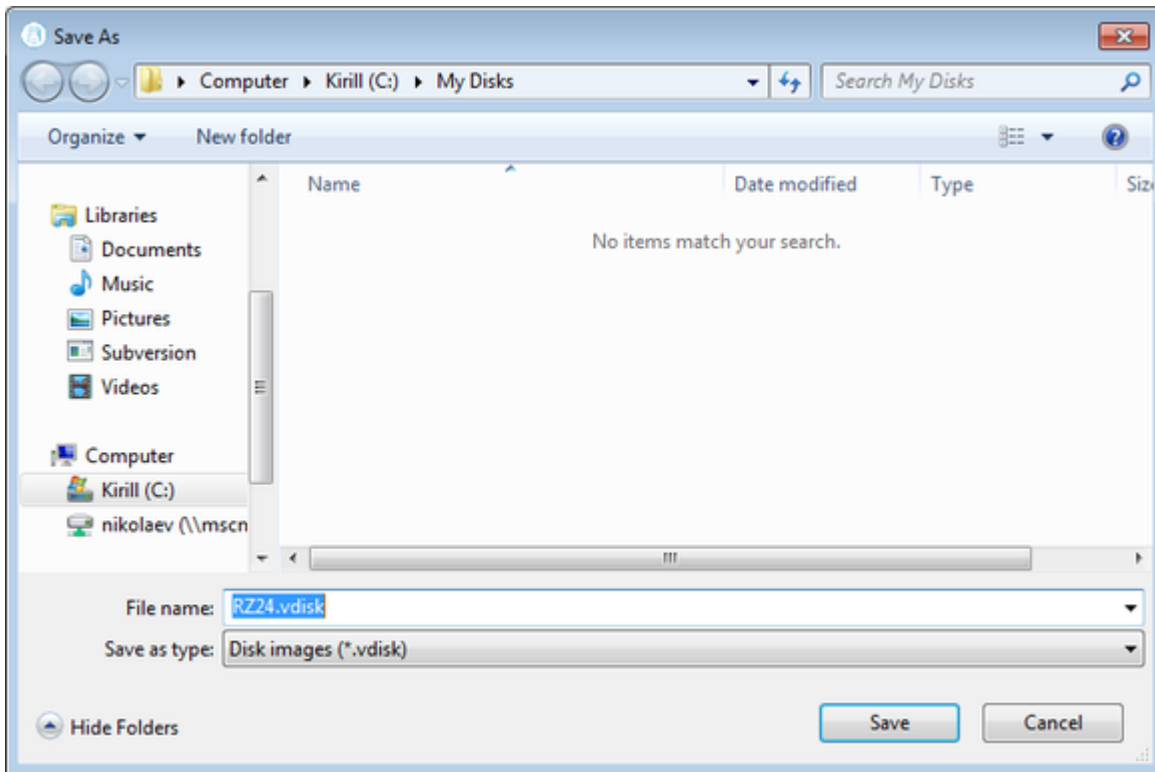
Browse to the target directory, specify the name of the disk image and press the "Save" button. The process of creating the disk container will start:
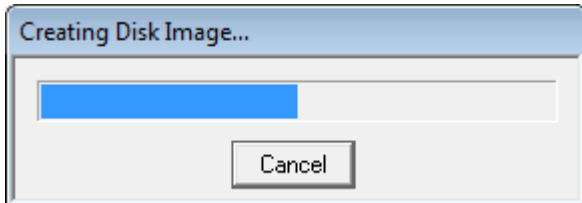
The target disk image is created - along with its metadata file (see below) once the process completes.

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.
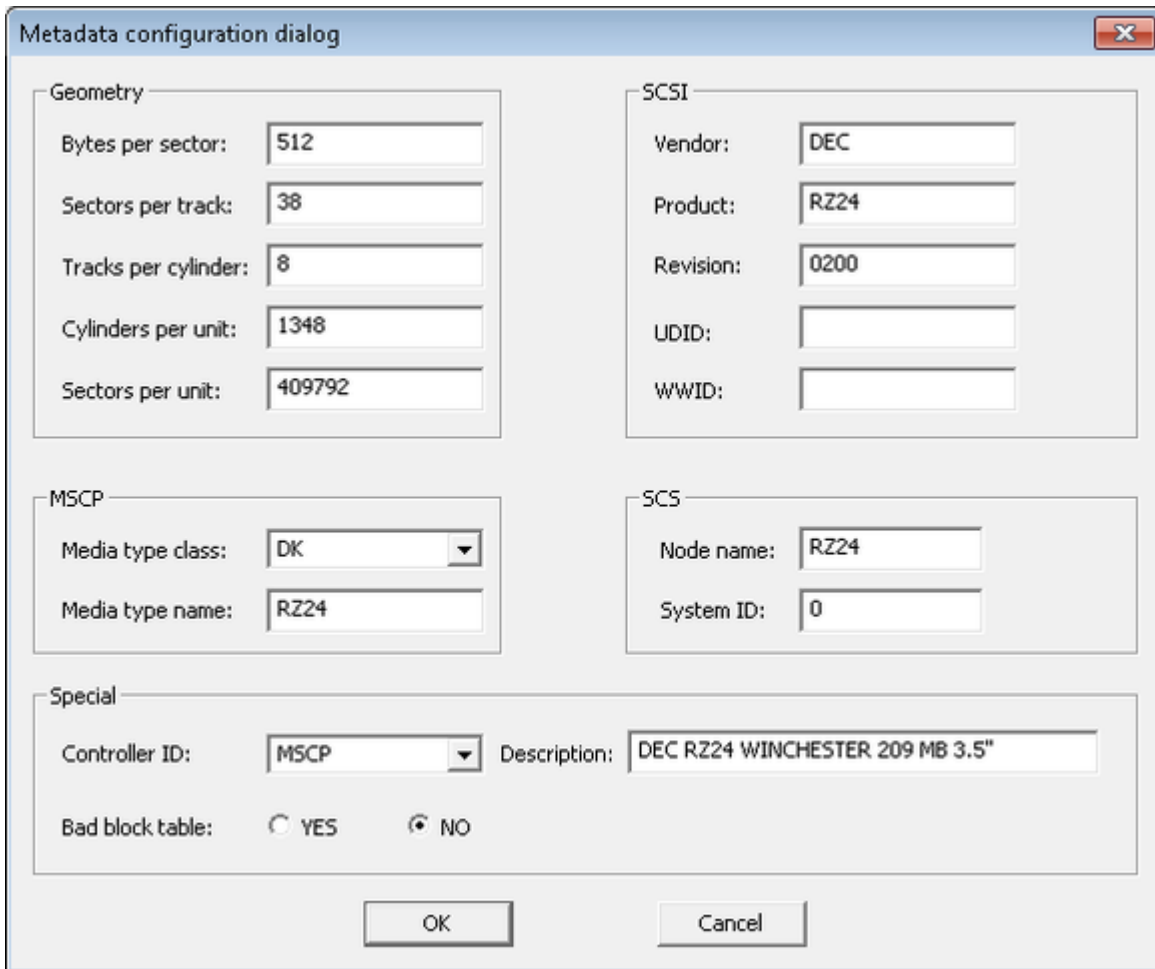
To create a metatada file press the "Create Metadata..." button:
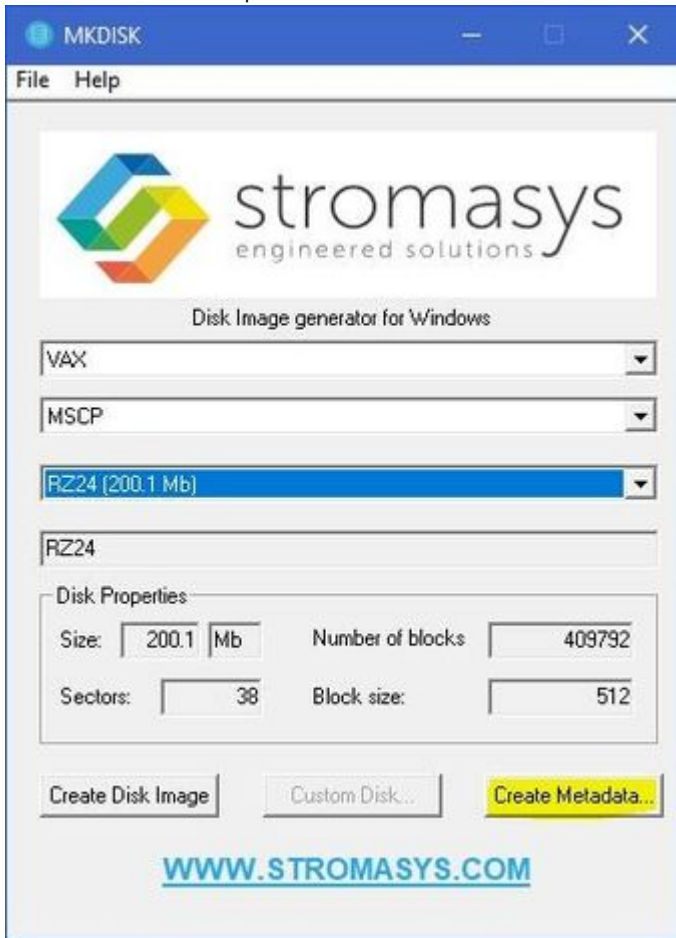
The "MkDisk" utility will ask you to select a disk container for which to create the metadata file:
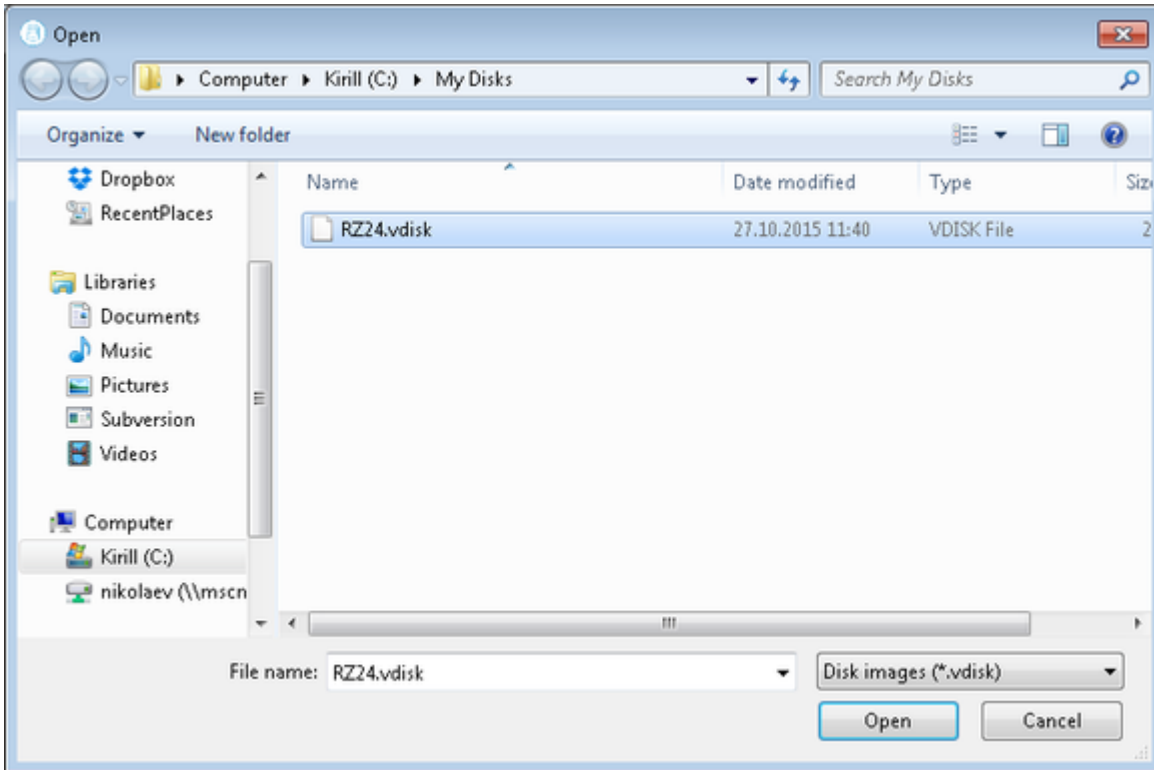
Select the target disk image and press the "Open" button. The utility will display a dialog for adjusting/correcting the disk image parameters:

Metadata configuration dialog	✕
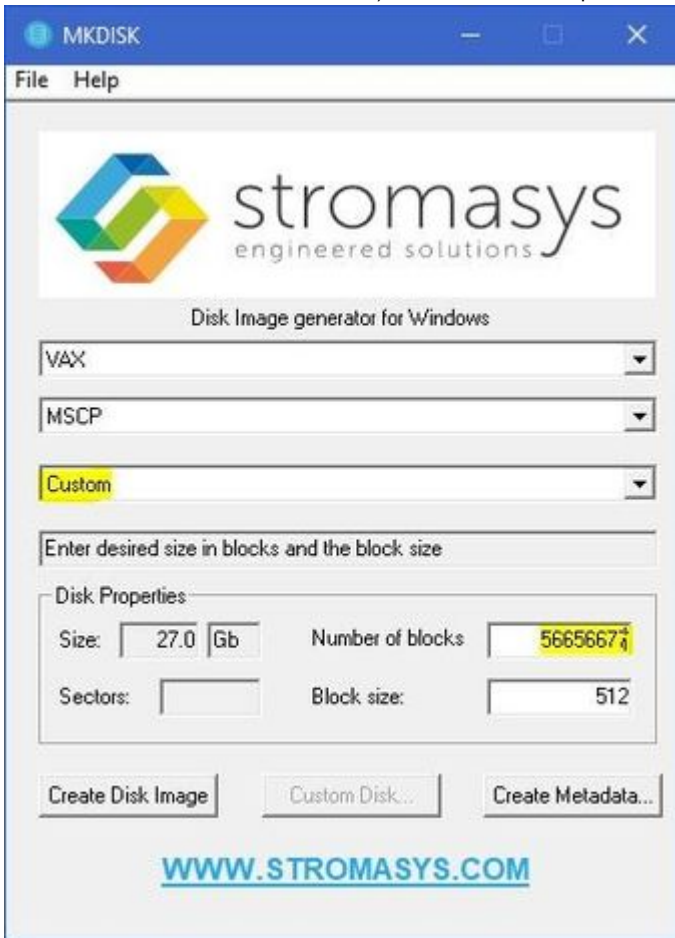
Geometry

Bytes per sector:	512
Sectors per track:	38
Tracks per cylinder:	8
Cylinders per unit:	1348
Sectors per unit:	409792

SCSI

Vendor:	DEC
Product:	RZ24
Revision:	0200
UDID:	
WWID:	

MSCP

Media type class:	DK ▼
Media type name:	RZ24

SCS

Node name:	RZ24
System ID:	0

Special

Controller ID:	MSCP ▼
Description:	DEC RZ24 WINCHESTER 209 MB 3.5"
Bad block table:	○ YES ⦿ NO

[OK]   [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:

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

Getting information about available disk types

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

Device	Description	Platform	Size (Kb)
RZ22	DEC RZ22 WINCHESTER 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

OK

MkDskCmd

Table of Contents

General Description

The "`MkDskCmd`" utility:

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

To start the utility open "`cmd.exe`" 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:

```
...> mkdskcmd -help

mkdisk for CHARON utility v. 1.17
Copyright: STROMASYS, 2017

Usage:
      mkdskcmd [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:
mkdskcmd -help
mkdskcmd -list
mkdskcmd -avtable mkdsk.vtable -output rk07.vdisk -disk rk07
mkdskcmd -output disks\custom.vdisk -blsize 512 -blcount 16384
mkdskcmd -r E:\disks\rz22.vdisk rz59 -a "C:\Program Files\CHARON\disks\mkdsk.vtable"
```

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

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

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

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

Open the CHARON Virtual Machines Manager, switch to the "Host Information & Utilities" tab and press the "HASP License Details" button:
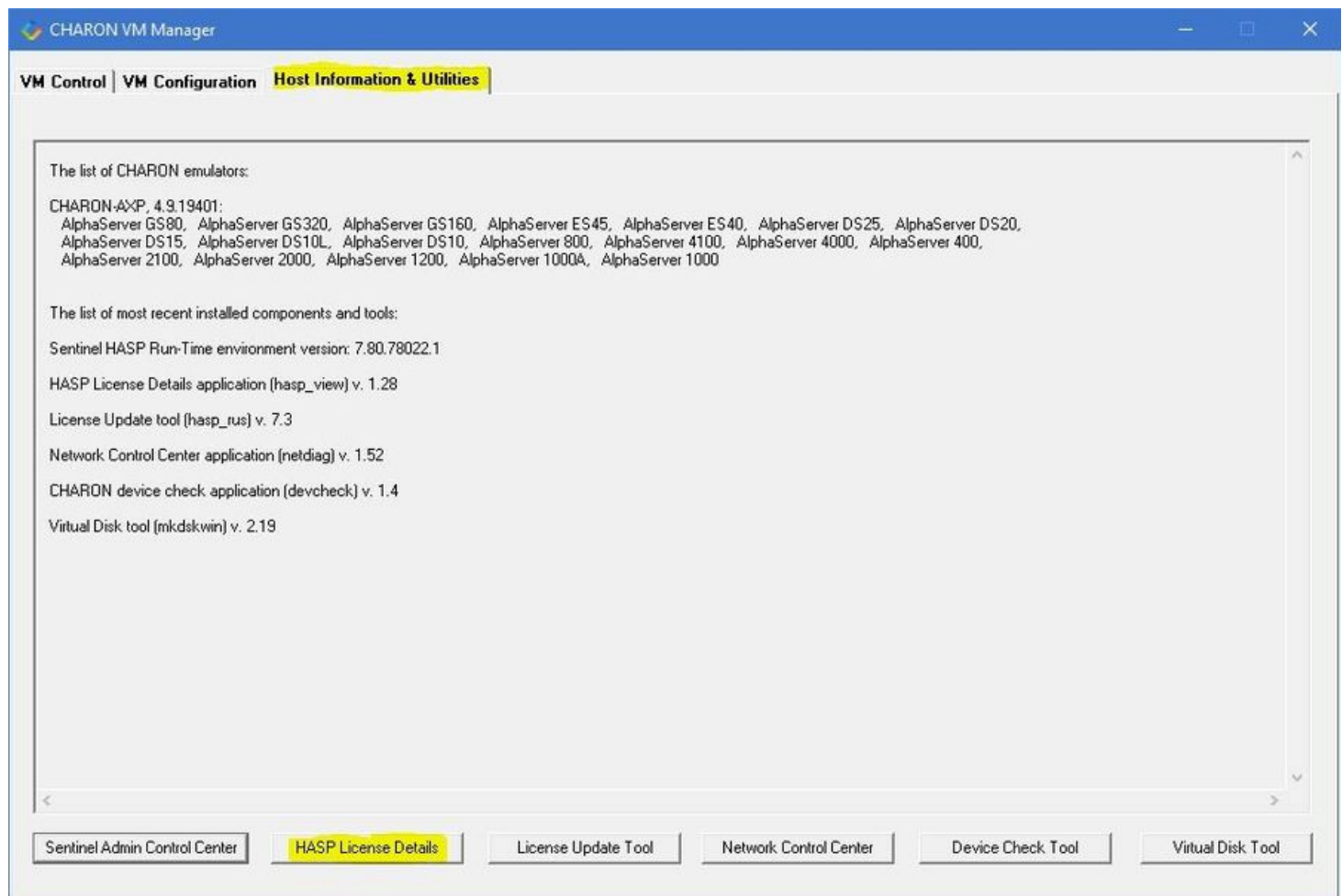
Getting CHARON licenses content

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

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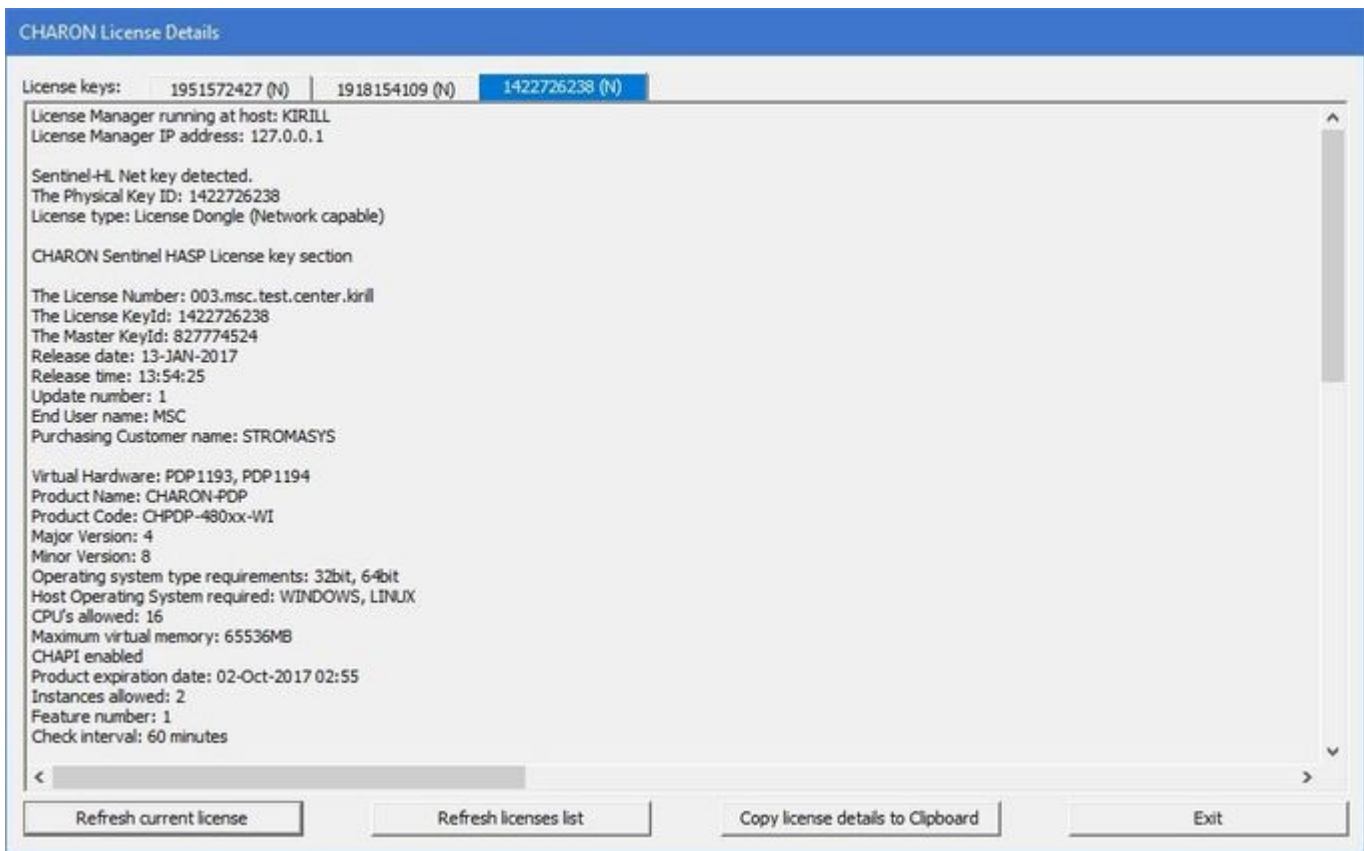
Use the "License keys" tabs to view the content of the CHARON licenses found.

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

The "License Update Service" utility is used to manage CHARON licenses, collect the host system fingerprint.

Open the CHARON Virtual Machines Manager, switch to the "Host Information & Utilities" tab and press the "License Update Tool" button:
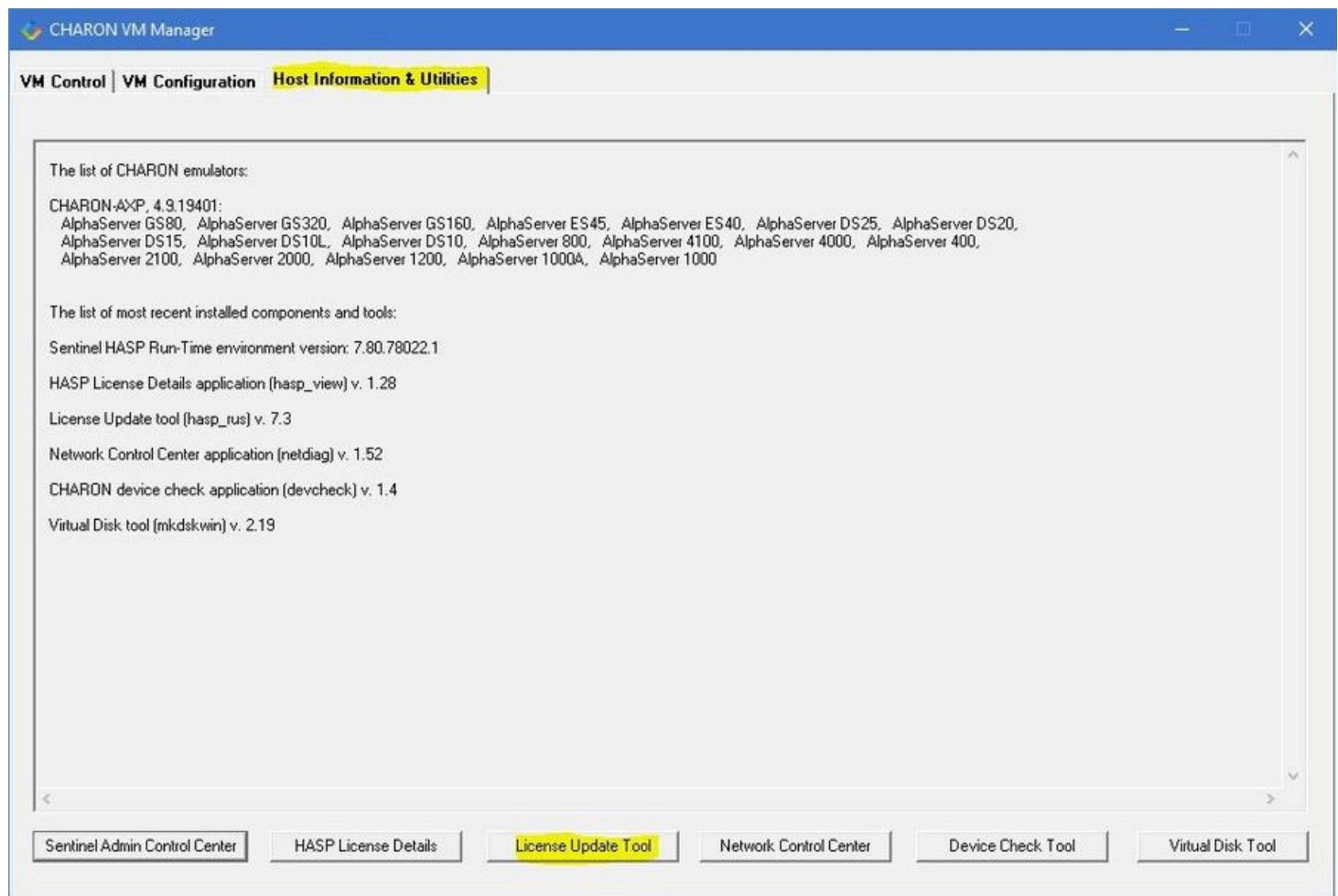
Collecting the host fingerprint and information on an existing license

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.
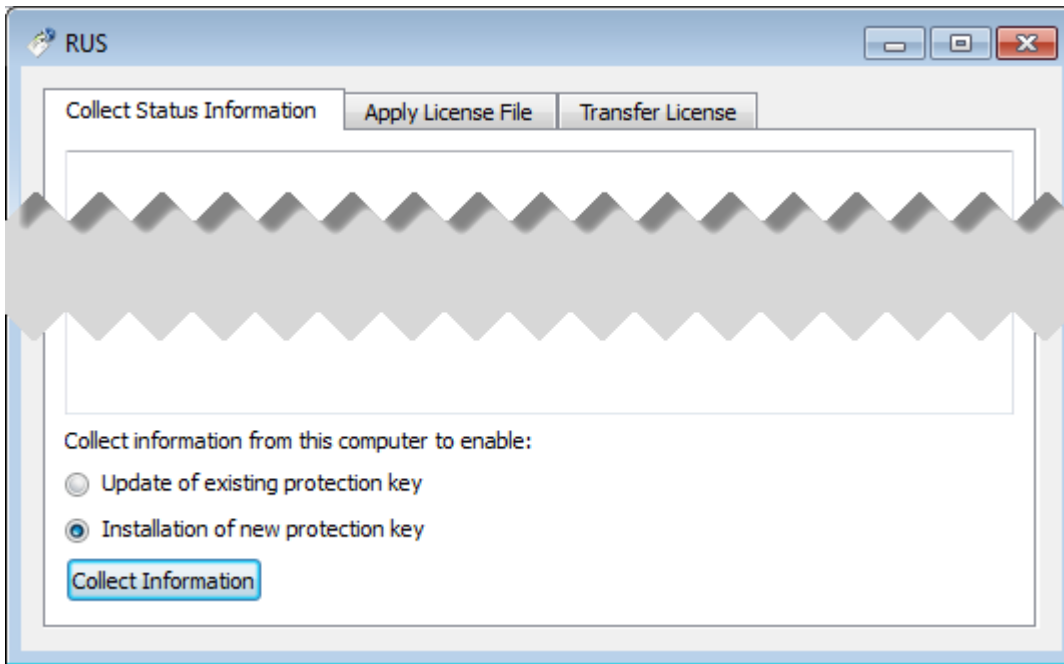
Press the "Collect Information" button and save the fingerprint or the information on the current license to a "*.c2v" file:
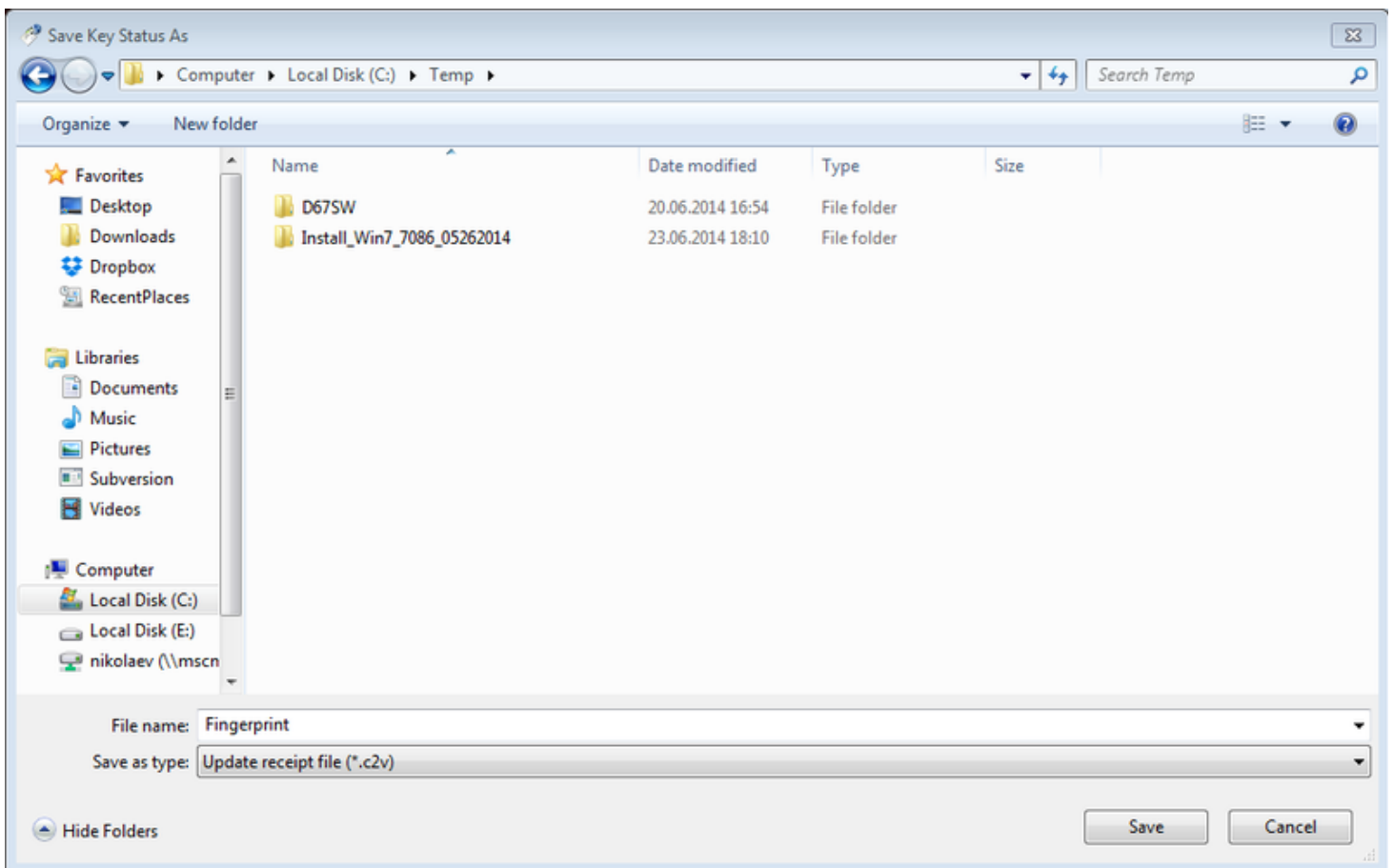
Press the "Save" button to create the "*.c2v" file and send this file to STROMASYS to receive an update of the current license or a new license.

Installation and update of a CHARON license

Open the "Apply License File" tab and press the "..." button:
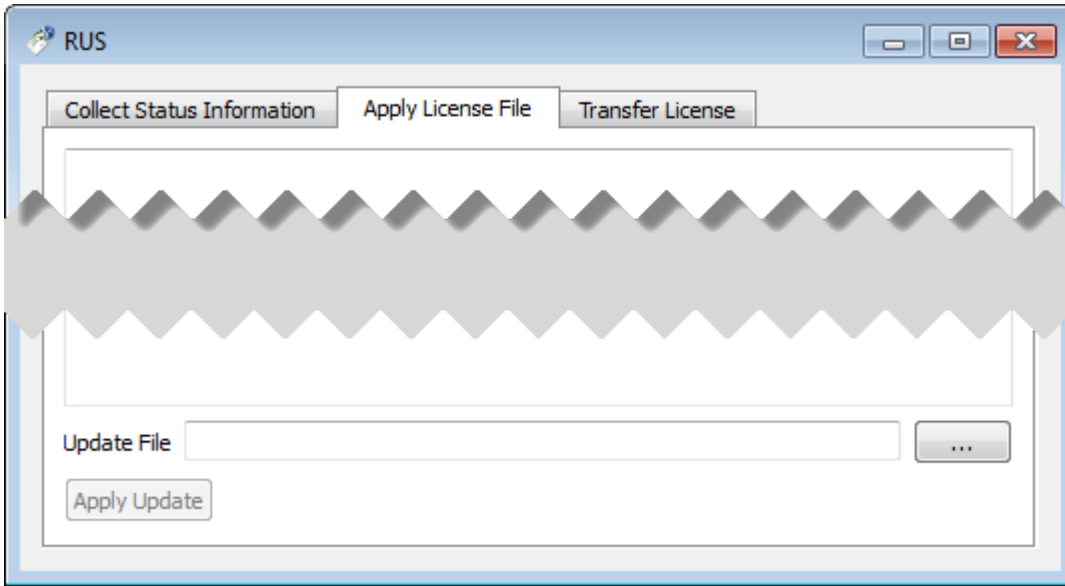
Select the license "*.v2c" file received from STROMASYS:
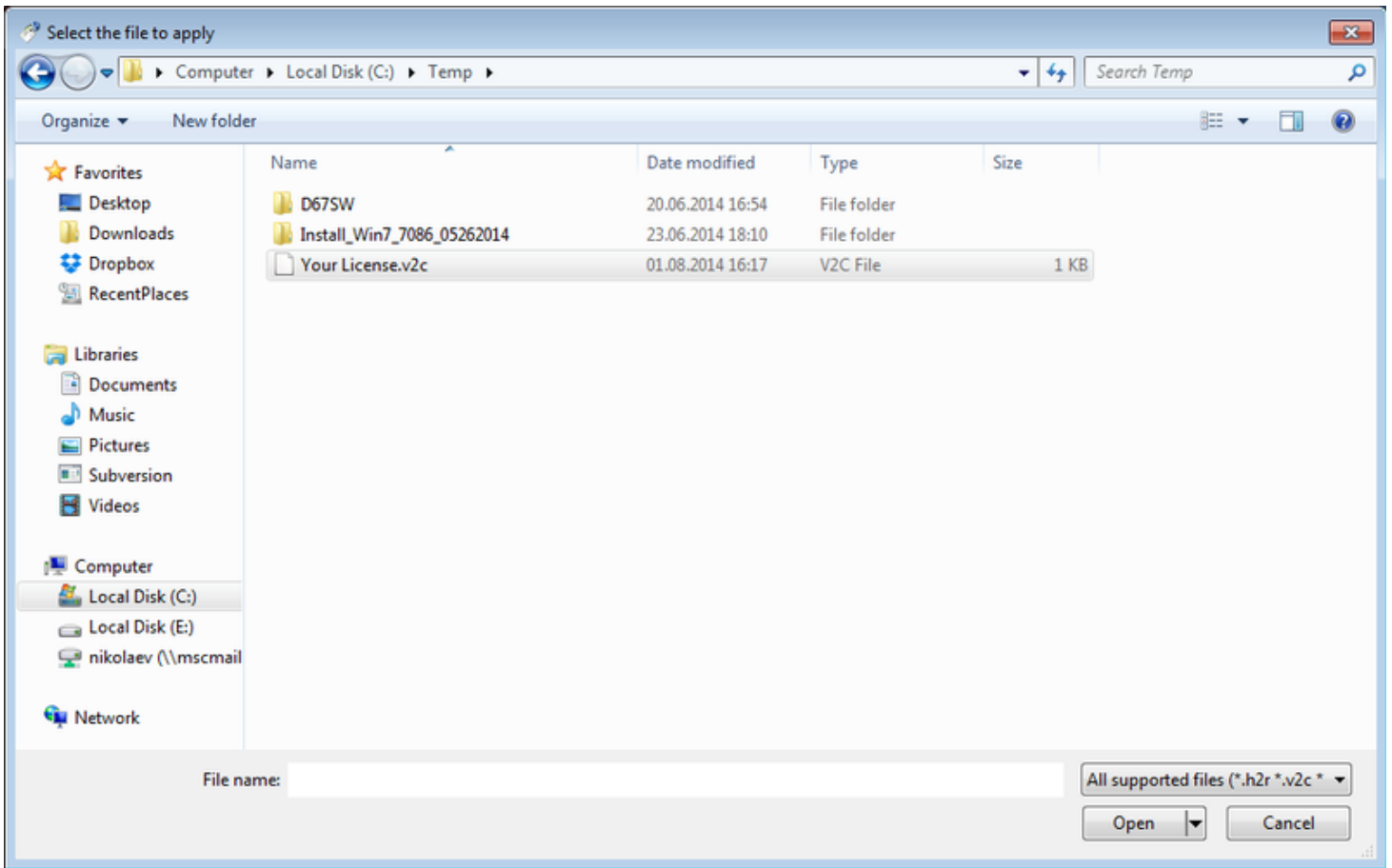
Press the "Open" button and then the "Apply Update" button in the main dialog box to apply the new license or to update an existing one.

License expiration check

Introduction

The license expiration check utility is provided with the CHARON-VAX V4.9 kit, 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.

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.

Related products	CHARON-AXP (≥ V4.5) and CHARON-VAX (≥ V4.5)
Operating systems	Windows 7, Windows 8.1, Windows Server 2008 R2, Windows Server 2012 (R2), Windows Server 2016

Table of contents

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 version 1.9.

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 are defined as follows:

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

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.
  - ℹ 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"
```

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 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**.
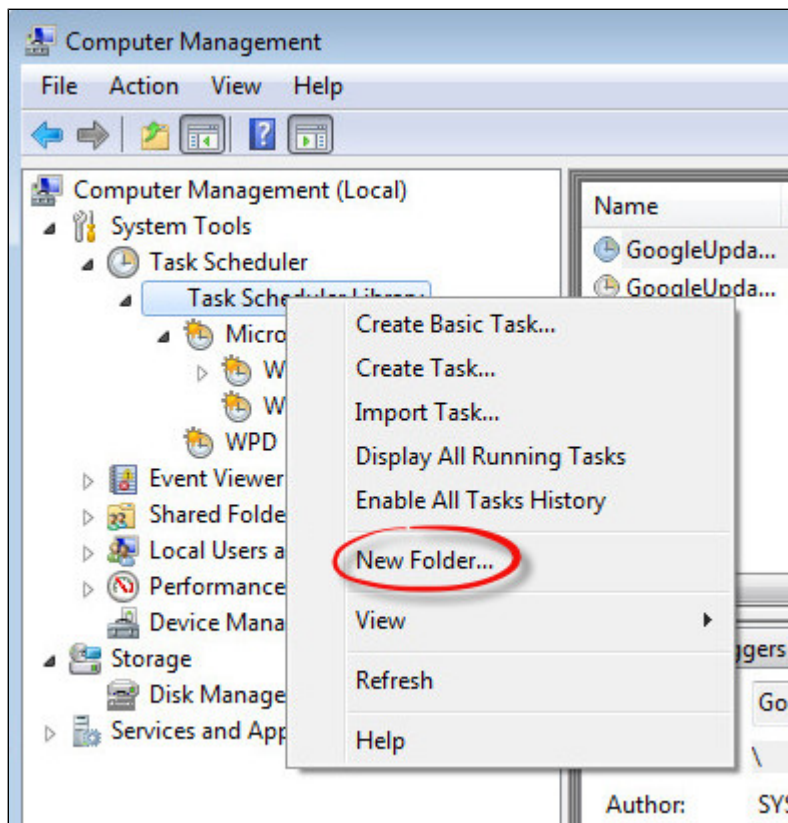
Select the newly created folder (**1**) and on the right pane, select "**Create Task...**" (**2**).

⚠ Do not use "Create Basic Task..." some of the options required aren't available in that wizard.
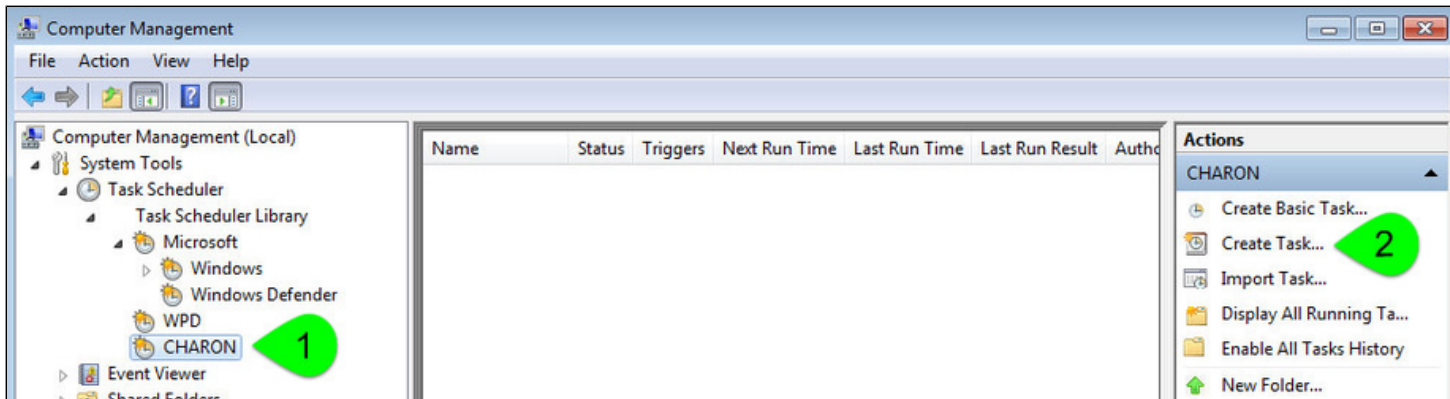
Fill the **General** tab as below:
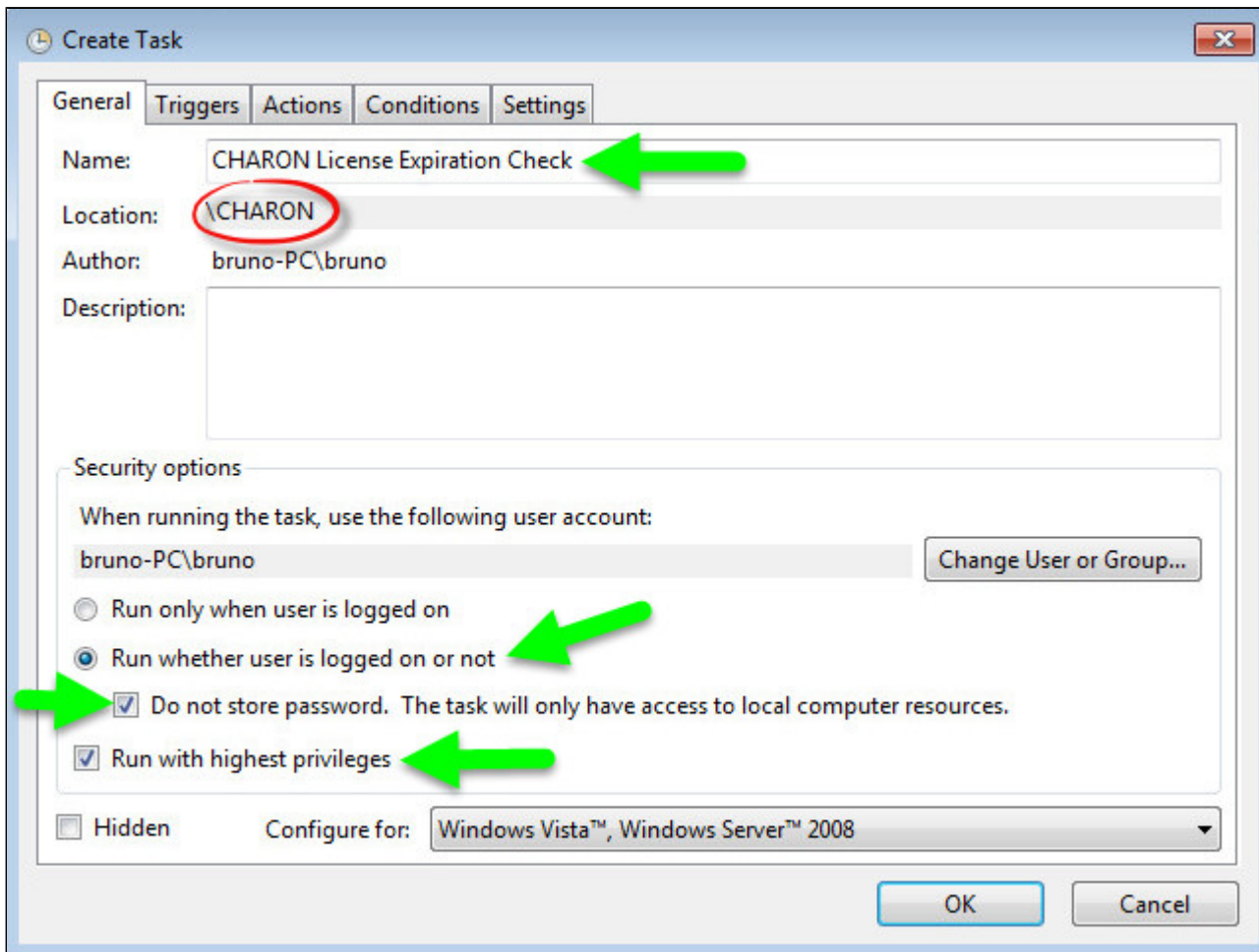
Select the **Triggers** tab and set the check interval, for example here: everyday at 09:00 AM

Select the **Actions** tab and configure powershell to be executed with the following parameters:
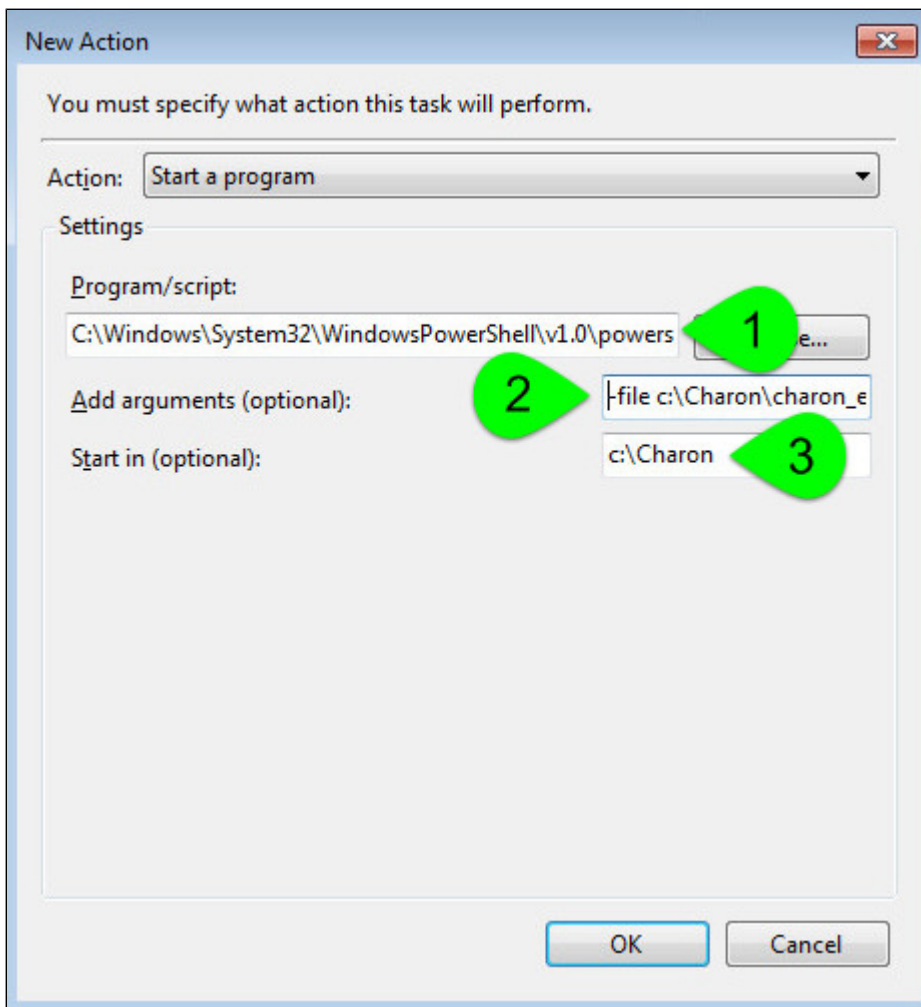
(**1**) : Powershell location here is "`C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe`"

(**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 "`c haron_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" ^

              /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:
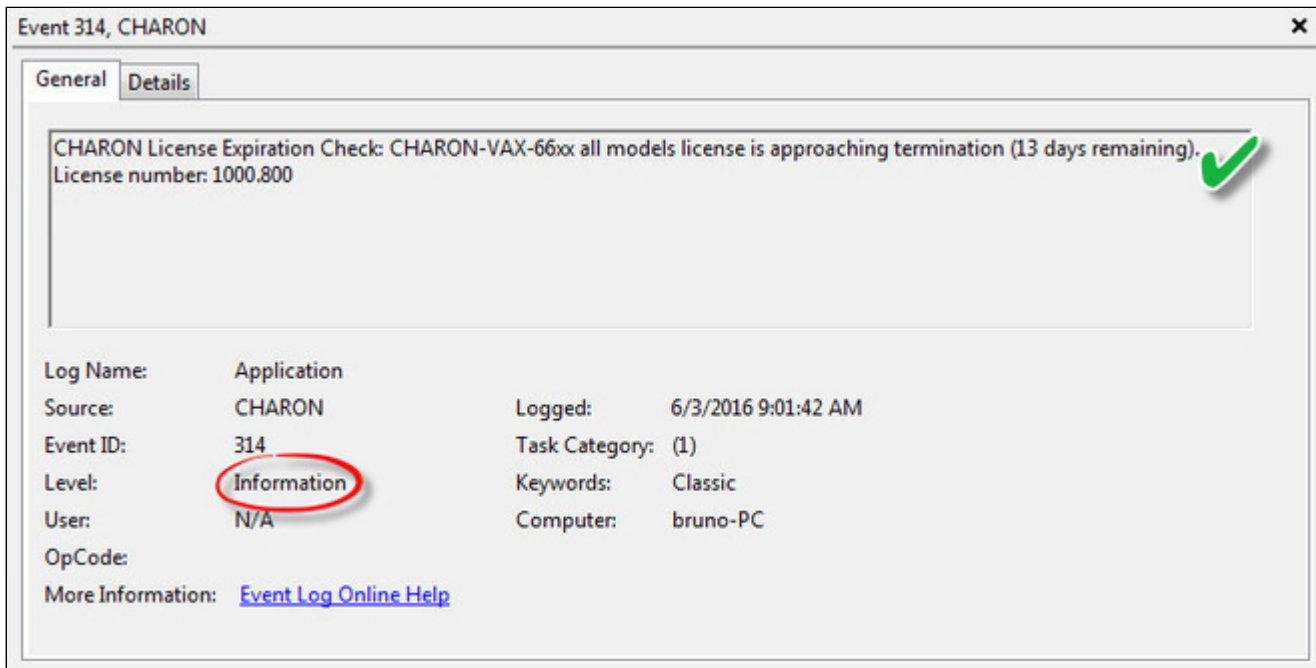
```
c:\Charon>schtasks /create /SC DAILY /ST 09:00 /NP /RL HIGHEST /TN "CHARON\CHARON License Expiration Check" ^

          /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:

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	Logged:	4/28/2016 3:46:17 PM
Event ID:	314	Task Category:	(1)
Level:	Error	Keywords:	Classic
User:	N/A	Computer:	bruno-PC
OpCode:	
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.

ℹ 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-clixml C:\Charon\creds.clixml

#--- 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" )
```

ℹ 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
```

ℹ This .bat file can be found in the kit downloaded previously.

<div style="background-color:green; color:white; padding:5px">**Scheduled task settings update**</div>

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:
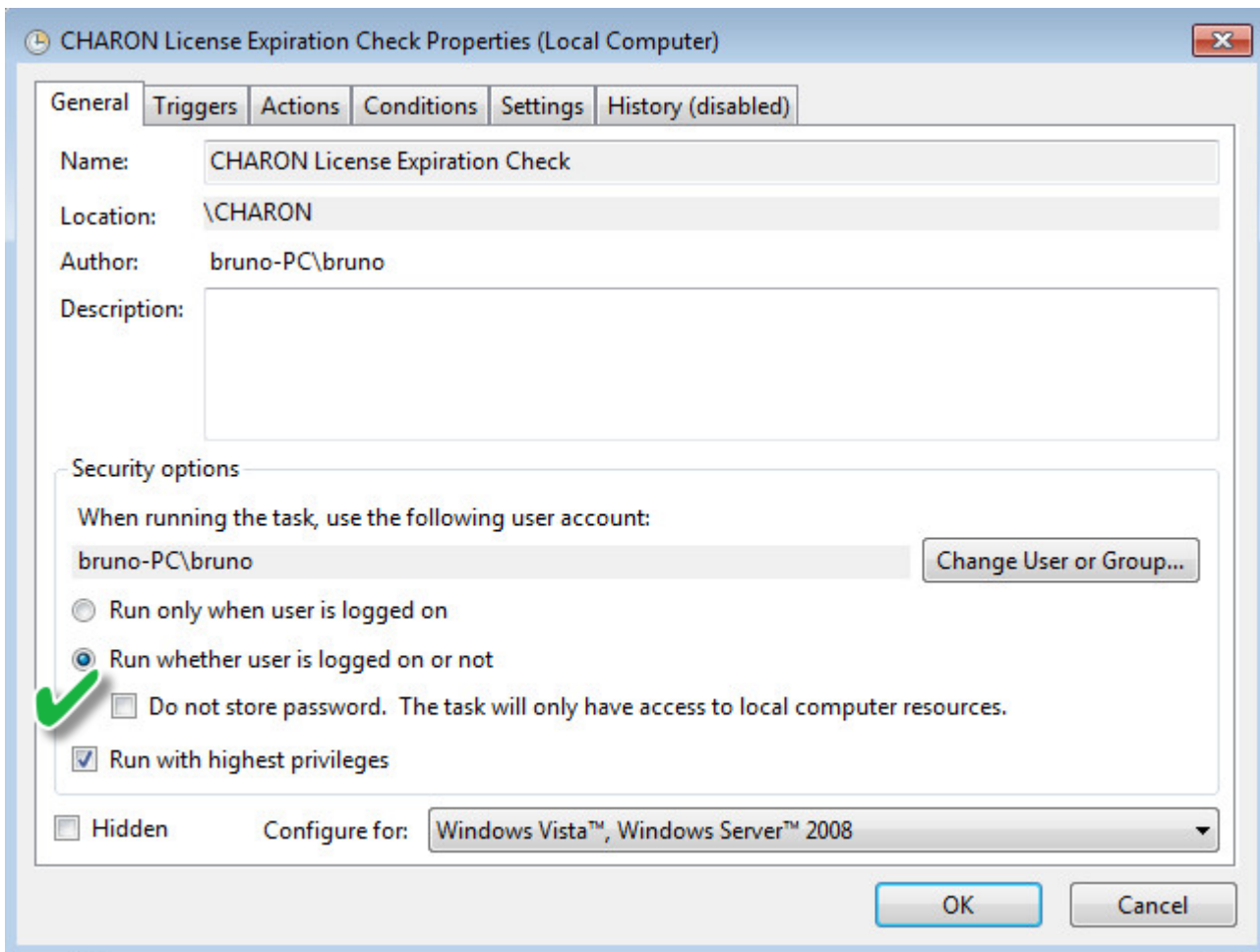
You will be prompted to store the password of the author.

This can also be done using the command line:

```
c:\Charon>schtasks /change /TN "CHARON\CHARON License Expiration Check" /rp <password>
```

<div style="background-color:teal; color:white; padding:5px">**Related articles**</div>

- Charon Log monitoring on Windows (logmond) - Best practices for V4.9 and V4.10
- CHARON-AXP and CHARON-VAX on Windows fail to start with digital signature error
- CHARON on Windows - Automated License Expiration Check
- CHARON on Windows - Automated License Expiration Check - Release Notes
- How-to install Charon license expiration check on a Linux server

mtd

Table of Contents

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")
`-s`	Write tape image in SMA format

-g	Gather statistics and print upon completion
-a	Do not print logo

Examples:

```
...> mtd -l tape1.txt -r 10 \\.\Tape0 "C:\Charon\Tapes\tape1.vtape"
```

```
...> mtd "C:\Charon\Tapes\tape1.vtape" \\.\Tape0
```

Tape container formats transfer

Use the following syntax to transfer the CHARON-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-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

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.

i 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.XXXXX\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

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.
	
	 Example 1:
	
	 `-printer=[\\.\Microsoft Office Document Image Writer]`
	
	 Example 2:
	
	 `-printer=[\\print_server\MSCLPS]`

-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

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

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 "`charoncp012.vdisk`" disk file in the "`C:\Program Files\CHARON\Virtual Disk Images\Tape utilities package`".

Supported OpenVMS versions: OpenVMS 6.1 and above.

ℹ 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.CHARONCP012.KIT]

Enter the products to be processed from the first distribution volume set.
* Products: CHARONCP012
```

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$STARTUP:SYSTARTUP_VMS.COM" ("SYS$STARTUP:SYSTARTUP_V5.COM" for VMS 5.5) file for the package to be loaded automatically at system startup::

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

After that the package will be loaded automatically on startup.

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.

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

Adjusting emulator's speed

CHARON speed can be adjusted using these commands:

Command	Description
$ **CHARONCP SET SPEED /UP=**<number of steps>	Increase the emulator speed by the specified number of steps.
$ **CHARONCP SET SPEED /DOWN=**<number of steps>	Decrease the emulator speed by the specified number of steps.
$ **CHARONCP SET SPEED /RESET**	Remove all speed stepping restraints.

Emulator's shutdown control

It is possible to schedule CHARON to shutdown directly from OpenVMS:

Command	Description

$ **CHARONCP SET SHUTDOWN /IN=**<*seconds, 60 is default*>	The number of seconds before the CHARON emulator is to shut down.
	The value can be between 0 and 65535 (approx. 18 hours).
$ **CHARONCP SET SHUTDOWN /RESET**	Clear a scheduled shutdown.

<div style="background-color:green; color:white">Virtual tapes management</div>

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"

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:

Command	Description
$ **CHARONCP SET MAGTAPE** <*device*> **/LOAD="**<*filename*>.**vtape"**	Create the specified host-file (if it does not already exist) and attach it to the specified virtual tape device. **Example**: `$ CHARONCP SET MAGTAPE MKA600: /LOAD="backup_01.vtape"`
$ **CHARONCP SET MAGTAPE** <*device*> **/UNLOAD**	Detach any file currently attached to te specified virtual tape device. **Example**: `$ CHARONCP SET MAGTAPE MKA600: /UNLOAD`

Possible errors:

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 fileextension of ".vtape" and optional directory).

ⓘ If some tape container has been already specified in the CHARON configuratioon 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 functionalilty while you are in CHARONCP:

Command	Description

`$ CHARONCP`
`CHARONCP> DEFINE /KEY <key-name> <equivalence-string>`
	Defines an equivalence string and a set of attributes with a key on the terminal keyboard.
	
	You can have a set of keys defined automatically for use with the CHARONCP utility by placing DEFINE/KEY commands in the file SYS$LOGIN:CHARONCP_KEYDEFS.INI
	
	Example:
	
	`$ CHARONCP`
	`CHARONCP> DEFINE /KEY F1 "SET MAGTAPE MKA600: /UNLOAD"`

`$ CHARONCP`
`CHARONCP> SHOW KEY <key-name>`
	Displays key definitions created with the DEFINE/KEY command. 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

Command	Description

`$ CHARONCP SHOW VERSION`	Displays the CHARONCP package version number and architecture. This can be useful for customers reporting issues with the CHARONCP software.
	
	Example:
	
	`$ CHARONCP SHOW VERSION`
	
	`CHARONCP version id is: V1.2`

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
- 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

General Settings

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

Parameter	hw_model
Type	Text string
Value	Virtual VAX/PDP11 system hardware model to be emulated. Use a default configuration template for each particular model as a starting point for a custom configuration. This would ensure that the parameter is set correctly. **Example**: `set session hw_model="VAX_6610"` Available models are: - PDP_1193 - PDP_1194 - MicroVAX_3100_Model_96 - MicroVAX_3100_Model_98 - MicroVAX_3600 - MicroVAX_3900 - MicroVAX_II - VAXserver_3600 - VAXserver_3600 - VAXserver_3600_128 - VAXserver_3600_512 - VAXserver_3900 - VAXserver_3900_128 - VAXserver_3900_512 - VAX_4000_Model_106 - VAX_4000_Model_108 - VAX_4000_Model_700 - VAX_4000_Model_705 - VAX_6000_Model_310 - VAXstation_4000_Model_90 - VAX_6610 - VAX_6620 - VAX_6630 - VAX_6640 - VAX_6650 - VAX_6660

configuration_name

Parameter	configuration_name
Type	Text string

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

log	
Parameter	log
Type	Text string
Value	The log file or directory name is where the log file for each CHARON-VAX / CHARON-PDP execution session is stored.

Log specified as a file name

It is possible to overwrite the existing log file or to extend it using the "General Settings#log_method" parameter.

ℹ The "log_method" parameter is effective only when a single log file is specified, not a directory.

Example:

```
set session log="C:\Charon\vax4106prod.log"
```

Log specified as a directory

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 General Settings#log_file_size) and/or when the log file is older than a specified number of days (see General Settings#log_rotation_period).

ℹ 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.

If the "configuration_name" parameter of the session is specified, the log file name is composed as follows:

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

If the "configuration_name" parameter is omitted, the log file name will have the following format:

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

where "xxxxxxxxx" is an increasing decimal integer, starting from 000000000 to separate log files with the same time of creation (in case the log is being created faster than one log file per second).

⚠ Only existing directory can be specified. If the directory specified does not exist, this will be considered as a flat file. No trailing backslash character is allowed.

Example:

```
set session configuration_name="vax4106prod"

set session log="C:\Charon\Logs"
```

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.

log_method

Parameter	log_method
Type	Text string
Value	• "append" (default) • "overwrite" Determines if the previous log information is maintained or overwritten. > This parameter must be specified only in addition to the "log" parameter on the same line. 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. **Example**: `set session log="log.txt" log_method="append"`

log_show_messages

Parameter	log_show_messages
Type	Text string
Value	• "all" (default) • "info" • "warning" • "error" Defines the message types to be shown. The parameter is a string of comma delimited words. **Example**: `set session log_show_messages="error, warning"`

log_file_size

Parameter	log_file_size
Type	Text string

Value	If log rotation is enabled, the log_file_size parameter determines the log file size threshold at which the log is automatically rotated.
	
	• "unlimited" or "0" (default) - the feature is disabled
	• "default" - default size is used (4Mb)
	• <size>[KMG] - size of the current log file in bytes with additional multipliers:
	  • K - Kilobyte - multiply by 1024
	  • M - Megabyte - multiply by 1024*1024
	  • G - Gigabyte - multiply by 1024*1024*1024
	
	Examples:
	
	`set session log_file_size="default"`
	
	`set session log_file_size=10M`
	
	⚠ Minimum LOG File size is 64K, maximum is 1G. Setting a size less then 64K effectively makes the log file size unlimited.

log_rotation_period

Parameter	log_rotation_period
Type	Text string

Value	"default" - default value, 7 days. This value is used even if the "log_rotation_period" is not specified. "daily" or "1" "weekly" or "7" "never" or "0" <N> - in N days 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.
	
	Examples:
	
	`set session log_rotation_period="weekly"`
	
	`set session log_rotation_period=14`

log_repeat_filter

Parameter	log_repeat_filter
Type	Text string
Value	Specifies if repeated messages should be filtered or not. Possible values are "on" and "off" (default). 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 *N* times." and the counter is cleared. ⚠️ Stromasys highly recommends to let this value to "off" otherwise important information will not be reported in the log file. **Example**: `set session log_repeat_filter="off"`

license_key_id

Parameter	license_key_id
Type	Text string
Value	A set of Sentinel Key IDs that specifies the license keys to be used by CHARON. It is also possibly to use a keyword **"any"** to force CHARON to look for a suitable license in all available keys if the license is not found in the specified keys. **Example**: ```set session license_key_id = "1877752571,354850588,any"``` Based on the presence of this parameter in the configuration file, CHARON behaves as follows: 1. **No keys are specified (the parameter is absent)** CHARON performs an unqualified search for any suitable key in unspecified order. If no key is found, CHARON exits. 2. **One or many keys are specified** 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). 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. 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.

license_id

Parameter	license_id
Type	Text string
Value	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. **Example**: ```set session license_id = "2718281828,314159265"``` 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.

affinity

Parameter	affinity
Type	Text string
Value	Overrides any initial process affinity mask provided by the host operating system. Once specified it binds the running instance of the emulator to particular host CPUs. Used for soft partitioning of the host CPU resources and/or for isolating host CPUs for other applications. By default the CHARON-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. 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). Host CPUs are enumerated as a comma separated list of host system assigned CPU numbers: `set session affinity="0, 2, 4, 6"`

n_of_io_cpus

Parameter	n_of_io_cpus
Type	Numeric
Value	This parameter specifies how many host CPUs CHARON-VAX / CHARON-PDP must use for I/O handling. Use of the "affinity" parameter may limit the number of CPUs available. 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. **Example**: `set session n_of_io_cpus=2`

license_key_lookup_retry

Parameter	license_key_lookup_retry
Type	Text String

Value	In case the CHARON-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.
	
	Syntax:
	
	`set session license_key_lookup_retry = "N [, T]"`
	
	Options:
	
	• N - Number of retries to look for license keys.
	• T - Time between retries in seconds. If not specified, 60 seconds are used
	
	Example 1:
	
	`set session license_key_lookup_retry = 1`
	
	If the license key is not found during initial scan, only one more attempt after 60 seconds is performed.
	
	Example 2:
	
	`set session license_key_lookup_retry = "1,30"`
	
	Same as above but retry in 30 seconds.
	
	Example 3:
	
	`set session license_key_lookup_retry = "3,10"`
	
	If the license key is not found during initial scan, do 3 more attempts waiting 10 seconds between them.
	
	Example 4:
	
	`set session license_key_lookup_retry = "5"`
	
	If the license key is not found during the initial scan, do 5 more attempts waiting 60 seconds between them.

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"
```

ℹ 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

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.

Parameter	ace_mode
Type	Boolean
Value	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
```

> "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

Parameter	size
Type	Numeric
Value	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

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

Parameter	container
Type	Text string
Value	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). By default it is left unspecified. ⓘ it is recommended to specify the full path to the TOY file.

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
```

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

Parameter	container
Type	Text string
Value	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. By default it is left unspecified. ℹ it is recommended to specify the full path to this file

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

Parameter	ext_rom
Type	Text string
Value	Specifies a file containing external boot code

ext_rom_address

Parameter	ext_rom_address
Type	Numeric
Value	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

Parameter	container
Type	Text string
Value	A string specifying a file name to store the content of the EEPROM. **Example**: ```set eeprom container="vx6k610.rom"``` 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. ℹ️ it is recommended to specify the full path to this file

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:

Console Parameter	halt
Description	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). The value is: • `"reboot"` **Example**: ```>>>set halt reboot```

> Please check that the "toy container" and "rom container" parameters are 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:

Parameter	bdr boot
Type	Text string
Value	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).The value is: • `"auto"` **Example**: ```set bdr boot=auto```

> Check that the "toy container" and "rom container" parameters are 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:

Parameters	cpu_0 auto_boot
Type	Text string
Value	Determines whether the PDP11/93 or PDP11/94 server boots automatically from the specified disk. The value is the disk drive to boot from. **Example**: `set cpu_0 auto_boot = "DU0"`

> Check that the "`toy container`" and "`rom container`" parameters are specified in the configuration file to store the boot flags.

VAX66x0

Parameter	xmi boot
Type	Text string
Value	Determines whether the CHARON VAX66x0 startup procedure stops at the ">>>" prompt after self-tests. The values are: • `"auto"` • `"manual"` (default) **Example**: `set xmi boot = "auto"` The value `"auto"` enables automatic boot from a default boot specification, previously configured in the VAX console. The value `"manual"` disables the automatic boot once the self tests are passed.

> Check that the "toy container" and "rom container" parameters are specified in the configuration file to store the boot flags.

Serial lines

Table of Contents

General Description

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:

```
load virtual_serial_line/chserial TTA1
set TTA1 port = 10003 application = "putty.exe -load TTA1_preset"
```

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 encoded in the "TTA1_preset" parameter of the "Putty" terminal emulator).
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.

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.

rts

Parameter	rts[<line>]
Type	Text string
Value	"On" - assert the RTS (Request To Send) signal "Off" - clear the RTS signal (default) "DTR" - assert the RTS signal as soon as the DTR signal is asserted (⚠ Applicable only for QUART)

dsr

Parameter	dsr[<line>]
Type	Text string
Value	"On" - always reports DSR signal assertion "Off" - always reports DSR signal deassertion "DSR" - use the DSR signal of the physical serial line (if configured) "CD", "DCD", "RLSD" - use the CD (carrier detect) signal of the physical serial line (if configured) This parameter is applicable only for line "2" of QUART. UART has no such parameter.

communication

Parameter	communication[<line>]
Type	Text string
Value	"ASCII" - for connection to terminals (default) "BINARY" - for serial lines carrying binary (packet) protocols, which are used mainly for communicating with PLCs

line

Parameter	line[<line>]
Type	Identifier
Value	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

Parameter	address
Type	Numeric
Value	Specifies CSR address. The address must be a valid QBUS 22-bit wide address in I/O space. 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. > This parameter is not applicable to DHW42-xx serial line controllers

vector

Parameter	vector
Type	Numeric
Value	Specifies the interrupt vector. The default value is 0300, which is the factory setting for asynchronous serial line multiplexers. > This parameter is not applicable to DHW42-xx serial line controllers

line

Parameter	line[N] N=0…3(7,15)
Type	Identifier
Value	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

Parameter	communication[N] N=0…4(7,15)
Type	Text String
Value	"ASCII" - for connection to terminals (default) "BINARY" - for serial lines carrying binary (packet) protocols, which are used mainly for communicating with PLCs

rts

Parameter	rts[N] N=0…3(7,15)
Type	Text String
Value	Controls the RTS signal of the Nth line of the multiplexer. "On" - assert the RTS (Request To Send) signal "Off" - clear the RTS signal (default) "DTR" - assert the RTS signal as soon as DTR signal is asserted When left blank (initial state), the level of the RTS signal is as requested by the VAX/PDP11 software.

dsr

Parameter	dsr[N] N=0…3(7,15)
Type	Text String
Value	"On" - always reports the DSR signal assertion "Off" - always reports the DSR signal deassertion "DSR" - use the DSR signal of the physical serial line (if configured) "CD", "DCD", "RLSD" - use the CD (carrier detect) signal of the physical serial line (if configured) This parameter is applicable only for the DZV11 and DZQ11 serial lines controllers

tx_q_max_depth

Parameter	tx_q_max_depth[N] N=0…3(7,15)
Type	Numeric
Value	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. This parameter is applicable only for the DHV11 serial lines controller

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

Parameter	line
Type	Text string
Description	A defined COM port on a host system in the form of "\\.\COMn" **Example**: ```set OPA0 line="\\.\COM1"```

BAUD

Parameter	baud
Type	Numeric
Description	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. **Example**: ```set OPA0 baud=38400```

BREAK_ON

Parameter	break_on
Type	Text string
Description	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). **Example**: ```set OPA0 break_on="Ctrl-P"``` The default value is "Break" for line 3 of QUART and "none" for other lines.

STOP_ON

Parameter	stop_on
Type	Text string
Description	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.
	⚠️ This parameter works only for the console line (for the one UART line and "line[3]" of QUART).
	Specify the one of the following values: "F6" or "none" ("none" disables triggering a STOP condition).
	Example:
	```set OPA0 stop_on="F6"```
	The default value is "none".
	Setting "F6" triggers the STOP condition upon receipt of the "`<ESC>[17~`" sequence. Terminals usually send these sequences by pressing the **F6** button

LOG

Parameter	log
Type	Text string
Description	A string specifying a file name to store the content of the console sessions or a directory where the log files for each individual session will be stored.
	If an existing directory is specified, CHARON-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.
	Examples:
	```set OPA0 log="log.txt"```
	```set OPA0 log="C:\Charon\Logs"```

LOG_FILE_SIZE

Parameter	log_file_size
Type	Text string
Description	If log rotation is enabled, the log_file_size parameter determines the log file size threshold at which the log is automatically rotated. • "unlimited" or "0" (default) - the feature is disabled • "default" - default size is used (4Mb) • \<size>[KMG] - size of the current log file in bytes with additional multipliers: • K - Kilobyte - multiply by 1024 • M - Megabyte - multiply by 1024*1024 • G - Gigabyte - multiply by 1024*1024*1024 **Examples**: ``` set OPA0 log_file_size="default" ``` ``` set OPA0 log_file_size=10M ``` ⚠ Minimum log file size is 64K, maximum is 1G. Setting size less then 64K effectively makes the log file unlimited.

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

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

PORT

Parameter	port
Type	Numeric
Description	The TCP/IP port number for the virtual serial line. A virtual serial line always listens on this port for incoming connection requests.

APPLICATION

Parameter	application
Type	Text string
Description	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. **Example**: ``` set OPA0 application = "putty.exe -load OPA0" ``` In this example the terminal emulator application: "putty" is started with the parameters "-load OPA0" telling it to load a specific saved session named "OPA0", (created separately) from the host registry. The "application" parameter is often combined with a "port" parameter: ``` set TTA1 port = 10003 application = "putty.exe -load TTA1" ```

BREAK_ON

Parameter	break_on
Type	Text string
Description	Specifies what byte sequences received over a virtual serial line triggers a HALT command. ⚠️ 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). Specify one of the following values: "Ctrl-P", "Break" or "none" to disable triggering a HALT condition. The commands are case insensitive. **Example**: ``` set OPA0 break_on="Ctrl-P" ``` The default value is "Break" for line 3 of QUART and "none" for other lines.

STOP_ON

Parameter	stop_on
Type	Text string
Description	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. ⚠️ 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). Specify one of the following values: "F6" or "none" ("none" disables triggering a STOP condition). The commands are case insensitive. **Example**: ``` set OPA0 stop_on="F6" ``` The default value is "none". Setting "**F6**" triggers the STOP condition upon receipt of the "`<ESC>[17~`" sequence.

LOG

Parameter	log
Type	Text string
Description	A string specifying the filename to store the content of the console sessions or a directory where log files for each individual session will be stored.

If an existing directory is specified, CHARON-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.

Examples:

```
set OPA0 log="log.txt"
```

```
set OPA0 log="C:\Charon\Logs"
``` |

LOG_FILE_SIZE

Parameter	log_file_size
Type	Text string
Description	If log rotation is enabled, the log_file_size parameter determines the log file size threshold at which the log is automatically rotated.

- "unlimited" or "0" (default) - the feature is disabled
- "default" - default size is used (4Mb)
- <size>[KMG] - size of the current log file in bytes with additional multipliers:
  - K - Kilobyte - multiply by 1024
  - M - Megabyte - multiply by 1024*1024
  - G - Gigabyte - multiply by 1024*1024*1024

Examples:

```
set OPA0 log_file_size="default"
```

```
set OPA0 log_file_size=10M
```

⚠ Minimum log file size is 64K, maximum is 1G. Setting size less then 64K effectively makes the log file unlimited. |

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
```

Disks and tapes

Contents

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 muti-volume backup for tape images. If some mutli-volume set (in form of tape images) has to be restored it is recommended to configure several tape drives in CHARON 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

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 "mkdskcmd" 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

Parameter	address
Type	Numeric
Value	Specifies the CSR address. This address must be a valid QBUS 22-bit address in IO space.
	Initially its value is 017772150, which is the factory setting for the RQDX3 disk controller.
	Use the "address" parameter for loading several instances of RQDX3. The "address" parameter value must be unique for every instance of the controller.

MAX_N_OF_UNITS

Parameter	max_n_of_units
Type	Numeric
Value	Specifies the maximum number of units supported by the controller. Possible values are 4…9999.
	The default value is 9999.

CONTAINER

Parameter	container[N] N=0…9999
Type	Text String
Value	Specifies the location of the disk container. It can be either the name of a ".vdisk" file or the name of a physical disk:

<div style="margin-left: 2em;">

- **Local fixed disks (IDE, SCSI, SATA)**
    - "\\.\PhysicalDrive<N>"
      "\\.\PhysicalDrive(DevID =XXX-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).

      Example:

      ```
      set DUA container[3]="\\.\PhysicalDrive(DevID= 6008-05F3-0005-2950-BF8E-0B86-A0C7-0001)"
      ```

      The WWID values can be obtained from "Host Device Check" utility ("All drives" section), for example:

> If the "Host Device Check" utility does not display any WWID, this means the target disk does not have one. Use the "\\.\PhysicalDrive<N>" mapping in this case.

- **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[0]="\\.\PhysicalDrive(iScsiTarget= iqn.2008-04:iscsi.charon-target-test1, LUN= 1)"
  ```

- **Floppy drives**
    - "\\.\A:"
    - "\\.\B:"

- **CD-ROM, DVD drives (IDE, SCSI, …)**
    - "\\.\CdRom<N>"

</div>

MEDIA_TYPE

Parameter	media_type[N]
	N=0…9999
Type	Text String
Value	Overrides the default (automatically determined) MSCP media type of a device.
	Syntax:
| | ```
"<device-name>[,<device-type>]"
``` |
	where:
	• <device-name> is one of "DU", "DK", "SCSI", "DI", "DSSI", "DJ"
	• <device-type> 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
	If not specified, the device name is set to "DU" and the device type is selected based on disk size.
	Initially not specified.

GEOMETRY

Parameter	geometry[N]
	N=0…9999
Type	Text String
Value	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.
	The string format is <X>"/"<Y>["/"<Z>] where: 1. X is the number of sectors per track; 2. Y is the number of tracks per cylinder; 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;
	If this parameter is not set, CHARON-VAX / CHARON-PDP will configure the geometry based on the most probable disk type.

USE_IO_FILE_BUFFERING

Parameter	use_io_file_buffering[N] N=0…9999
Type	Text String
Value	Instructs CHARON-VAX to enable host operating system I/O cache on reading/writing operations. The following values are possible: • "true" or "write-back" - the "write back" caching mode is used • "write-through" - the "write through" caching mode is used • "false" - caching is switched off ℹ 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". **Example**: `set DUA use_io_file_buffering[2]=true`

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

Parameter	address
Type	Numeric
Value	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.
	The initial values are 017774500 (TQK50) and 0774500 (TUK50), which is the factory setting for these tape controllers.
	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.

CONTAINER

Parameter	container[N] N=0…9999
Type	Text String

Value	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:
	
	• **Tape images**
	    ◦ <file name>.vtape
	
	• **Local physical tape drives recognized by Windows**
	    ◦ "\\.\Tape<N>"

MEDIA_TYPE

Parameter	media_type[N] N=0…9999
Type	Text String
Value	Overrides the default (automatically determined) TMSCP media type of a device. Syntax: ```"<device-name>[,<device-type>]"``` where: • <device-name> is either: "MU", "MK", "SCSI", "MI", "DSSI", or "MJ" • <device-type> 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 If not specified, the device name is set to "MU" and the device type is set to "TK50" Initially not specified.

GEOMETRY

Parameter	geometry[N] N=0...9999
Type	Text String

Value	Specifies the size of a tape image and (optionally) the size of an "early-warning" area at the end of a tape image.
	
	Syntax:
	
	```"<image-size>[, <early-warning-zone-size>]"```
	
	where:
	
	• <image-size> is the tape size in MB
	• <early-warning-zone-size> 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.
	
	Example:
	
	```load TQK50 MUA```
	```set MUA geometry[0] = 90```

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

Parameter	xmi_node_id
Type	Numeric
Value	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

Parameter	container[N] N=0…9999
Type	Text String

Value	Possible values of the parameter are strings in one of the following forms:

- **Physical disk drives**
  - "\\.\PhysicalDrive<X>", where X is 0, 1, …
    "\\.\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).

    Example:

    ```
    set PUA container[3]="\\.\PhysicalDrive(DevID= 6008-05F3-0005-2950-BF8E-0B86-A0C7-0001)"
    ```

    The WWID values can be obtained from "Host Device Check" utility ("All drives" section), for example:

    > If the "Host Device Check" utility does not display any WWID, this means the target disk does not have one. Use the
    > "\\.\PhysicalDrive<N>" mapping in this case.

- **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 PUA container[0]="\\.\PhysicalDrive(iScsiTarget= iqn.2008-04:iscsi.charon-target-test1, LUN= 1)"
    ```

- **Physical tape drives recognized by Windows**
  - "\\.\Tape<X>", where X is 0, 1, …

- **Floppy drives**
  - "\\.\A:"
  - "\\.\B:"

- **CD-ROM drives (read-only)**
  - "\\.\CdRom<X>", where X is 0, 1, …

- **CHARON-VAX disk images**
  - [<drive>":\"<path-name>"\"]<file-name>[".vdisk"]

- **CHARON-VAX tape images**
  - [<drive>":\"<path-name>"\"]<file-name>".vtape"

MEDIA_TYPE

Parameter	media_type[N] N=0…9999
Type	Text String
Value	Overrides the default (automatically determined) MSCP media type of the device. Syntax: `"<device-name>,<device-type>"` where: • <device-name> is either: "DU", "MU", "DK", "MK", "SCSI", "DI", "MI", "DSSI", "DJ", "MJ" • <device-type> 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 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. Initially not specified.

GEOMETRY

Parameter	geometry[N] N=0…9999
Type	Text String
Value	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 <X>"/"<Y>["/"<Z>] where: • X is the number of sectors per track • Y is the number of tracks per cylinder • 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 If this parameter is not set, CHARON-VAX will configure the geometry based on the most probable disk type. Initially not set. The syntax described above is applicable only to disk storage elements. If the container is a tape image, the following format is used instead: Syntax: `"<image-size>[, <early-warning-zone-size>]"` where: • <image-size> is the tape size in MB • <early-warning-zone-size> 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.

USE_IO_FILE_BUFFERING

Parameter	use_io_file_buffering[N]
	N=0…9999
Type	Text String
Value	Instructs CHARON-VAX to enable host operating system I/O cache on reading/writing operations.

The following values are possible:

- "true" or "write-back" - the "write back" caching mode is used
- "write-through" - the "write through" caching mode is used
- "false" - caching is switched off

ℹ 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".

Example:

```
set PUA use_io_file_buffering[0]=true
```

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 *VAXBI* 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

Parameter	vax_bi_node_id
Type	Numeric
Value	Specifies the VAXBI 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

Parameter	container[N] N=0…9999
Type	Text String

Value	Possible values of the parameter are strings in one of the following forms:
	
	• **Physical disk drives**
	    ○ "\\.\PhysicalDrive<X>", where X is 0, 1, …
	       "\\.\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).
	
	    Example:
	
	    `set PUA container[3]="\\.\PhysicalDrive(DevID= 6008-05F3-0005-2950-BF8E-0B86-A0C7-0001)"`
	
	    The WWID values can be obtained from "Host Device Check" utility ("All drives" section), for example:
	
	    If the "Host Device Check" utility does not display any WWID, this means the target disk does not have one. Use the "\\.\PhysicalDrive<N>" mapping in this case.
	
	• **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 PUA container[0]="\\.\PhysicalDrive(iScsiTarget= iqn.2008-04:iscsi.charon-target-test1, LUN= 1)"`
	
	• **Floppy drives**
	    ○ "\\.\A:"
	    ○ "\\.\B:"
	
	• **CD-ROM drives (read-only)**
	    ○ "\\.\CdRom<X>", where X is 0, 1, ...
	
	• **CHARON-VAX disk images**
	    ○ [<drive>":\"<path-name>"\"]<file-name>[".vdisk"]
	
	This parameter is initially not set, thus creating NO storage elements on the storage controller

MEDIA_TYPE

Parameter	media_type[N] N=0…9999
Type	Text String
Value	Overrides the default (automatically determined) MSCP media type of the device. Syntax: `"<device-name>,<device-type>"` where: • <device-name> is either: "DU", "DK", "SCSI", "DI", "DSSI" or "DJ" • <device-type> 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 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. Initially not specified.

GEOMETRY

Parameter	geometry[N] N=0…9999
Type	Text String
Value	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. The string format is <X>"/"<Y>["/"<Z>] where: • X is the number of sectors per track • Y is the number of tracks per cylinder • 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 If this parameter is not set, CHARON-VAX / CHARON-PDP will configure the geometry based on the most probable disk type. Initially not set.

USE_IO_FILE_BUFFERING

Parameter	use_io_file_buffering[N] N=0…9999
Type	Text String
Value	Instructs CHARON-VAX to enable host operating system I/O cache on reading/writing operations. The following values are possible: • "true" or "write-back" - the "write back" caching mode is used • "write-through" - the "write through" caching mode is used • "false" - caching is switched off ℹ 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". **Example**: `set PUA use_io_file_buffering[0]=true`

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

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
virtual_scsi_disk	For disk image containers and physical disks
floppy_scsi_device	For floppy drives
virtual_scsi_tape	For tape image containers
virtual_scsi_cdrom	For host CD-ROM and *.iso images
physical_scsi_device	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.

ℹ️ 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

Parameter	scsi_bus
Type	Identifier
Value	Name of the emulated SCSI disk controller: "pka" or "pkb"

SCSI_ID

Parameter	scsi_id
Type	Numeric
Value	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.

> There is no direct correspondence between the host hardware SCSI ID and these CHARON-VAX SCSI addresses. Set the correspondence between the physical SCSI addresses on the host system and the CHARON-VAX SCSI bus ID in the configuration file.

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.

ℹ️ 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

Parameter	container[N] N=0…7
Type	Text String

Value	A string containing the device name to map to the emulator.

- **Disk images**
  - [<path-name>"\"]<file-name>[".vdisk"]

  Format: A string containing the full path to a disk container.

  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

  "C:\Program Files (x86)\CHARON\Build_XXXXX\x86" (Windows x64; for Windows x86 the path is "C:\Program Files\CHARON\Build_XXXXX\x86").

  > Using compressed folders to store virtual disks and tapes is not supported

- **Local fixed disks (IDE, SATA)**
  - "\\.\PhysicalDrive<X>", where X is 0, 1, …
  - "\\.\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).

  Example:

  ```
  set pka_1 container="\\.\PhysicalDrive(DevID= 6008-05F3-0005-2950-BF8E-0B86-A0C7-0001)"
  ```

  The WWID values can be obtained from "Host Device Check" utility ("All drives" section), for example:

  > If the "Host Device Check" utility does not display any WWID, this means the target disk does not have one. Use the "\\.\PhysicalDrive<N>" mapping in this case.

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

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

  Example:

  ```
  set pka_2 container="\\.\PhysicalDrive(iScsiTarget= iqn.2008-04:iscsi.charon-target-test1, LUN= 1)"
  ```

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

media_type

Parameter	media_type[N] N=0…7
Type	Text String
Value	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

Parameter	geometry[N] N=0…7
Type	Text String
Value	This formatted string value specifies the explicit geometry of the disk storage element The string format is \<X\>"/"\<Y\>["/"\<Z\>] where: • "X" is the number of sectors per track; • "Y" is the number of tracks per cylinder; • "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; If this parameter is not set, CHARON-VAX will configure the geometry based on the most probable disk type.

use_io_file_buffering

Parameter	use_io_file_buffering[N] N=0…7
Type	Text String
Value	Instructs CHARON-VAX to enable host operating system I/O cache on reading/writing operations. The following values are possible: • "true" or "write-back" - the "write back" caching mode is used • "write-through" - the "write through" caching mode is used • "false" - caching is switched off ⓘ Note that this caching has a significant effect only in case of mapping to disk 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". **Example**: `set pka_0 use_io_file_buffering[0]=true`

removable

Parameter	removable[N] N=0…7
Type	boolean
Value	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

Parameter	container[N] N=0…7
Type	Text String
Value	"\\.\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

Parameter	container[N] N=0…7
Type	Text String
Value	A string containing the full path to a tape container. If the specified tape image does not exist, CHARON-VAX creates it. 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"). > If the "CHARON Guest Utilities for OpenVMS" (CHARONCP) package is used, the syntax is different. Please read the corresponding chapter. > Using compressed folders to store virtual disks and tapes is not supported

media_type

Parameter	media_type[N] N=0…7
Type	Text String
Value	Overrides PRODUCT ID in the default SCSI INQUIRY data. Valid values may contain uppercase letters, integers and spaces. Length of this string cannot exceed 16 characters. By default the PRODUCT ID returned is "TLZ04"

geometry

Parameter	geometry[N] N=0…7
Type	Text String

Value	Specifies the size of the tape image and (optionally) the size of the "early-warning" area at the end of the tape image.
	
	Syntax:
	
	`"<image-size>[, <early-warning-zone-size>]"`
	
	where:
	
	• <image-size> is the tape size in MB
	• <early-warning-zone-size> 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.
	
	Example:
	
	`load virtual_scsi_tape/chscsi pka_0`
	`set pka_0 geometry[0] = 90`

⚠ 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

Parameter	container[N] N=0…7
Type	Text String

Value	A string containing the device name to map to the emulator.
	
	   • **ISO images**
	       • [<path-name>"\"]<file-name>[".iso"]
	
	       Format: A string containing the full path to an *.iso image.
	
	       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
	
	       "C:\Program Files (x86)\CHARON\Build_XXXXX\x86" (Windows x64; for Windows x86 the path is "C:\Program Files\CHARON\Build_XXXXX\x86").
	
	   • **CD-ROM and DVD drives (IDE and SATA)**
	       • "\\.\CdRom<X>", where X is 0, 1, …
	
	This parameter is initially not set, thus creating NO storage elements on the controller

media_type

Parameter	media_type[N] N=0…7
Type	Text String
Value	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

Parameter	geometry[N] N=0…7
Type	Text String

Value	This formatted string value specifies the explicit geometry of the disk storage element
	
	The string format is <X>"/"<Y>["/"<Z>] where:
	
	   • "X" is the number of sectors per track;
	   • "Y" is the number of tracks per cylinder;
	   • "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;
	
	If this parameter is not set, CHARON-VAX will configure the geometry based on the most probable disk type.

	use_io_file_buffering
Parameter	use_io_file_buffering[N] N=0…7
Type	Text String
Value	Instructs CHARON-VAX to enable host operating system I/O cache on reading/writing operations. The following values are possible: • "true" or "write-back" - the "write back" caching mode is used • "write-through" - the "write through" caching mode is used • "false" - caching is switched off ⓘ Note that this caching has a significant effect only in case of mapping to ISO images, 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". **Example**: <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

Parameter	container[N] N=0…7
Type	Text String

Value	A string containing the device name to map to the emulator.
	
	 • **Physical tape drives recognized by Windows**
	  • "\\.\Tape<X>", where X is 0, 1, …
	
	 • **Tape changer**
	  • "\\.\Changer<X>", where X is 0, 1, …
	
	 This parameter is initially not set, thus creating NO storage elements on the controller

media_type

Parameter	media_type[N] N=0…7
Type	Text String
Value	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

Parameter	geometry[N] N=0…7
Type	Text String
Value	This formatted string value specifies the explicit geometry of the disk storage element The string format is <X>"/"<Y>["/"<Z>] where: • "X" is the number of sectors per track; • "Y" is the number of tracks per cylinder; • "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; If this parameter is not set, CHARON-VAX will configure the geometry based on the most probable disk type.

removable

Parameter	removable[N] N=0…7
Type	boolean
Value	Enables the logical unit to appear as a removable SCSI disk drive. Initially set to "false" (fixed, non-removable).

disconnect_timeout

Parameter	disconnect_timeout[N] N=0…7
Type	Numeric
Value	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. **Example**: 48 days timeout <pre>set pka_0 disconnect_timeout[1] = 0x400000</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 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

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

Parameter	port[N]
	N=0…7
Type	Numeric
Value	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

Parameter	host[N]
	N=0…7
Type	Text String
Value	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

Parameter	scs_node_name[N]
	N=0…7
Type	Text String
Value	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

Parameter	scs_system_id[N]
	N=0…7
Type	Numeric
Value	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

Parameter	mscp_allocation_class[N] N=0…7
Type	Numeric
Value	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.

259 / 325

CONTAINER

Parameter	container[N] N=0…7
Type	Text String

Value	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:).

Possible values of the parameter are strings in one of the following forms:

- **Physical disk drives**
    - "\\.\PhysicalDrive<X>", where X is 0, 1, …
    - "\\.\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).

    Example:

    ```
    set PAA container[2]="\\.\PhysicalDrive(DevID= 6008-05F3-0005-2950-BF8E-0B86-A0C7-0001)"
    ```

    The WWID values can be obtained from "Host Device Check" utility ("All drives" section), for example:

> If the "Host Device Check" utility does not display any WWID, this means the target disk does not have one. Use the "\\.\PhysicalDrive<N>" mapping in this case.

- **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 PAA container[3]="\\.\PhysicalDrive(iScsiTarget= iqn.2008-04:iscsi.charon-target-test1, LUN= 1)"
    ```

- **Physical tape drives recognized by Windows**
    - "\\.\Tape<X>", where X is 0, 1, …

- **Floppy drives**
    - "\\.\A:" or
    - "\\.\B:"

- **CD-ROM drives (read-only)**
    - "\\.\CdRom<X>", where X is 0, 1, …

- **CHARON-VAX disk images**
    - [<drive>":\"<path-name>"\"]<file-name>[".vdisk"]

- **CHARON-VAX tape images**
    - [<drive>":\"<path-name>"\"]<file-name>".vtape"

	This parameter is initially not set, thus creating NO storage elements on the bus with corresponding DSSI node id.

MEDIA_TYPE

Parameter	media_type[N] N=0…7
Type	Text String
Value	Overrides default (automatically determined) MSCP media type of the device. Syntax: `"<device-name>,<device-type>"` where: • <device-name> is either: "DU", "MU", "DK", "MK", "SCSI", "DI", "MI", "DSSI", "DJ" or "MJ" • <device-type> 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 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. Initially not specified.

GEOMETRY

Parameter	geometry[N] N=0…7
Type	Text String
Value	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 <X>"/"<Y>["/"<Z>] where: 1. X is the number of sectors per track 2. Y is the number of tracks per cylinder 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 If this parameter is not set, CHARON-VAX will configure the geometry based on the most probable disk type. Initially not set. The syntax above is applicable only to disk storage elements. If the container is a tape image, the following format is used: Syntax: `"<image-size>[, <early-warning-zone-size>]"` where: • <image-size> is the tape size in MB • <early-warning-zone-size> 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.

USE_IO_FILE_BUFFERING

Parameter	use_io_file_buffering[N] N=0…7
Type	Text String
Value	Instructs CHARON-VAX to enable host operating system I/O cache on reading/writing operations. The following values are possible: • "true" or "write-back" - the "write back" caching mode is used • "write-through" - the "write through" caching mode is used • "false" - caching is switched off ℹ 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". **Example**: `set PAA use_io_file_buffering[0]=true`

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\dia1-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 SCSNODENAME as the instance name.The line below loads an emulated HSD50 storage controller, assigns it the instance name SCSNODE 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

Parameter	dssi_host
Type	Text String
Value	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

Parameter	dssi_node_id
Type	Numeric
Value	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

Parameter	scs_node_name
Type	Text String
Value	A string value that specifies the SCSNODENAME 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

Parameter	scs_system_id
Type	Numeric
Value	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

Parameter	mscp_allocation_class
Type	Numeric
Value	An integer value that specifies the ALLOCLASS of the emulated HSD50 storage controller.Possible values are 0 through 255 (initially set to 0).

CONTAINER

Parameter	container[N]N=0…9999
Type	Text String
Value	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:

- **Physical disk drives**
  - "\\.\PhysicalDrive<X>", where X is 0, 1, …
  - "\\.\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).**Example**:

```
set DISKS container[2]="\\.\PhysicalDrive(DevID= 6008-05F3-0005-2950-BF8E-0B86-A0C7-0001)"
```

The WWID values can be obtained from "Host Device Check" utility ("All drives" section), for example:

If the "Host Device Check" utility does not display any WWID, this means the target disk does not have one. Use the "\\.\PhysicalDrive<N>" mapping in this case.

- **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 DISKS container[3]="\\.\PhysicalDrive(iScsiTarget= iqn.2008-04:iscsi.charon-target-test1, LUN= 1)"
```

- **Physical tape drives recognized by Windows**
  - "\\.\Tape<X>", where X is 0, 1, …

- **Floppy drives**
  - "\\.\A:" or
  - "\\.\B:"

- **CD-ROM drives (read-only)**
  - "\\.\CdRom<X>", where X is 0, 1, …

- **CHARON-VAX disk images**
  - [<drive>":\"<path-name>"\"]<file-name>[".vdisk"]

- **CHARON-VAX tape images**
  - [<drive>":\"<path-name>"\"]<file-name>".vtape"

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

MEDIA_TYPE

Parameter	media_type[N]N=0…9999
Type	Text String
Value	Overrides the default (automatically determined) MSCP media type of the device.Syntax: `"<device-name>,<device-type>"` where: • <device-name> is either: "DU", "MU", "DK", "MK", "SCSI", "DI", "MI", "DSSI", "DJ" or "MJ" • <device-type> 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 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.

GEOMETRY

Parameter	geometry[N]N=0…9999
Type	Text String
Value	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 <X>"/"<Y>["/"<Z>] where: 1. X is the number of sectors per track 2. Y is the number of tracks per cylinder 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 If this parameter is not set, CHARON-VAX will configure the geometry based on the most probable disk type.Initially not set.The syntax above is applicable only to disk storage elements.If the container is a tape image, the following format is used:Syntax: `"<image-size>[, <early-warning-zone-size>]"` where: • <image-size> is the tape size in MB • <early-warning-zone-size> 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.

USE_IO_FILE_BUFFERING

Parameter	use_io_file_buffering[N]N=0…9999
Type	Text String
Value	Instructs CHARON-VAX to enable host operating system I/O cache on reading/writing operations.The following values are possible: • "true" or "write-back" - the "write back" caching mode is used • "write-through" - the "write through" caching mode is used • "false" - caching is switched off ⓘ 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".**Example**: `set DISKS use_io_file_buffering[0]=true`

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

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 XMI 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

Parameter	port[N] N=0…127
Type	Numeric
Value	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

Parameter	host[N] N=0…127
Type	Text String
Value	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

Parameter	xmi_node_id
Type	Numeric
Value	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

Parameter	ci_node_id
Type	Numeric
Value	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\dua0-rz24-vms-v6.2.vdisk"

set DISKS container[1]="C:\Charon\Disks\dua1-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

Parameter	ci_host
Type	Text String
Value	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

Parameter	ci_node_id
Type	Numeric
Value	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

Parameter	scs_node_name
Type	Text String
Value	A string value that specifies the SCSNODENAME of the emulated HSJ50 storage controller. The string can contain up to 6 characters. Possible characters are uppercase letters A through Z and the integers 0 through 9. Initially set to the name of the emulated HSJ50 controller. Therefore the name of the emulated HSJ50 controller must follow the above rules.

SCS_SYSTEM_ID

Parameter	scs_system_id
Type	Numeric
Value	An integer value that specifies the SCSSYSTEMID of the emulated HSJ50 storage controller. Initially set to an arbitrary value that is guaranteed to be unique within the running emulated VAX 66x0 node.

MSCP_ALLOCATION_CLASS

Parameter	mscp_allocation_class
Type	Numeric
Value	An integer value that specifies the ALLOCLASS of the emulated HSJ50 storage controller. Possible values are 0 through 255 (initially set to 0).

CONTAINER

Parameter	container[N]	
	N=0…9999	
Type	Text String	

Value	Possible values of the parameter are strings in one of the following forms:

	• **Physical tape drives**
	    ◦ "\\.\PhysicalDrive<X>", where X is 0, 1, …
	    ◦ "\\.\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).
	
	Example:
	
	`set DISKS container[2]="\\.\PhysicalDrive(DevID= 6008-05F3-0005-2950-BF8E-0B86-A0C7-0001)"`
	
	The WWID values can be obtained from "Host Device Check" utility ("All drives" section), for example:
	
	If the "Host Device Check" utility does not display any WWID, this means the target disk does not have one. Use the "\\.\PhysicalDrive<N>" mapping in this case.
	
	• **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 DISKS container[3]="\\.\PhysicalDrive(iScsiTarget= iqn.2008-04:iscsi.charon-target-test1, LUN= 1)"`
	
	• **Physical tape drives recognized by Windows**
	    ◦ "\\.\Tape<X>", where X is 0, 1, …
	
	• **Floppy drives**
	    ◦ "\\.\A:" or
	    ◦ "\\.\B:"
	
	• **CD-ROM drives (read-only)**
	    ◦ "\\.\CdRom<X>", where X is 0, 1, …
	
	• **CHARON-VAX disk images**
	    ◦ [<drive>":\"<path-name>"\"]<file-name>[".vdisk"]
	
	• **CHARON-VAX tape images**
	    ◦ [<drive>":\"<path-name>"\"]<file-name>".vtape"
	
	This parameter is initially not set, thus creating NO storage elements on the controller

MEDIA_TYPE

Parameter	media_type[N] N=0…9999
Type	Text String
Value	Overrides the default (automatically determined) MSCP media type of the device. Syntax: ```"<device-name>,<device-type>"``` where: • <device-name> is either: "DU", "MU", "DK", "MK", "SCSI", "DI", "MI", "CI", "DJ" or "MJ" • <device-type> 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 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.

GEOMETRY

Parameter	geometry[N] N=0…9999
Type	Text String
Value	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. The string format is <X>"/"<Y>["/"<Z>] where: 1. X is the number of sectors per track 2. Y is the number of tracks per cylinder 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 If this parameter is not set, CHARON-VAX will configure the geometry based on the most probable disk type. Initially not set. The syntax above is applicable only to disk storage elements. If the container is a tape image, the following format is used: Syntax: ```"<image-size>[, <early-warning-zone-size>]"``` where: • <image-size> is the tape size in MB • <early-warning-zone-size> 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.

USE_IO_FILE_BUFFERING

Parameter	use_io_file_buffering[N] N=0…9999
Type	Text String
Value	Instructs CHARON-VAX to enable host operating system I/O cache on reading/writing operations. The following values are possible: • "true" or "write-back" - the "write back" caching mode is used • "write-through" - the "write through" caching mode is used • "false" - caching is switched off ℹ️ 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". **Example**: `set DISKS use_io_file_buffering[0]=true`

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\dua1-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
```

ℹ 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

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.

SGEC Ethernet Controller

The built-in SGEC controller emulator ("EZA") has the following parameters that are specified with the "set" command:

interface

Parameter	interface
Type	Text string
Value	Name of the corresponding instance of the "packet_port" component

station_address

Parameter	station_address
Type	Text String

Value	The "station_address" provides the ability to configure the adapter's permanent address. By default the adapter's permanent address is read from the host system's NIC.
	
	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.
	
	Format:
	
	`XX-XX-XX-XX-XX-XX`
	
	or
	
	`XX:XX:XX:XX:XX:XX`
	
	Example:
	
	`set eza station_address="AF:01:AC:78:1B:CC"`

rx_fifo_size

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

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

Parameter	address
Type	Numeric

Value	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.
	
	 Use the address parameter if loading several instances of DEQNA, DESQA, DELQA, DEUNA and DELUA Ethernet controllers.
	
	 "address" parameter values must be unique for every instance of an DEQNA, DESQA, DELQA, DEUNA and DELUA Ethernet controller.
	
	 Example:
	
	 ``` load DEQNA/DEQNA XQA address=017774440```
	``` load DEQNA/DEQNA XQB address=017764460```

interface

Parameter	interface
Type	Text String
Value	Name of the corresponding instance of the "packet_port" component

station_address

Parameter	station_address
Type	Text String

Value	"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.
	
	 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.
	
	 Format:
	
	 `XX-XX-XX-XX-XX-XX`
	
	 or
	
	 `XX:XX:XX:XX:XX:XX`
	
	 Example:
	
	 `set QNA station_address="AF:01:AC:78:1B:CC"`

rx_fifo_size

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

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

Parameter	xmi_node_id
Type	Number
Value	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

Parameter	interface
Type	Text String
Value	Name of the corresponding instance of the "packet_port" component

station_address

Parameter	station_address
Type	Text String

Value	The "station_address" provides the ability to configure the adapter's permanent address. By default the adapter's permanent address is read from the host system's NIC.
	
	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.
	
	Format:
	
	`XX-XX-XX-XX-XX-XX`
	
	or
	
	`XX:XX:XX:XX:XX:XX`
	
	Example:
	
	`set EXA station_address="AF:01:AC:78:1B:CC"`

rx_fifo_size

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

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

Parameter	vax_bi_node_id
Type	Number
Value	Specifies the VAXBI slot in which the virtual DEBNI controller is placed.

interface

Parameter	interface
Type	Text String
Value	Name of the corresponding instance of the "packet_port" component

station_address

Parameter	station_address
Type	Text String

Value	The "station_address" provides the ability to configure the adapter's permanent address. By default the adapter's permanent address is read from the host system's NIC.
	
	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.
	
	Format:
	
	`XX-XX-XX-XX-XX-XX`
	
	or
	
	`XX:XX:XX:XX:XX:XX`
	
	Example:
	
	`set EXA station_address="AF:01:AC:78:1B:CC"`

rx_fifo_size

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

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

Parameter	interface
Type	Text String
Value	Name of the corresponding instance of the "packet_port" component

station_address

Parameter	station_address
Type	Text String

Value	The "station_address" provides the ability to configure the adapter's permanent address. By default the adapter's permanent address is read from the host system's NIC.
	
	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.
	
	Format:
	
	`XX-XX-XX-XX-XX-XX`
	
	or
	
	`XX:XX:XX:XX:XX:XX`
	
	Example:
	
	`set ECA station_address="AF:01:AC:78:1B:CC"`

rx_fifo_size

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

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

Parameter	interface
Type	Text string

Value	This parameter identifies an Ethernet adapter of the host system dedicated to CHARON-VAX / CHARON-PDP with a keyword "connection" preceeding it.
	
	Syntax:
	
	`set <name> interface="connection:<adapter>"`
	
	Example:
	
	`set pp_1 interface="connection:Charon"`

port_enable_mac_addr_change

Parameter	port_enable_mac_addr_change
Type	Boolean
Value	If "true" is specified (default value), CHARON sets the appropriate Ethernet address automatically. If "false" is specified, set the Ethernet address manually. **Example**: `set pp_1 port_enable_mac_addr_change=false`

port_retry_on_tx

Parameter	port_retry_on_tx
Type	Numeric
Value	The "port_retry_on_tx" parameter controls the number of times a port will attempt to transmit a packet before giving up. By default, the value is 3. Increasing this value may introduce problems in carrier loss logic, because not all NIC drivers support a carrier status query. Typically, you do not need to increase the value. **Example**: `set pp_1 port_retry_on_tx=8`

port_pending_rx_number

Parameter	port_pending_rx_number
Type	Numeric
Value	The "port_pending_rx_number" parameter sets the number of pending receive buffers. The default value is 63. The maximum value allowed is 195. You may want to increase the "port_pending_rx_number" when you have very busy networking and experience problems like losing connections not related to the carrier loss. Typically, you do not need to change this parameter. **Example**: `set pp_1 port_pending_rx_number=128`

port_pending_tx_number

Parameter	port_pending_tx_number
Type	Numeric
Value	The "port_pending_tx_number" parameter sets the number of buffers the port uses to transmit. The default value is 62. You may want to increase the "port_pending_tx_number" value if the log file indicates dropped TX packets due to TX queue overflow. Typically, you do not need to change this parameter. **Example**: `set pp_1 port_pending_tx_number=128`

suspend_msg_on_mac_change

Parameter	suspend_msg_on_mac_change
Type	Boolean
Value	To avoid confusion arising from non critical errors during a MAC address change, logging is by default suppressed (default value is "true"). To enable tracing during a MAC address change set this parameter to "false" **Example**: `set pp_1 suspend_msg_on_mac_change=false`

log

Parameter	log
Type	Text string
Value	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.

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.

In certain situations enabling this parameter may help to detect loss of packets.

Example:

```
set pp_1 log="pp_1.log"
```

```
set pp_1 log="C:\Charon\Logs"
```

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

legacy_mode

Parameter	legacy_mode
Type	Boolean
Value	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.

This setting may be needed if CHARON-VAX runs in VMware environment for example and you do not want to allow promiscuous mode.

In other situations please do not change its default value ("false").

Example:

```
set pp_1 legacy_mode=true
```

> 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.
>
> Thus for ESXi networking configuration:
>
> 1. Create the port with e1000 adapter;
> 2. Enable the promiscuous traffic;
> 3. Do not use legacy_mode option in Charon's configuration file

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

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

Parameter	address
Type	Numeric
Value	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

Parameter	vector
Type	Numeric
Value	Specifies the interrupt vector. The initial value is 0200, which is the factory setting for an LPV11.

parameters

Parameter	*parameters*
Type	Text String
Value	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 HOSTPrint, (along with some additional parameters):

Parameters	Description
"\\.\LPT<N>"	Host port to connect to a physical printer
"port"	TCP/IP port to connect to an external application
"application"	An application to run after opening the port. Note: the name of the application including its own parameters must be framed within ' characters (apostrophe).
-port	The TCP/IP port the HOSTprint utility will connect to.
-fontsize	The font size to be used for printing.
-font	The font to be used for printing. The font name must be framed with \ (backslash) symbols.

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 and VAX/VMS boots with a "D" (workstation) type license, the license is accepted because VMS recognizes that a VCB02 is present.

As the dummy VCB02 is located on the secondary address (0x3fff02), VMS boots using the normal console terminal without graphics capability.

The format is as follows:

```
load dummy_vcb02/null_vcb02 dummy
```

> Since this dummy device cannot draw graphics, CHARON-VAX will fail the initial hardware diagnostics, but it will still 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

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 chosing the QUART
# line number (in square brackets). See OPA0 example, below.
#
#-------------------------------------------------------------------------------

#load physical_serial_line/chserial TTA0 line="\\.\COMn"
#load virtual_serial_line/chserial TTA0 port=10000
#load virtual_serial_line/chserial TTA0 port=10000 application="tta0.ht"
#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="\\.\COM1"
#load virtual_serial_line/chserial TTA1 port=10001
#load virtual_serial_line/chserial TTA1 port=10001 application="tta1.ht"
#load virtual_serial_line/chserial TTA1 port=10001 application="putty.exe -load TTA1"
#set quart line[1]=TTA1

#load physical_serial_line/chserial TTA2 line="\\.\COM2"
#load virtual_serial_line/chserial TTA2 port=10002
#load virtual_serial_line/chserial TTA2 port=10002 application="tta2.ht"
#load virtual_serial_line/chserial 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/chserial OPA0 line="\\.\COM3"
#load virtual_serial_line/chserial OPA0 port=10003
#load virtual_serial_line/chserial OPA0 port=10003 application="opa0.ht"
load virtual_serial_line/chserial OPA0 port=10003 application="putty.exe -load OPA0"
set quart line[3]=OPA0

#===============================================================================
#
# 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 physical_scsi_device 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 then 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

#===============================================================================
```

```
#
# Uncomment to enable the built-in SGEC Ethernet Adapter (EZA).
#
#-------------------------------------------------------------------------------

#set EZA interface=EZA0

#===============================================================================
#
# Connect the SGEC Ethernet Adapter (EZA) to the host's NIC.
#
#-------------------------------------------------------------------------------

#load packet_port EZA0 interface="(disabled)"
#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/chserial TXA0 line="\\.\COMn"
#load virtual_serial_line/chserial TXA0 port=10010
#load virtual_serial_line/chserial TXA0 port=10010 application="txa0.ht"
#set TXA line[0]=TXA0

#load physical_serial_line/chserial TXA1 line="\\.\COMn"
#load virtual_serial_line/chserial TXA1 port=10011
#load virtual_serial_line/chserial TXA1 port=10011 application="txa1.ht"
#set TXA line[1]=TXA1

#load physical_serial_line/chserial TXA2 line="\\.\COMn"
#load virtual_serial_line/chserial TXA2 port=10012
#load virtual_serial_line/chserial TXA2 port=10012 application="txa2.ht"
#set TXA line[2]=TXA2

#load physical_serial_line/chserial TXA3 line="\\.\COMn"
#load virtual_serial_line/chserial TXA3 port=10013
#load virtual_serial_line/chserial TXA3 port=10013 application="txa3.ht"
#set TXA line[3]=TXA3

#load physical_serial_line/chserial TXA4 line="\\.\COMn"
#load virtual_serial_line/chserial TXA4 port=10014
#load virtual_serial_line/chserial TXA4 port=10014 application="txa4.ht"
#set TXA line[4]=TXA4

#load physical_serial_line/chserial TXA5 line="\\.\COMn"
#load virtual_serial_line/chserial TXA5 port=10015
#load virtual_serial_line/chserial TXA5 port=10015 application="txa5.ht"
#set TXA line[5]=TXA5

#load physical_serial_line/chserial TXA6 line="\\.\COMn"
#load virtual_serial_line/chserial TXA6 port=10016
#load virtual_serial_line/chserial TXA6 port=10016 application="txa6.ht"
#set TXA line[6]=TXA6

#load physical_serial_line/chserial TXA7 line="\\.\COMn"
#load virtual_serial_line/chserial TXA7 port=10017
#load virtual_serial_line/chserial 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=...

#==========================================================================
#
# Configuring an optional DELQA Ethernet adapters (QBUS).
#
#--------------------------------------------------------------------------

#load DELQA/DEQNA XQA

#load packet_port XQA0 interface="connection:<connection-name>"
#set XQA interface=XQA0

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

#==========================================================================
#
# 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/chserial TXA0 line="\\.\COMn"
#load virtual_serial_line/chserial TXA0 port=10010
#load virtual_serial_line/chserial TXA0 port=10010 application="txa0.ht"
#set TXA line[0]=TXA0

#load physical_serial_line/chserial TXA1 line="\\.\COMn"
#load virtual_serial_line/chserial TXA1 port=10011
#load virtual_serial_line/chserial TXA1 port=10011 application="txa1.ht"
#set TXA line[1]=TXA1

#load physical_serial_line/chserial TXA2 line="\\.\COMn"
#load virtual_serial_line/chserial TXA2 port=10012
#load virtual_serial_line/chserial TXA2 port=10012 application="txa2.ht"
#set TXA line[2]=TXA2

#load physical_serial_line/chserial TXA3 line="\\.\COMn"
#load virtual_serial_line/chserial TXA3 port=10013
#load virtual_serial_line/chserial TXA3 port=10013 application="txa3.ht"
#set TXA line[3]=TXA3

#load physical_serial_line/chserial TXA4 line="\\.\COMn"
#load virtual_serial_line/chserial TXA4 port=10014
```

```
#load virtual_serial_line/chserial TXA4 port=10014 application="txa4.ht"
#set TXA line[4]=TXA4

#load physical_serial_line/chserial TXA5 line="\\.\COMn"
#load virtual_serial_line/chserial TXA5 port=10015
#load virtual_serial_line/chserial TXA5 port=10015 application="txa5.ht"
#set TXA line[5]=TXA5

#load physical_serial_line/chserial TXA6 line="\\.\COMn"
#load virtual_serial_line/chserial TXA6 port=10016
#load virtual_serial_line/chserial TXA6 port=10016 application="txa6.ht"
#set TXA line[6]=TXA6

#load physical_serial_line/chserial TXA7 line="\\.\COMn"
#load virtual_serial_line/chserial TXA7 port=10017
#load virtual_serial_line/chserial 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

#
# It is possible to reduce the size of the log file using filtering
# possibilities:
#
# 1) Based on message type (info, warning, error)
# 2) Based on repeatetive messages removing.
#
# You can specify which type of messages should be logged using option
# 'log_show_messages' which is a string option containing mentioned above
# message types delimited by comma. Default is "all" which means to log
# all existing message types.
#
# You can specify that repeatetive messages should be filtered out using
# 'log_repeat_filter' "on"/"off" option. Default value is "off"
#

# The same as default "all"
set session log_show_messages="info, warning, error"

# Show only information and error messages
#set session log_show_messages="info, error"

# Show only error messages
#set session log_show_messages="error"

# Filter repeatetive messages out
#set session log_repeat_filter="on"


#
# The following lines tell the emulator where to preserve NVRAM content.
# It will keep the current time of the emulated VAX (when you do not
```

```
# run the emulator) and console parameters (such as default boot device).
# Both files must be enabled to correctly preserve the saved state of the console.
#

#set toy container="charon.dat"
#set eeprom container="charon.rom"

#
# Specify the size of RAM (default 512MB). Remember that the dongle
# license might limit the maximum amount of memory.
#

set ram size=32
#set ram size=64
#set ram size=128
#set ram size=256
#set ram size=512

#
# Now assign console built-in serial line. Currently the
# emulator offers three 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). And the third one is to connect to MOXA
# NPort 5210 box ports (via np5210_serial_line).
#
# Once desired way of connection is chosen load the interface
# like OPA0 which is expected by built-in console controller.
#

# MOXA NPort 5210 box connection
#load np5210_serial_line/chserial OPA0
#set OPA0 ip="xxx.xxx.xxx.xxx" rs_port=n

# Physical COM port connection
#load physical_serial_line/chserial OPA0 line="COM1:"

#
# Virtual serial line connection listening for the TCP/IP port
# number 10003 on the host system. It is possible to connect to
# this port from any appropriate application (standard terminal,
# user written terminal, etc...).
#
#load virtual_serial_line/chserial OPA0 port=10003

#
# Virtual serial line connection listening for the TCP/IP port
# number 10003 on the host system. Specified program is started
# automatically.
#
load virtual_serial_line/chserial OPA0
#set OPA0 port=10003 application="opa0.ht"
set OPA0 port=10003 application="putty.exe -load OPA0"
#set OPA0 port=10003 application="c:\kea\user\opa0.ktc"


# load DWMBB XMI-to-VAXBI adapter into slot 14 of the XMI
load DWMBB XBA xmi_node_id=14

#
# Slots 1 - 15 (1 - F) of the VAXBI are able to handle I/O adapters.
#

# load KDB50 MSCP disk controller into slot 1 of the VAX BI
load KDB50 PUA vax_bi_node_id=1

#
# Attach unit 0 of the PUA controller to disk image (.dsk or .vdisk);
# VMS would see it as "DUA0:"
#
#set PUA container[0]="..\vdisk\vmssys.vdisk"

#
# Attach unit 1 of the PUA controller to physical disk drive (N to be
# replaced with decimal number); VMS would see it as "DUA1:"
```

```
#set PUA container[1]="\\.\PhysicalDriveN"
#

# ...

#
# Attach unit 9 of the PUA controller to CDROM drive (0 might to be
# replaced with another decimal number, if necessary); VMS would see
# it as "DUA9:"
#set PUA container[9]="\\.\CdRomN"

#
# Attach unit 10 of the PUA controller to floppy disk drive ('A' to be
# replaced with 'B', if necessary); VMS would see it as "DUA10:"
#set PUA container[10]="\\.\A:"

# ...

# load DEBNI Ethernet Controller into slot 2 of the VAXBI
load DEBNI/DEMNA ETA vax_bi_node_id=2

# connect the ETA Ethernet controller to host network interface
#load packet_port/chnetwrk ETA0 interface="connection:<connection-name>"
#set ETA interface=ETA0

# Load VAXBI to UNIBUS adapter
load DWBUA UBA vax_bi_node_id=14

# Load TMSCP tape to have the tape instead of non-implemented TBK70
load TUK50 MUA

#
# Attach MUA controller to tape image (.mtd or .vtape); VMS would see
# it as "MUA0:"
#
set MUA container[0]="mua0.mtd"

#
# Attach MUA controller to physical tape drive (N to be replaced with
# decimal number); VMS would see it as "MUA0:"
#
#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"

#==============================================================================
#
# 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>"

#===============================================================================
#
# DEMNA XMI Ethernet Adapter EXA.
#
# Uncomment both lines to enable emulation of EXA.
#
#-------------------------------------------------------------------------------

#load DEMNA EXA xmi_node_id = 13 interface = EXA0
#load packet_port EXA0 interface = "(disabled)"

#-------------------------------------------------------------------------------
#
# Connect EXA0 to a host NIC.
#
#-------------------------------------------------------------------------------

#set EXA0 interface = "connection:<connection-name>"

#===============================================================================
#
# DEMNA XMI Ethernet Adapter EXB.
#
```

```
# Uncomment both lines to enable emulation of EXB.
#
#------------------------------------------------------------------------------

#load DEMNA EXB xmi_node_id = 14 interface = EXB0
#load packet_port EXB0 interface = "(disabled)"

#------------------------------------------------------------------------------
#
# Connect EXB0 to a host NIC.
#
#------------------------------------------------------------------------------

#set EXB0 interface = "connection:<connection-name>"

# this is the end of the configuration file ################################
```

CHARON-VAX and CHARON-PDP for Windows deinstallation

Deinstallation procedure

Perform a clean shutdown of the running guests, open the CHARON Virtual Machines 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 Machines 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 Machines 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 Machines Manager" is still running the following dialog will be displayed:

Do not close the uninstallation dialog but exit from "CHARON Virtual Machines 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

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.

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:

```
myds20.cfg - Notepad
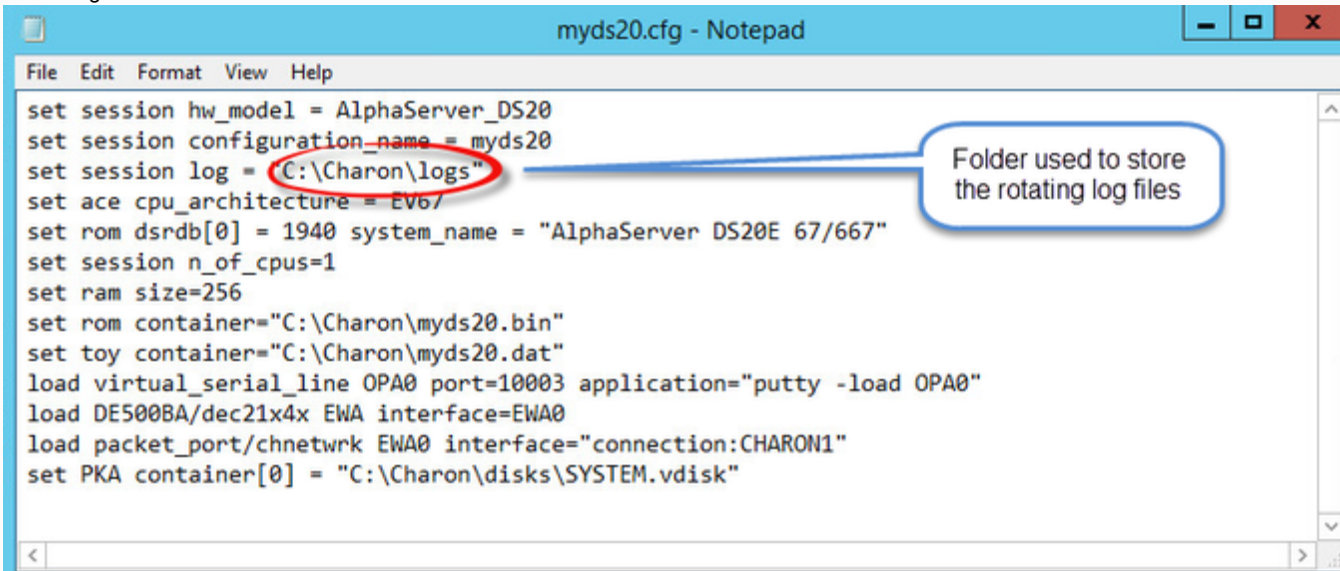
File  Edit  Format  View  Help

set session hw_model = AlphaServer_DS20
set session configuration_name = myds20
set session log = C:\Charon\logs"          ← Folder used to store
set ace cpu_architecture = EV67              the rotating log files
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

  ℹ 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**

■ **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

- 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**).

  ℹ 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

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.
   ℹ 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'"
```

ℹ 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:

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

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
-usessl -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
-usessl -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 Log monitoring on Windows (logmond) - Best practices for V4.9 and V4.10
- CHARON on Windows - Automated License Expiration Check
- CHARON on Windows - Automated License Expiration Check - Release Notes
- Charon Log monitoring on Windows (logmond) - Best practices for V4.8
- Charon Log monitoring on Windows (logmond) - Best practices for V4.6 and V4.7