CHARON-VAX and CHARON-PDP V4.7 for Windows - Users Guide

Contents

Introduction . . . 3
Conventions . . . 6
CHARON-VAX and CHARON-PDP for Windows installation . . . 7
Running CHARON-VAX and CHARON-PDP for Windows . . . 34
CHARON-VAX and CHARON-PDP for Windows configuration . . . 63
Migration to CHARON-VAX and CHARON-PDP for Windows . . . 80
CHARON-VAX for Windows DSSI cluster . . . 90
CHARON-VAX for Windows CI cluster . . . 96
CHARON-VAX and CHARON-PDP for Windows licensing . . . 100
CHARON-VAX and CHARON-PDP for Windows utilities . . . 117
- CHARON Launcher . . . 118
- CHARON Service Manager . . . 129
- Host Device Check . . . 145
- Network Control Center . . . 148
- MkDisk . . . 185
- MkDskCmd . . . 194
- HASP View . . . 197
- License Update Service . . . 199
- mtd . . . 204
- CHARON Log Monitor and Dispatcher . . . 205
- Performance Improvement Package . . . 208
- HOSTPrint . . . 210
- VT-30TV . . . 212

CHARON-VAX and CHARON-PDP for Windows configuration details . . . 213
- General Settings . . . 214
- Core Devices . . . 221
- Serial lines . . . 228
- Disks and tapes . . . 240
  - MSCP and TMSCP Controllers . . . 241
  - SCSI Controllers . . . 256
  - DSSI Subsystem . . . 267
  - CI Subsystem . . . 277
- Networking . . . 285
- Specific Controllers . . . 297
- Sample configuration files . . . 300
  - PDP-11/93 configuration file . . . 301
  - VAX 4000 Model 108 configuration file . . . 304
  - VAX 6310 configuration file . . . 310
  - VAX 6610 configuration file . . . 313

CHARON-VAX and CHARON-PDP for Windows deinstallation . . . 317

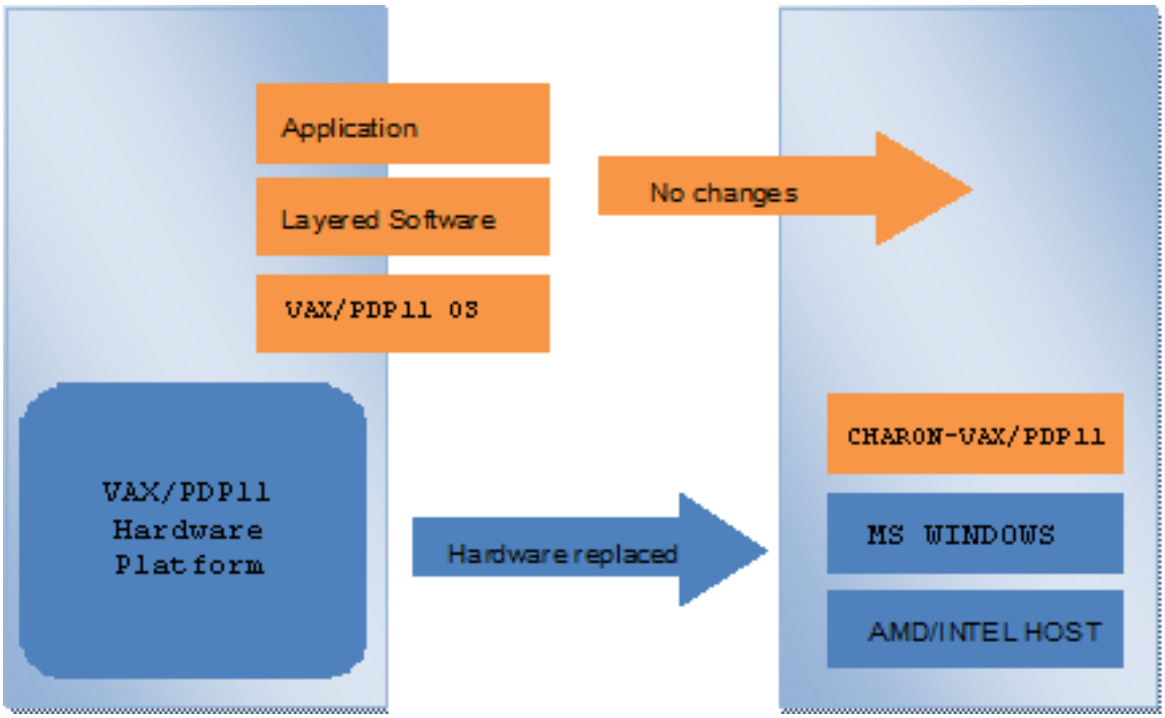
Introduction

Table of Contents

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

General Description

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



This approach is best understood when the VAX and PDP11 Hardware Virtualization Software is viewed as a special interface between the old VAX and PDP11 software and a new hardware platform. Basically, the CHARON software presents a VAX and PDP11 hardware interface to the original VAX and PDP11 software, so that the existing software cannot detect a difference. This means no changes have to be made to the existing software. User programs and data can be copied to a new modern industry standard server (64-bit or 32-bit Intel or AMD) and continue to run for many more years.

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

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

Back to Table of Contents

© Stromasys, 2017

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:

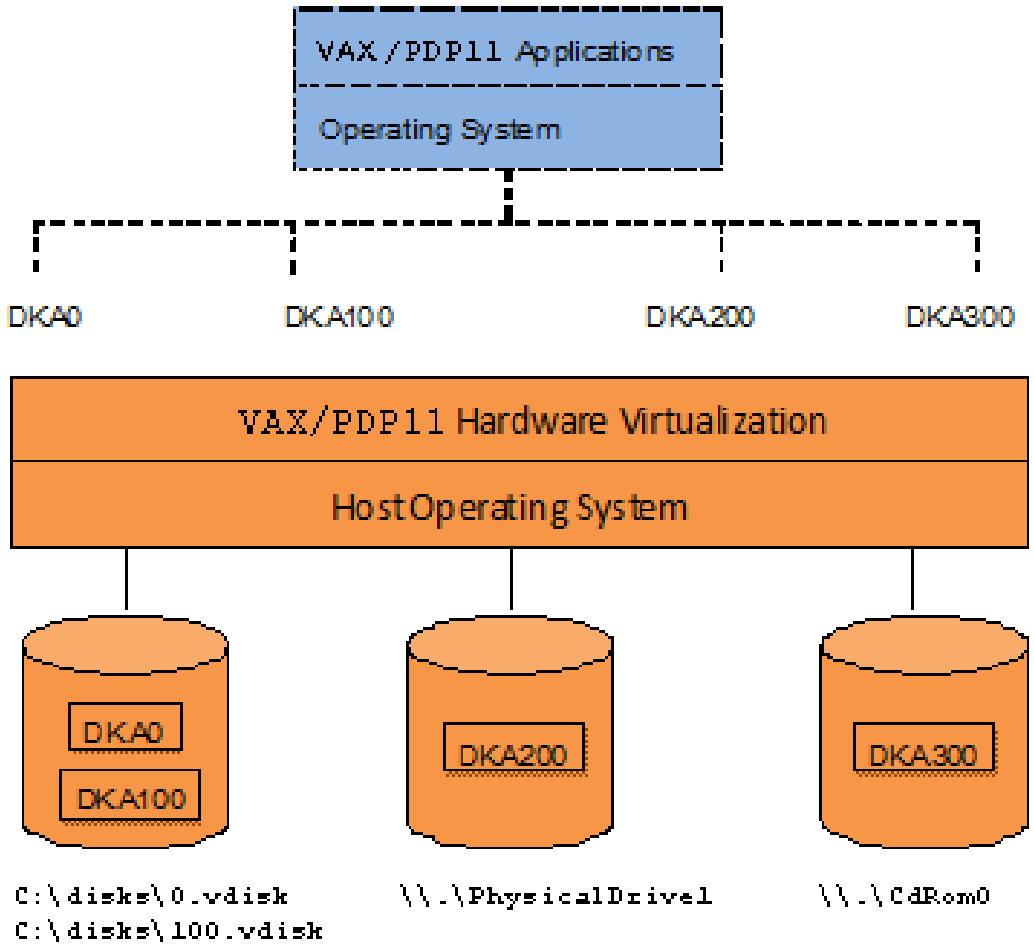
Subsystem	Covered VAX and PDP11 hardware
Serial Lines Controllers	UART, QUART, CXA16, CXB16, CXY08, DHQ11, DHV11, DZV11, DZQ11, DL11, DLV11, DZ11, DHW42-AA, DHW42-BA, DHW42-CA
XMI Disks/Tapes Controllers	KDM70
BI Disks Controllers	KDB50
QBUS Disks Controllers	RQDX3
QBUS/UNIBUS Tapes Controllers	TQK50,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

Back to Table of Contents

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.

© Stromasys, 2017



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

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

Back to Table of Contents

© Stromasys, 2017

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 or MPE

© Stromasys, 2017

CHARON-VAX and CHARON-PDP for Windows installation

Table of contents

- Introduction
- Hardware Requirements
  - Number of CPU cores
  - CPU type and speed
  - Operative memory
  - Disk storage
  - Ethernet adapters
- Software Requirements
- Host system preparation
- Before installation
- Distribution preparation
- Installation
- CHARON-VAX / CHARON-PDP home directory
- License installation
  - Regular HASP USB dongle
  - Network HASP USB dongle
  - Software license
- License validity verification
  - Troubleshooting
- Network configuration
  - For Windows 2008 R2, Windows 2012 R2, Windows 7, 8 and 8.1
  - For older versions of Windows
- After installation

Introduction

CHARON-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.

Back to Table of Contents

© Stromasys, 2017

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

> **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 50 MB for the CHARON software plus space required for the normal host OS.

© Stromasys, 2017

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!

Back to Table of Contents

Software Requirements

- Microsoft Windows Server 2012 R2 Standard and Datacenter Editions, 64 bit version (please notice that only R2 revision is supported)
- Microsoft Windows Server 2008 R2 (SP1) Standard and Enterprise Editions, 64 bit version (please notice that only R2 revision is supported)
- Microsoft Windows 7 Professional and Ultimate (SP1) Editions, 32 bit and 64 bit versions
- Microsoft Windows 8.1 Professional Edition, 32 and 64 bit versions
- VMware ESXi 5.x and 6.0 (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)

Back to Table of Contents

Host system preparation

> All antivirus, screen saver, automatic (scheduled) backup and any other CPU consuming software as well as software that is able to temporarily lock CHARON files (such as automatic indexing) must be turned off, uninstalled and disabled.
>
> The power scheme must be set to "High Performance" with all the "Sleep" and "Standby" modes turned off.
>
> The automatic installation of Microsoft updates must be disabled. MS updates to the CHARON host must be done only in specific service 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 clients side*: open UDP ports 30000-65535
- *Both on server and client sides*: setup default gateway

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

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

© Stromasys, 2017

2. Find the "AlwaysDisableMCN" parameter in the following way:

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

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

4. Reboot the host system.

Back to Table of Contents

Before installation

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

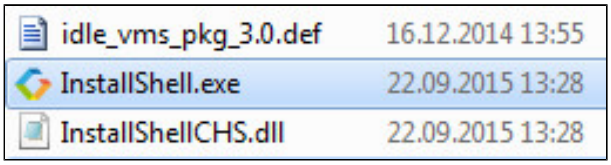
> For correct work of CHARON the installation procedure will create a special group called CHARON-GRP and the current user will be automatically included to this group. Do not remove this group and do not extract CHARON user from it, otherwise a problem with virtual memory allocation may appear on CHARON start.
>
> The created user may belong to some domain, but please note that in this case you have to add this user to the CHARON-GRP manually as it is described in this article and then reboot CHARON host.

3. Stay logged in as local system administrator ("Administrator") or log off and login as the CHARON-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.

Back to Table of Contents

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:



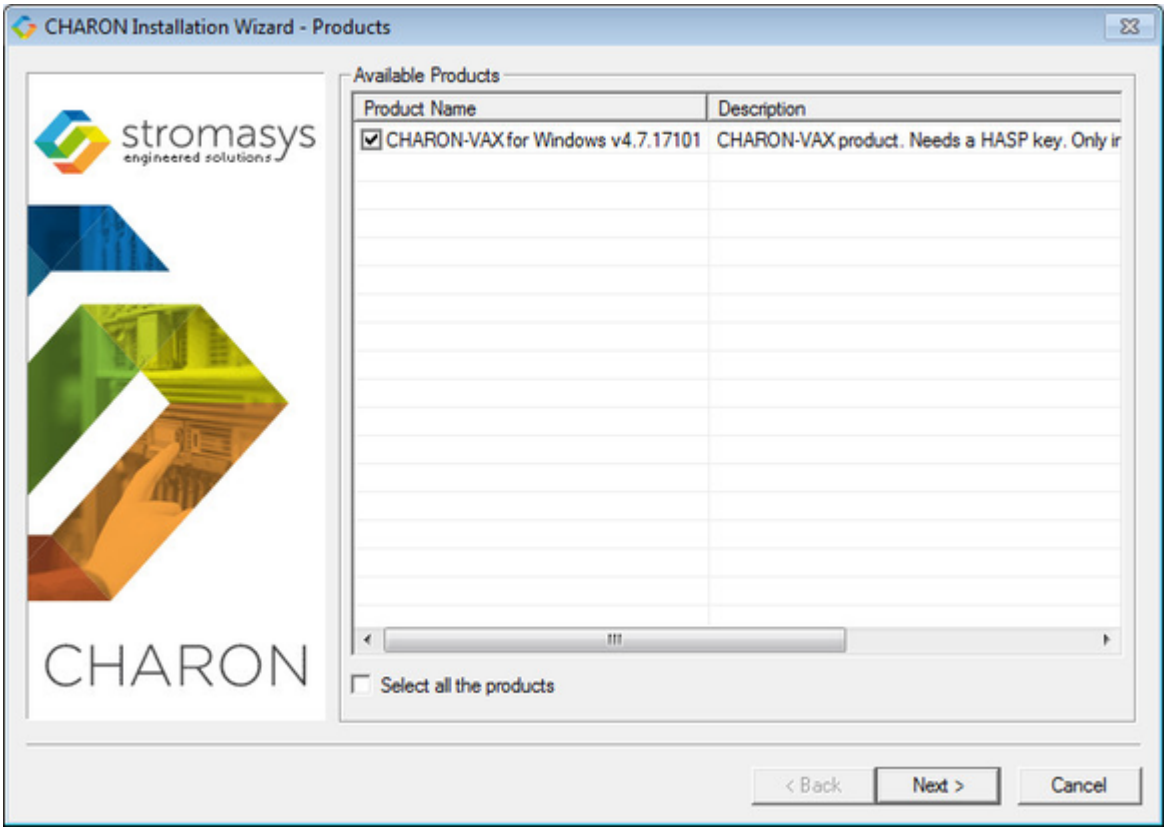
Back to Table of Contents

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.

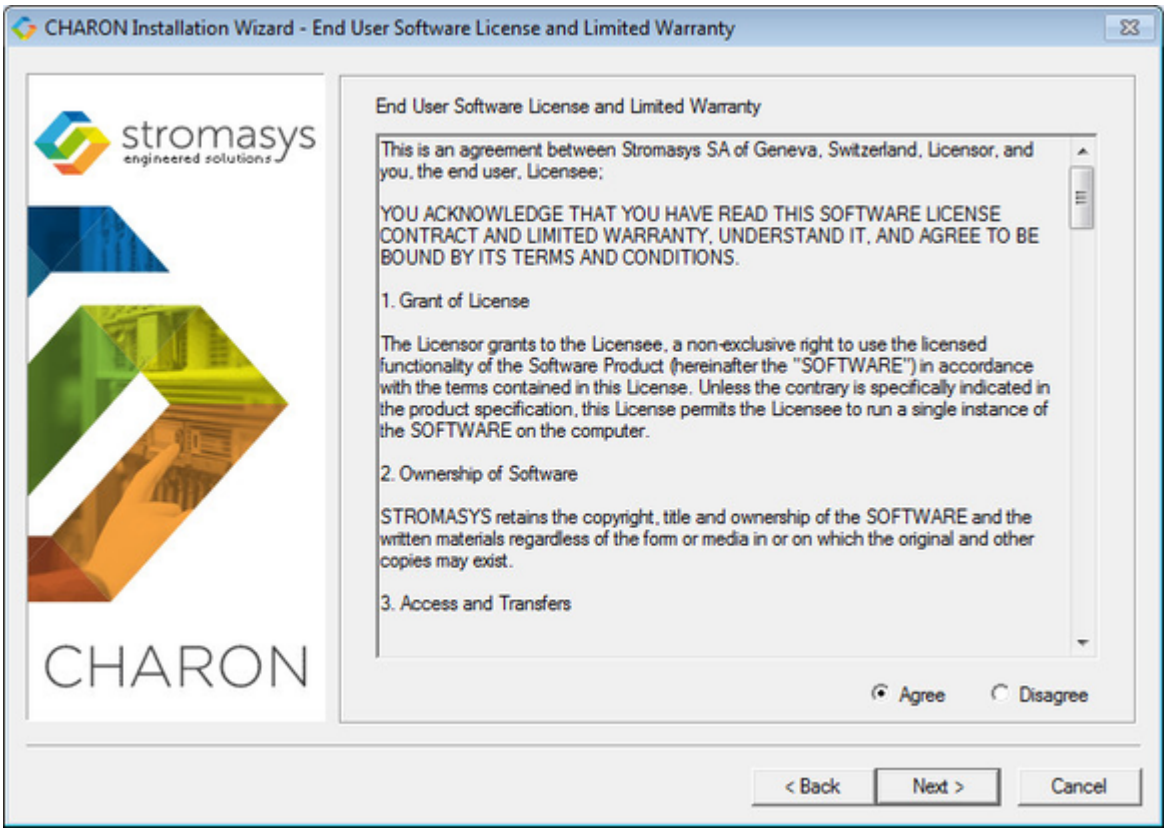
The first dialog lists the CHARON-VAX / CHARON-PDP product in this particular distribution:



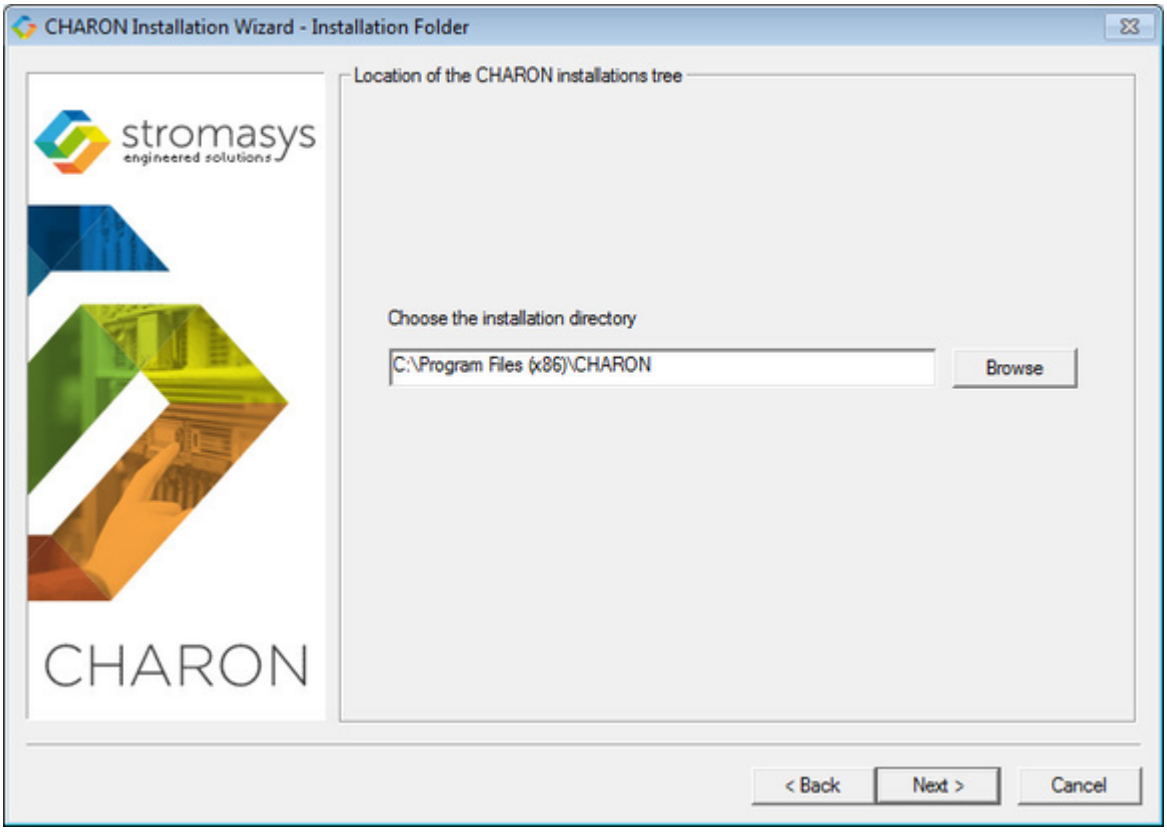
Select the required product or all products by checking the corresponding box.

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

Press the "Next" button.

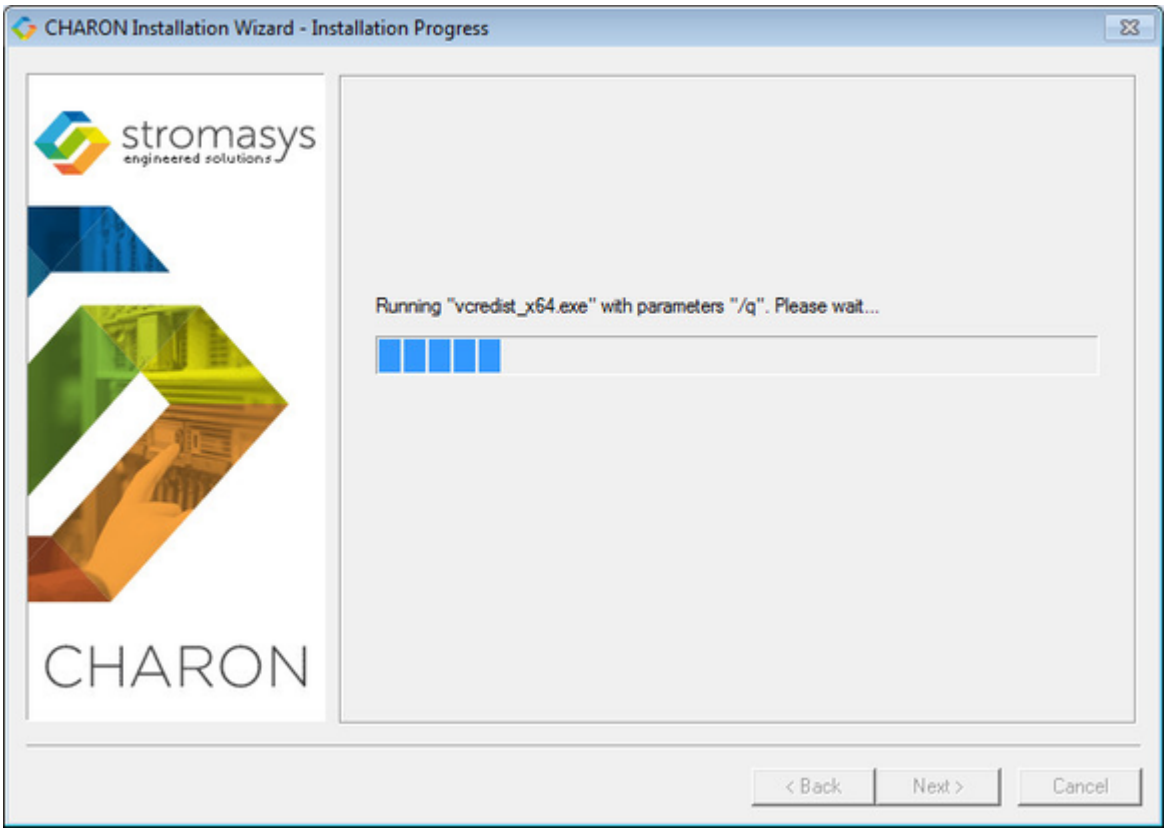


Read the license, check "Agree" if you agree and press the "Next" button.

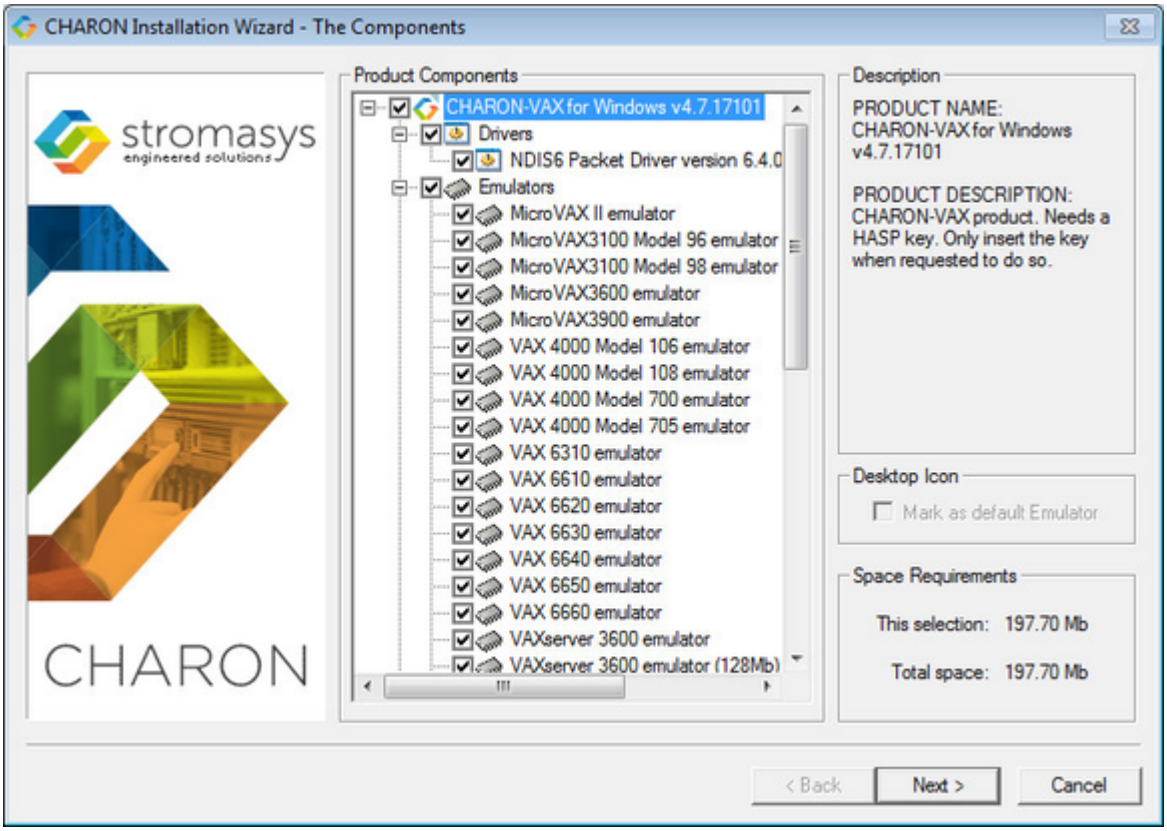


Select a directory to be used for CHARON installation or use the default. The path can either be entered manually or selected using the "Browse" button. Once done, press the "Next" button.

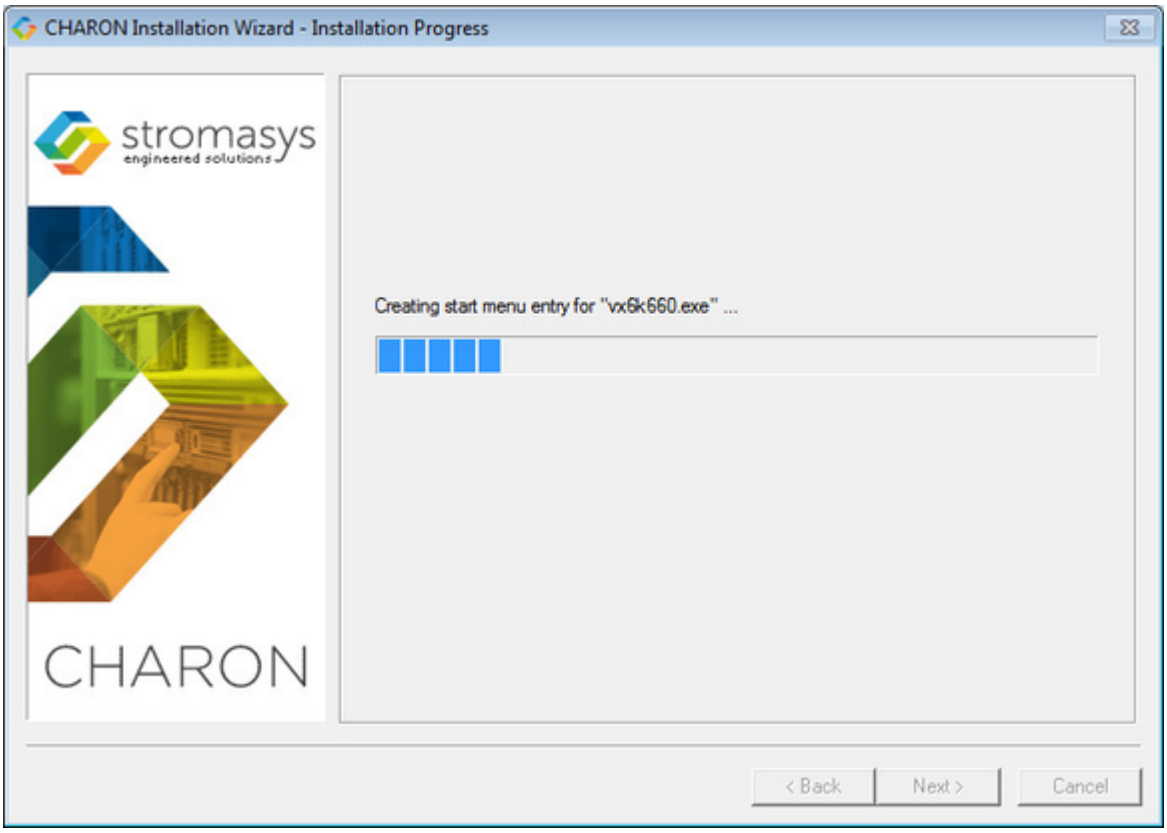
The CHARON installation procedure will install software packages required for CHARON-VAX / CHARON-PDP11, for example, Microsoft Visual C++ redistributives and Sentinel HASP Run-time:



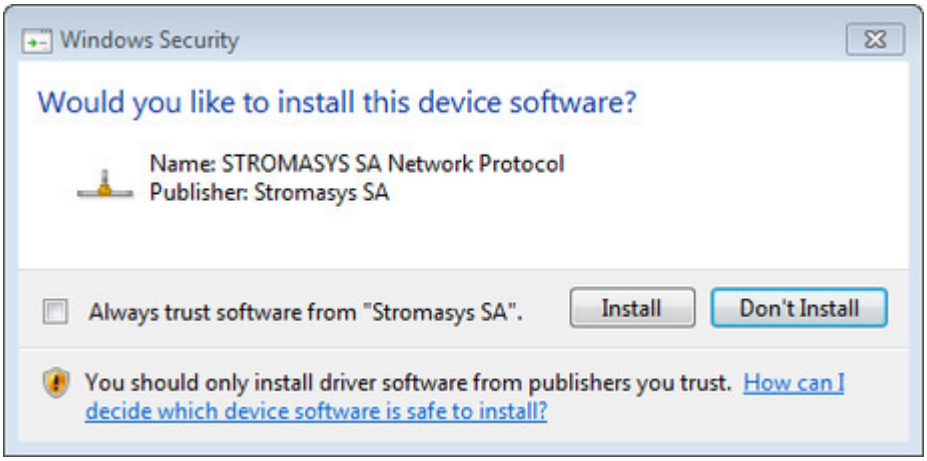
Once the packages are installed, the setup procedure will display a dialog for the chosen CHARON product components:



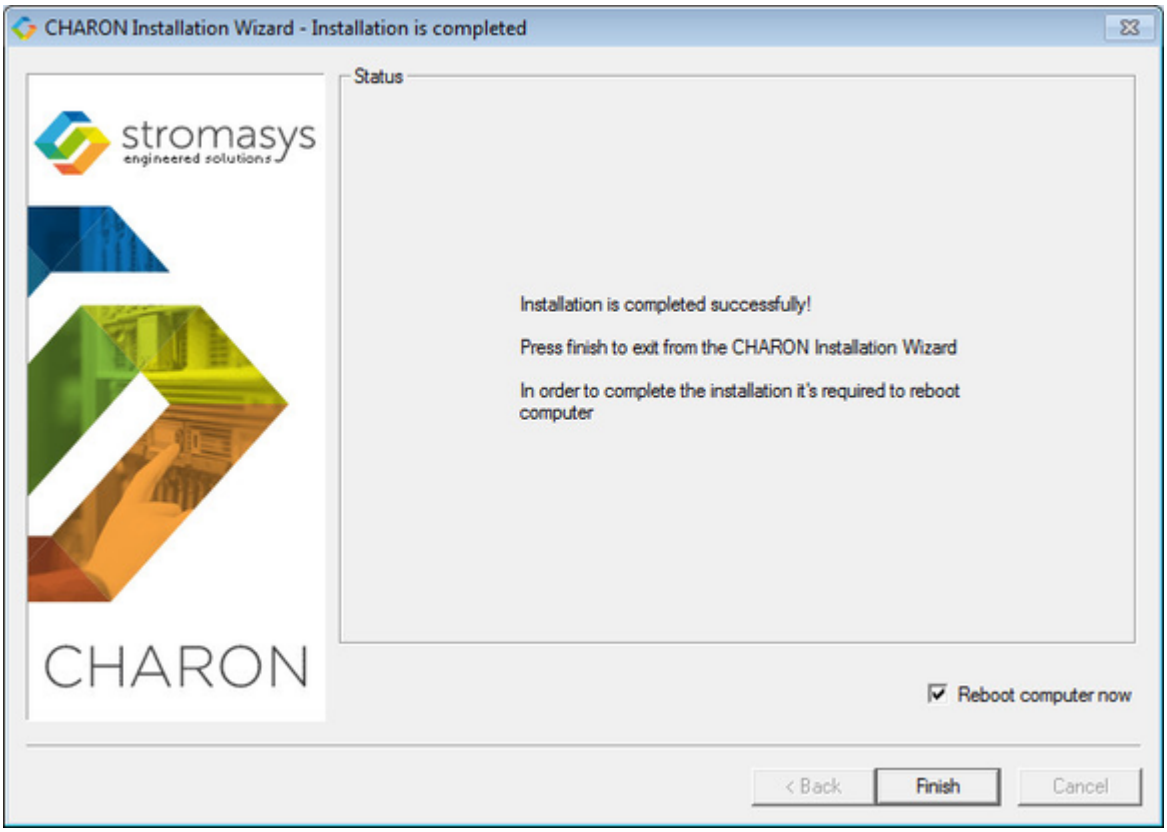
⚠ It is strongly recommended to leave the selection as it is unless you clearly understand what you're doing if you uncheck some modules. Once done, press the "Next" button to start the installation of the selected components.



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



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



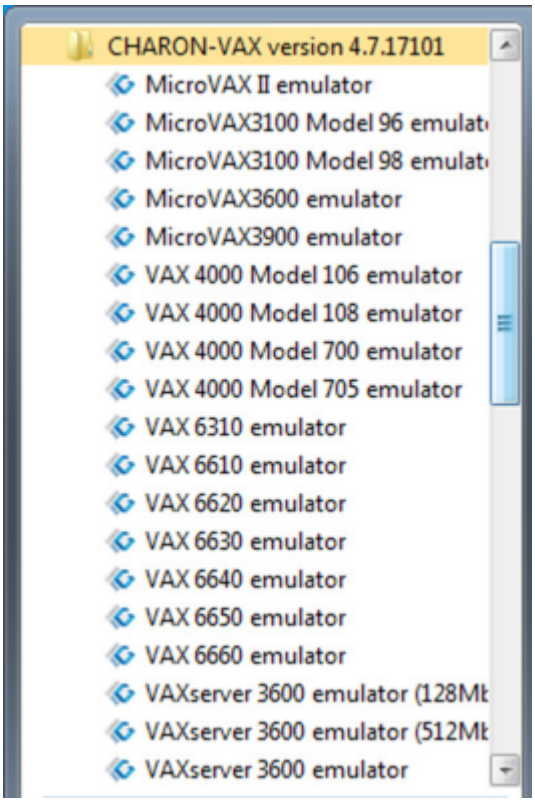
⚠ It is strongly recommended to reboot the CHARON host system immediately: leave the "Reboot computer now" box checked and press the "Finish" button.

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

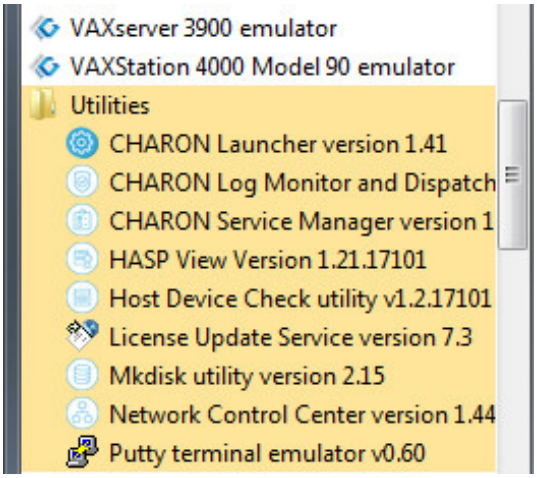
Once the CHARON-VAX / CHARON-PDP license is installed normally (see below) and the configuration file has been modified to map devices, clicking a shortcut will start the corresponding model of VAX/PDP11 emulator.

© Stromasys, 2017

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



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



Back to Table of Contents

© Stromasys, 2017

CHARON-VAX / CHARON-PDP home directory

By default CHARON is installed in the "C:\Program Files (x86)\CHARON" (x64) or "C:\Program Files\CHARON" (x86) directory. It has the following subdirectories:

Directory	Description
Auxiliary	Contains auxiliary material
Build_XXX\x86	Contains product executables, libraries and templates of configuration files
Chapi	Contains CHARON API files used for creating new emulated devices
Drivers	Contains CHARON drivers
HASP_X.XX	Contains Sentinel HASP Run-time installation (already installed)
InstallShell	Contains CHARON installation procedure executables
Logs	Contains CHARON installation log
Redistributables	Contains redistributables needed for CHARON running (already installed)
Utilities_X.X.XXXXX\x64	Contains CHARON utilities designed for Windows x64
Utilities_X.X.XXXXX\x86	Contains CHARON utilities designed for Windows x86
Virtual Disk Images	Contains CHARON virtual disk images, for example "idle_vms_pkg_v2.0"

The "Build_XXX\x86" directory contains template configuration files with examples of typical configuration parameters and commentaries. The template files are used to build the emulator instance. We will pay our attention to this subject in the next chapter.

Back to Table of Contents

License installation

Regular HASP USB dongle

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

> If the CHARON host is accessed remotely, please note that the contents of a regular HASP license cannot be displayed using RDP. ILO or iDRAC or some other console-capable program must be used.
>
> Also, CHARON cannot be started manually from an RDP connection. In this case, a workaround is possible to install CHARON as a service. This procedure will be described later.

Network HASP USB dongle

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

In case of remote usage:

- Copy "hasp_install\Sentinel_LDK_Run-time_cmd_line_X.XX.zip" file from CHARON distributive to the server in any directory, for example "C:\Temp"
- Extract the 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

© Stromasys, 2017

Example:

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

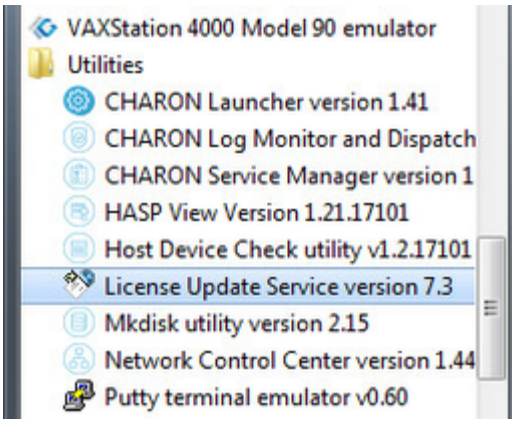
- Connect the network HASP dongle to the server USB port.

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

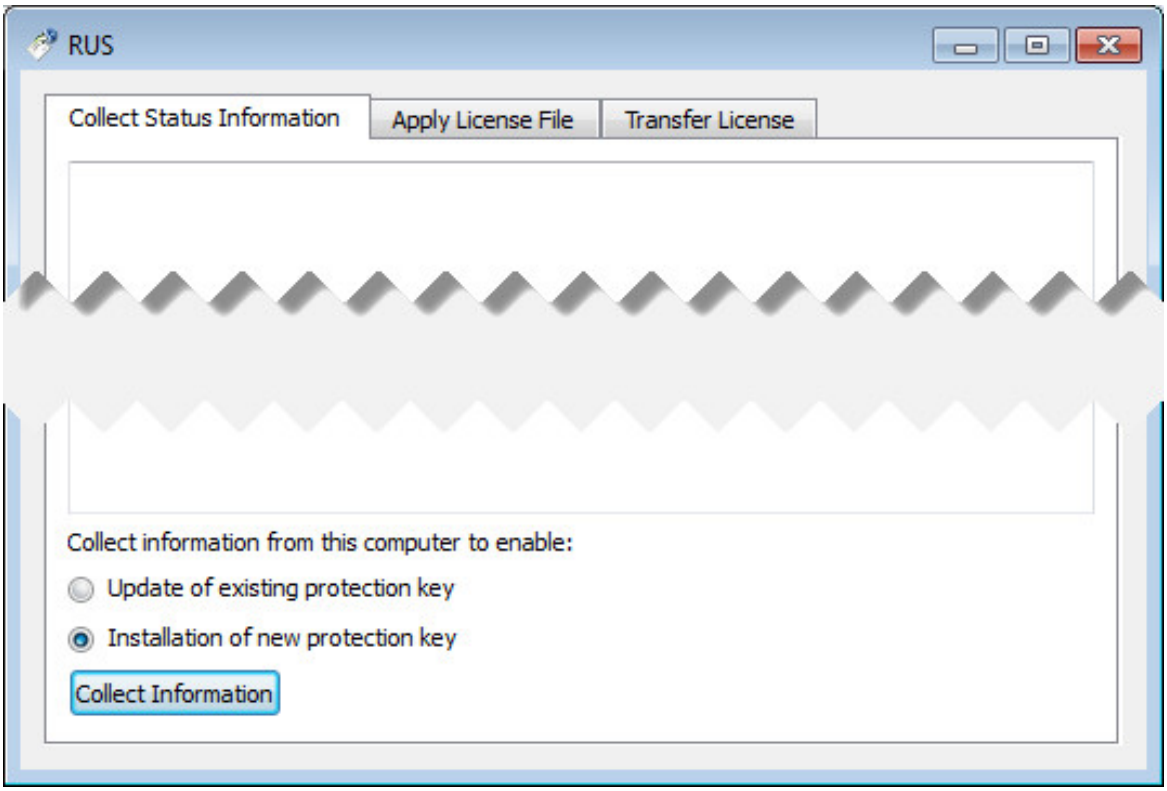
Software license

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

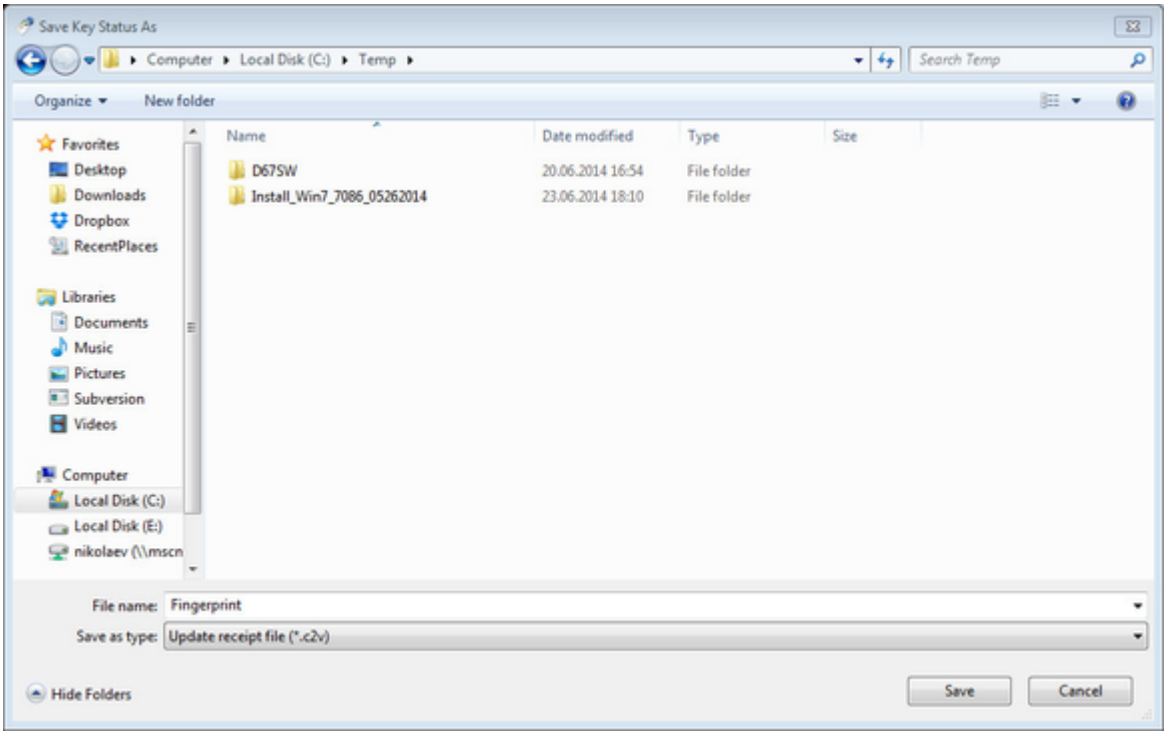
1. Press the Start button, select "All programs", "Charon" then the Charon product/version folder, "Utilities" and execute the "License Update Service " program:



The following default dialog window will appear:

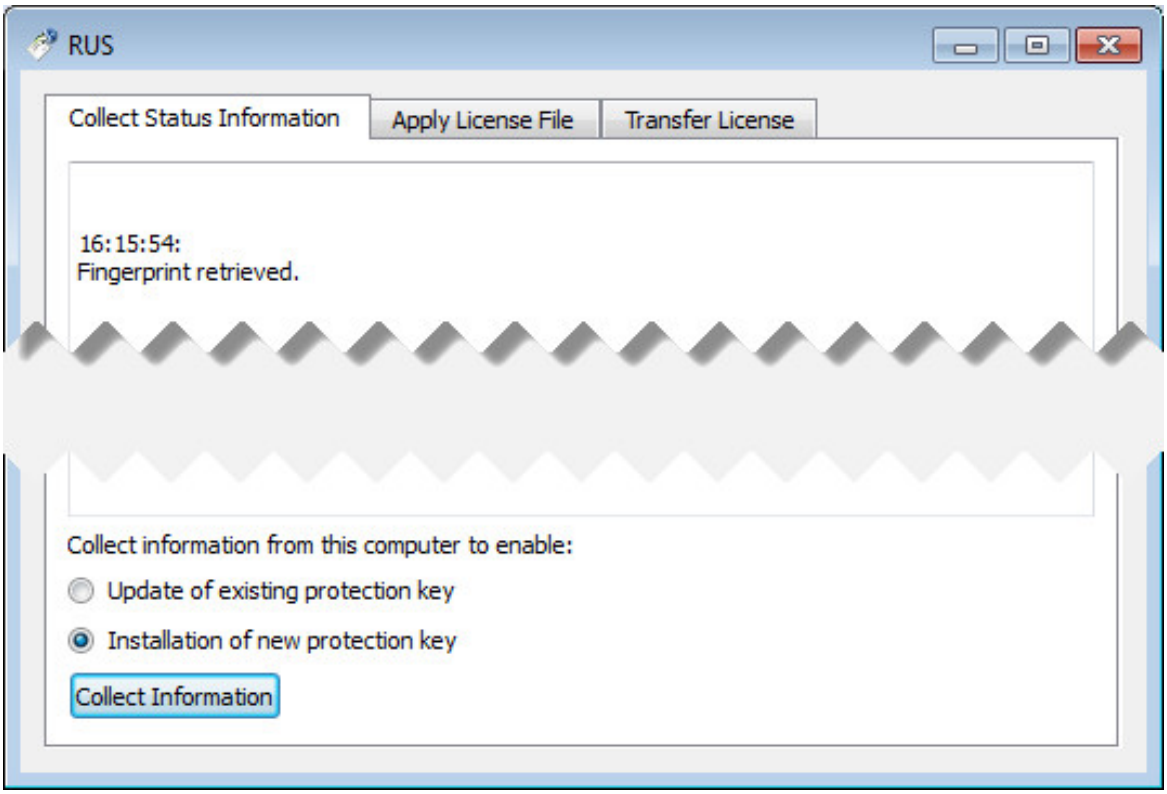


2. Select the "Installation of new protection key" radio-box and press the "Collect Information" button. The following window will appear:

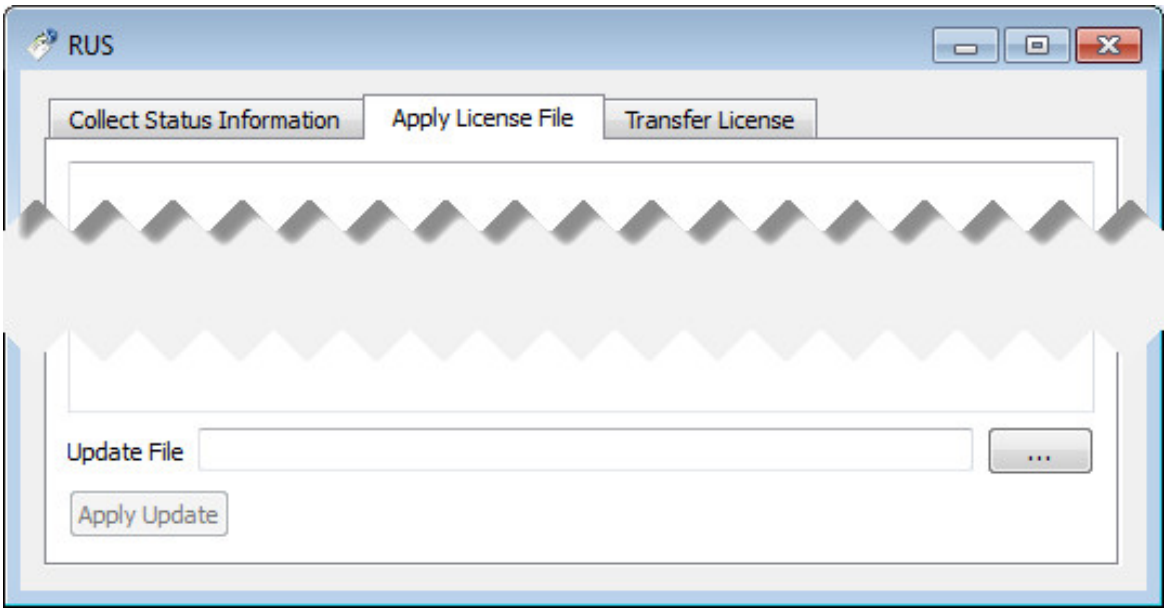


Specify the directory and name of the system "fingerprint" file to be created; press "Save".

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

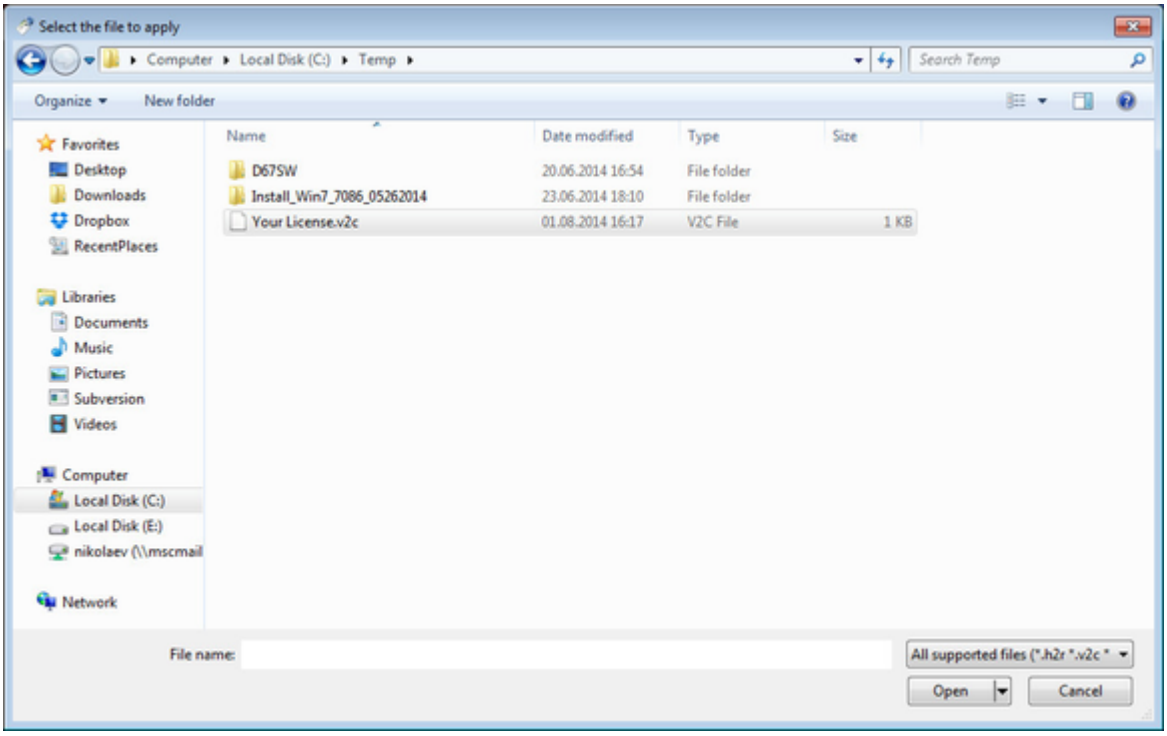


4. Send the resulting file to STROMASYS. In return STROMASYS will provide you with a "*.v2c" file, for example "Your License.v2c"
5. Copy the received v2c file to CHARON host in the folder of your choice and select the "Apply License File" tab:



© Stromasys, 2017

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



   Press "Open" to apply the license.

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

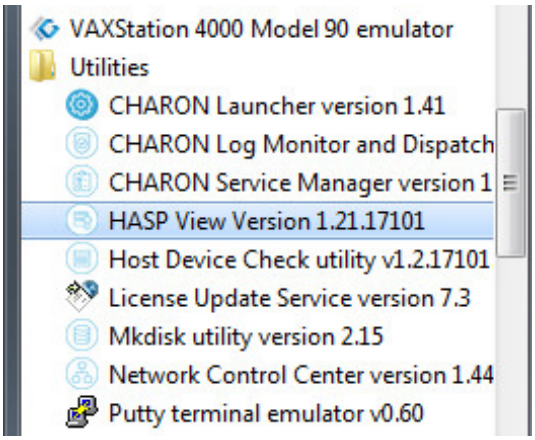
> Network-wide software licenses have no restrictions with remote access, whereas regular software licenses cannot be displayed and used in this case

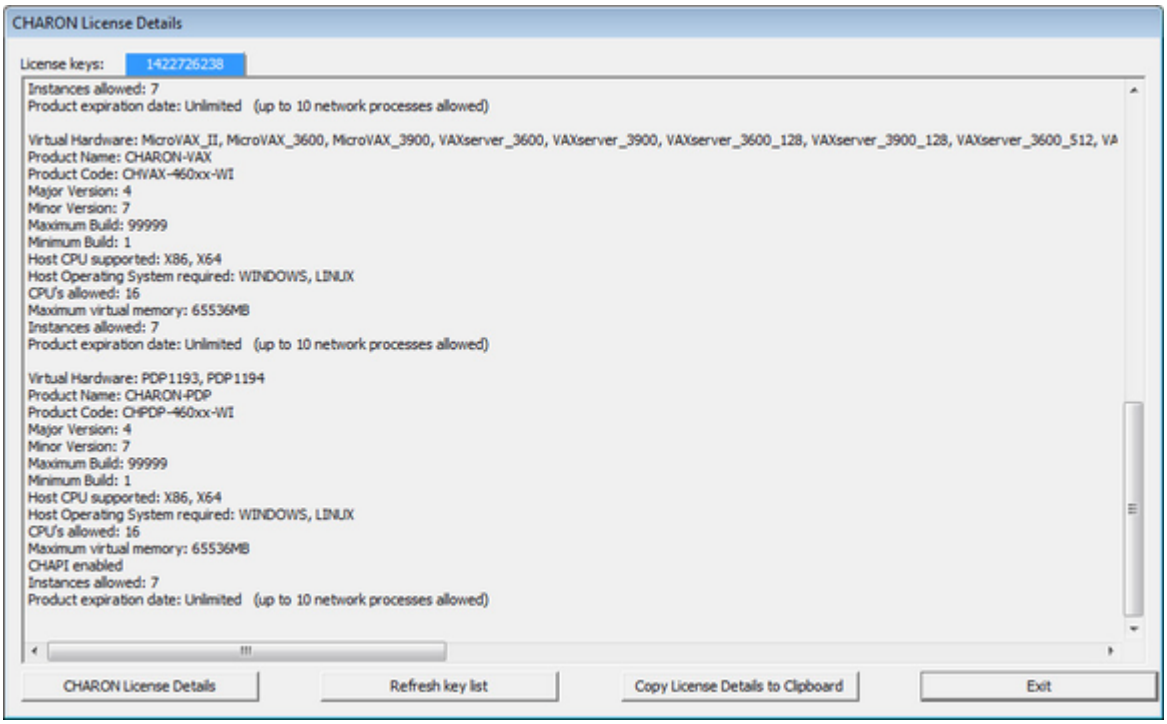
> A so-called "Provisional" (demo) license does not require a collected fingerprint. For its installation, proceed directly with the action (5) of the sequence above

Back to Table of Contents

License validity verification

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





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

© Stromasys, 2017

Troubleshooting

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

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

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

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

If several License keys / SLs are displayed remove all of them and leave only the one provided by STROMASYS for the just installed version of CHARON.moving licenses can be done by physical disconnection of the corresponding USB HASP keys from CHARON host and physical disconnection of the network HASP keys from all hosts on a local network (or by disabling remote access to network licenses from CHARON host - see detailed explanations below).

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

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

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

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

Example:

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

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

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

CHARON-VAX / CHARON-PDP checks its licenses from time to time starting with the main license. If it is not accessible, CHARON tries to access the backup license.

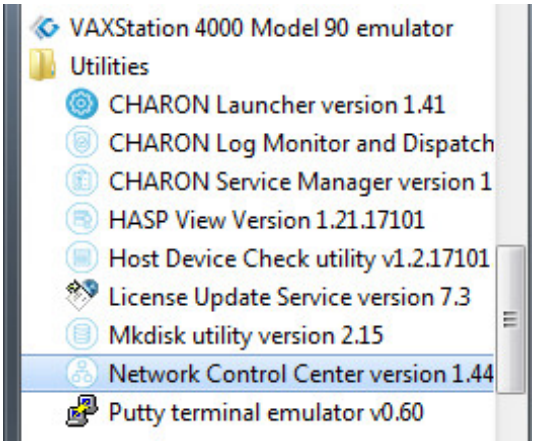
Back to Table of Contents

Network configuration

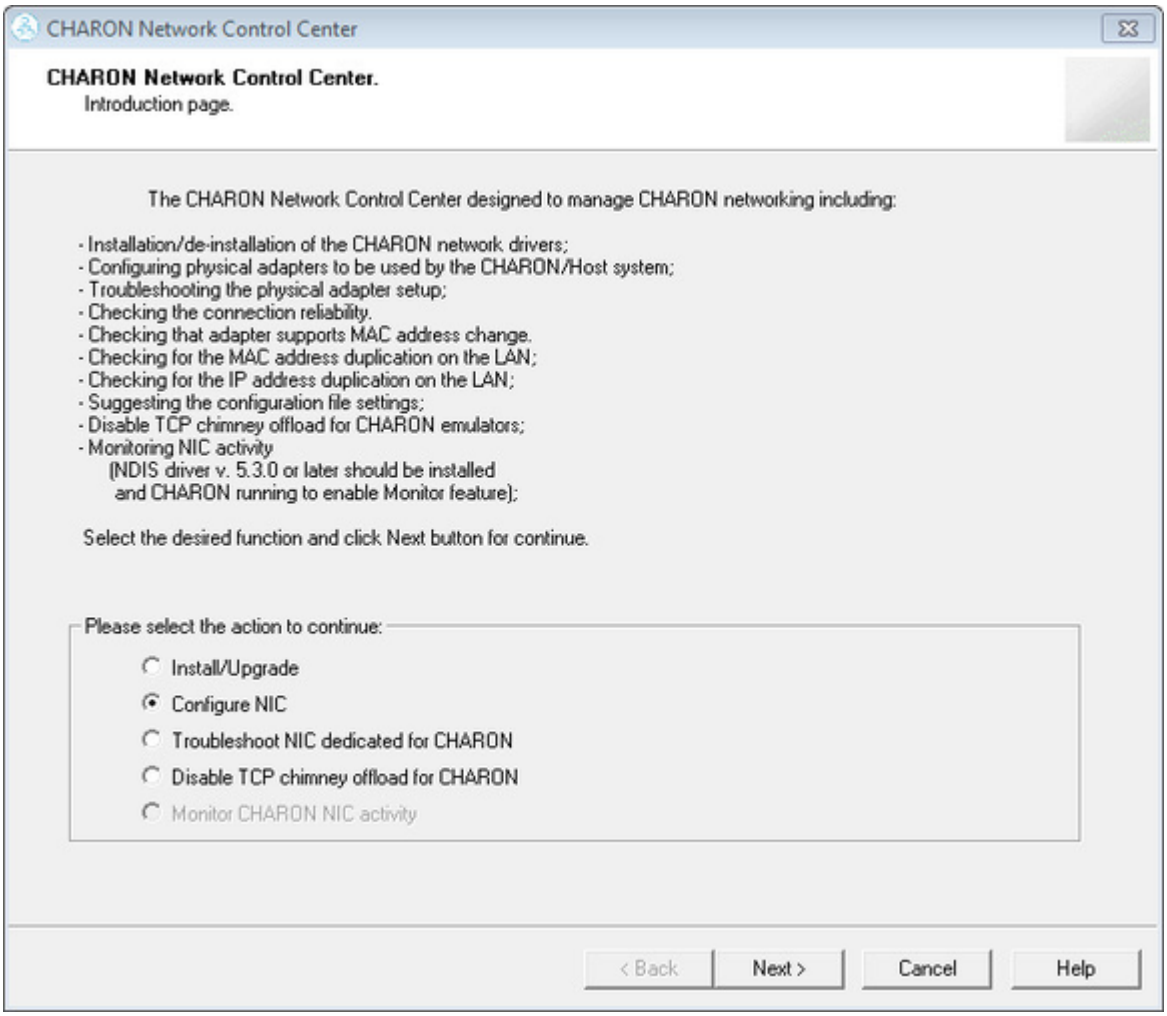
In most cases, CHARON will use a network. If so, CHARON requires a dedicated network interface cleared from any other protocols including TCP/IP.

© Stromasys, 2017

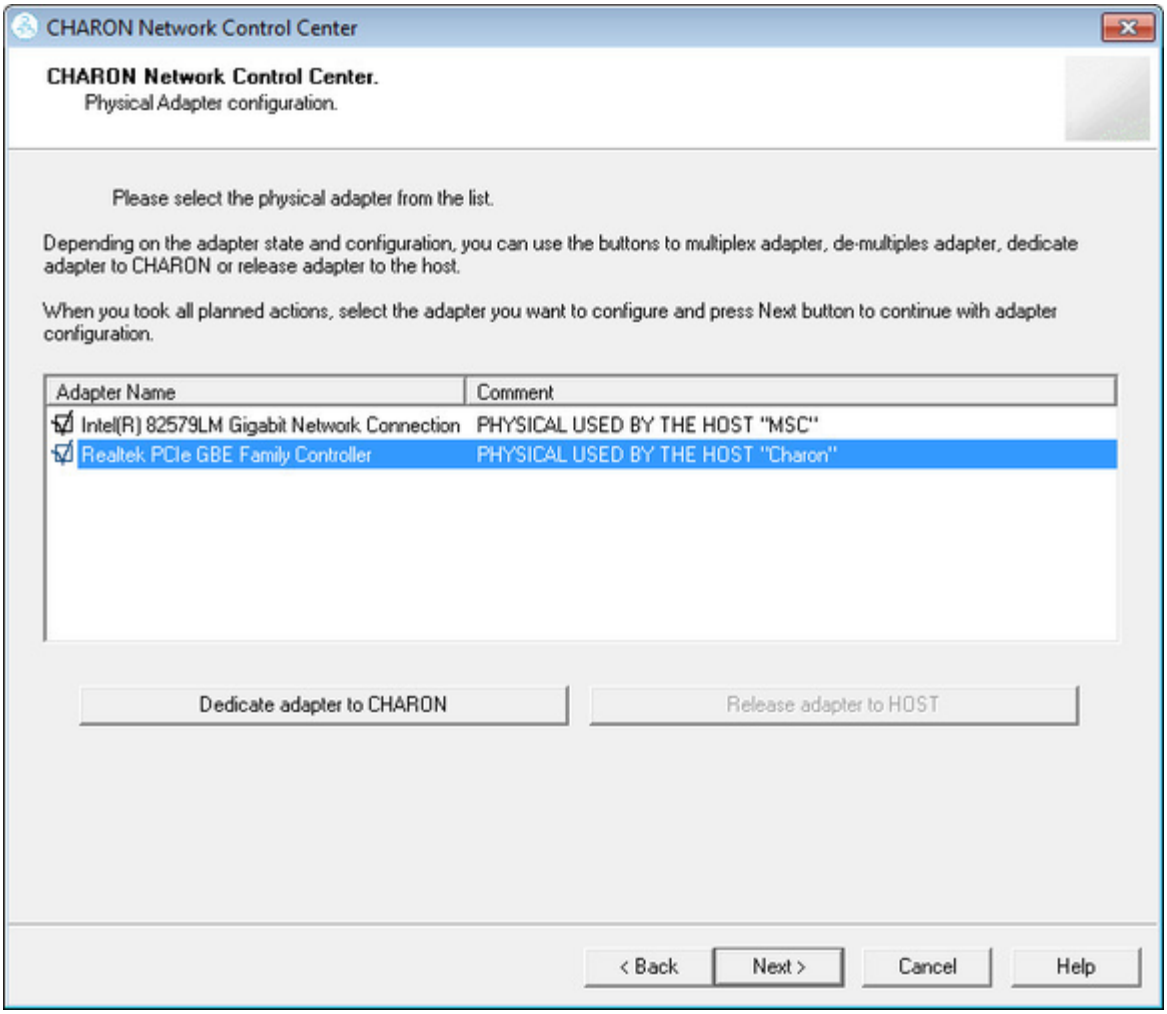
Network configuration is done with the help of the "Network Control Center" (NCC) utility:



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



Press the "Next" button. The following dialog will appear:

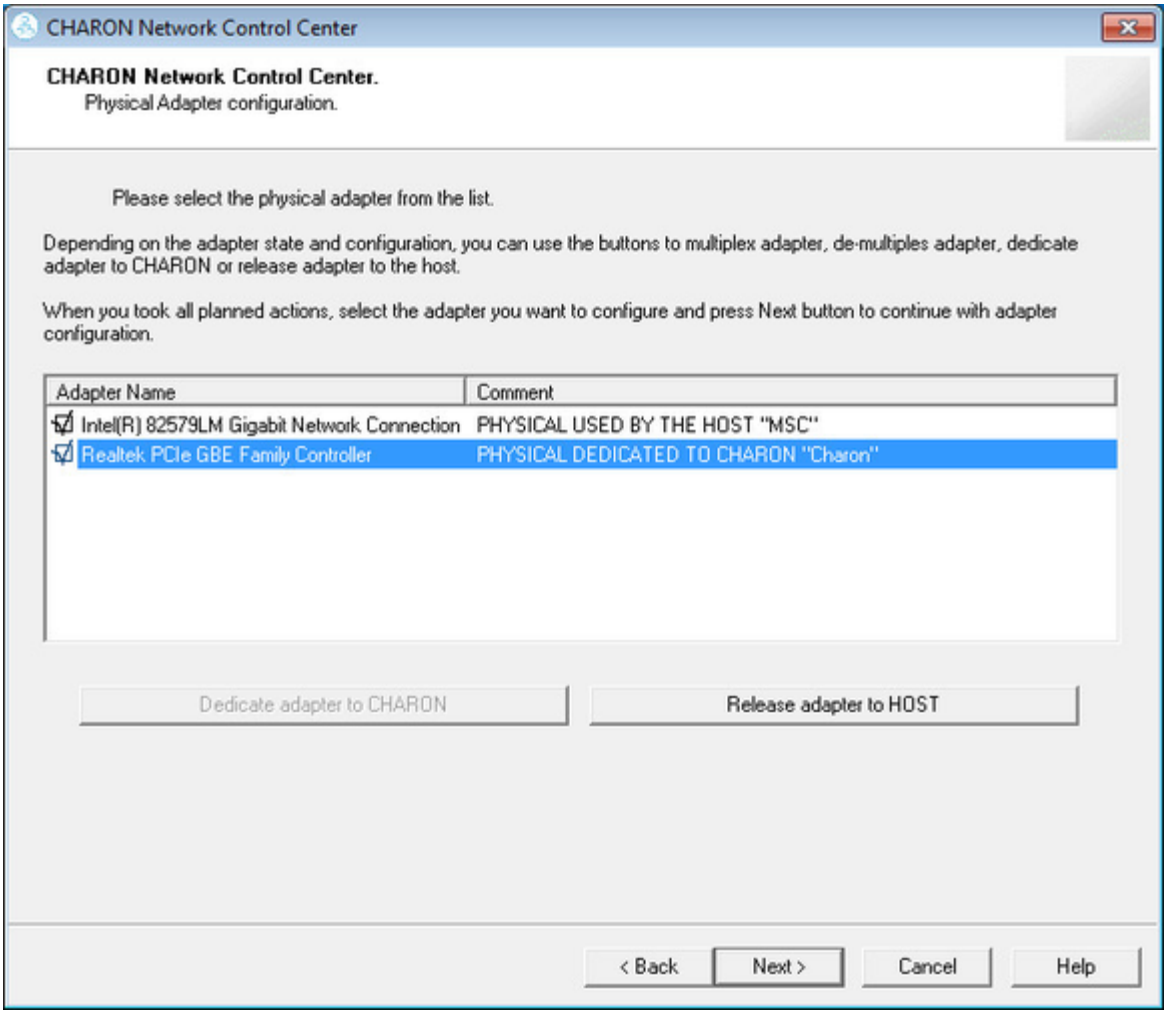


Select the interface to be dedicated to CHARON (in our example it is Realtek PCIe GBE Family Controller with a name "Charon") and press the button "Dedicate adapter to CHARON".

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

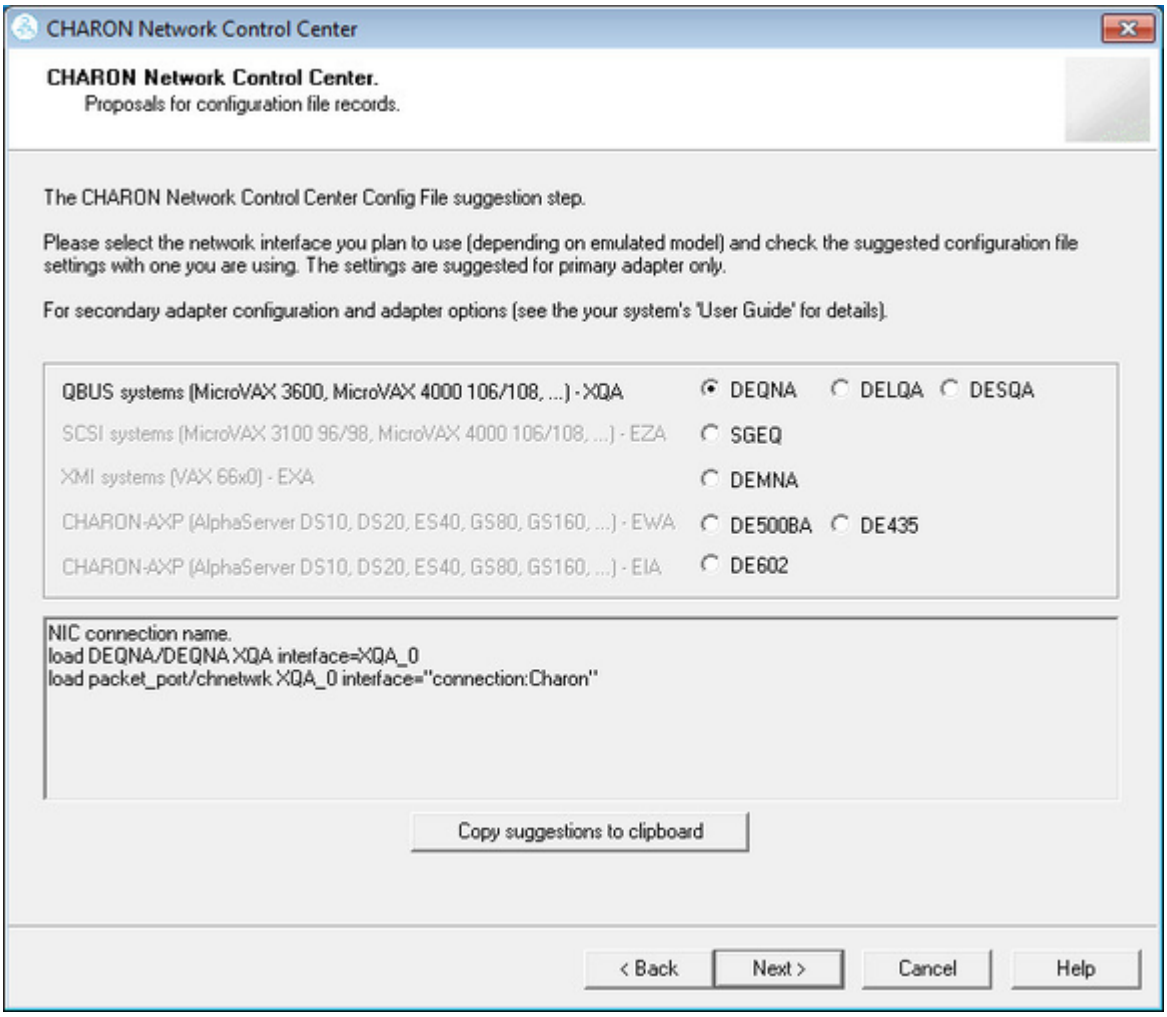
© Stromasys, 2017

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



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

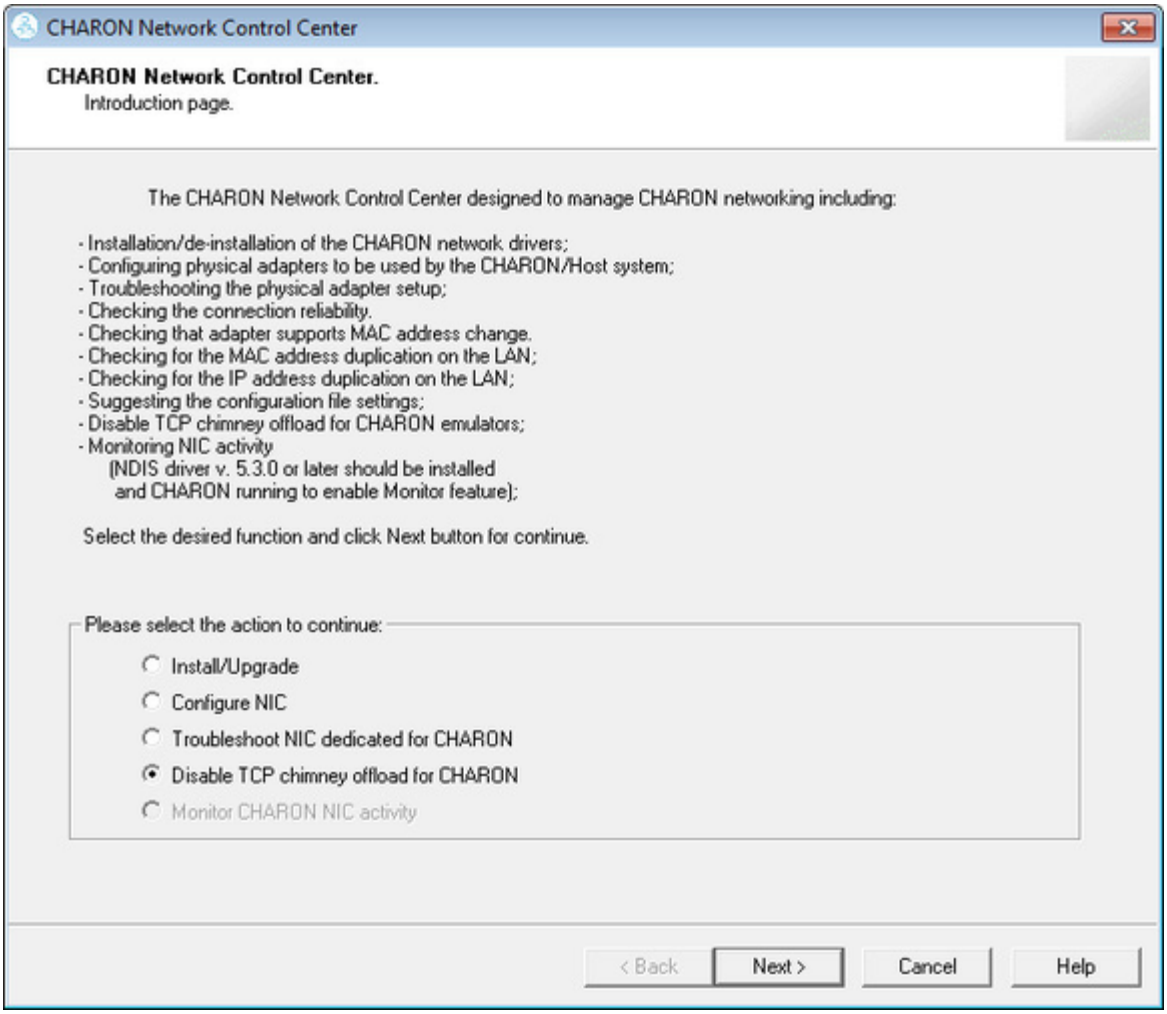
The next step displays text to be included in the CHARON configuration file for the interface dedicated to CHARON. Select the target interface and press the "Next" button. The following dialog will appear:



Selecting the target emulated network interface displays the configuration lines specific for the selected CHARON host network interface. Press the "Copy suggestions to clipboard" button to copy the suggested configuration lines. They can be pasted into the CHARON-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 the "Next" button to apply the "offload" parameters settings.

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

If you've installed CHARON-VAX Version 4.7 Build 171-01, please do not use the "Disable TCP chimney offload" feature since it may lead to some critical problems relevant to CHARON license checking.This issue is solved starting with Build 171-02 (patch).

If for any reason you have already used this feature proceed with the following:

1. Run CMD with administrative privileges
2. Issue the command: **netsh interface tcp show chimneyapplications**
3. Then for each installed CHARON executable from the list run the following command: **netsh inter tcp delete chimneyapplication "C:\<full path to CHARON executable.exe>"**

⚠ 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.

More information and details on disabling TCP chimney offload...

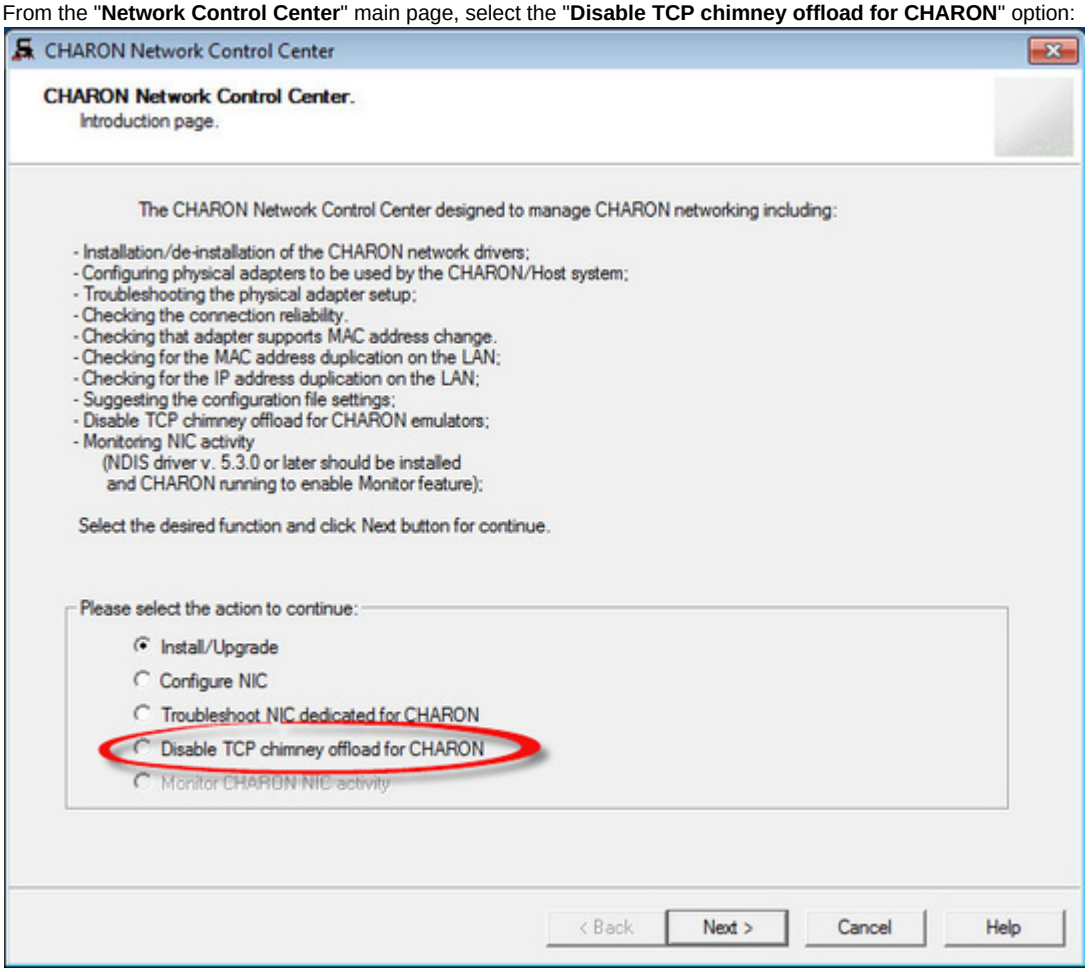
For Windows 2008 R2, Windows 2012 R2, Windows 7, 8 and 8.1

TCP Chimney offload must be disabled.

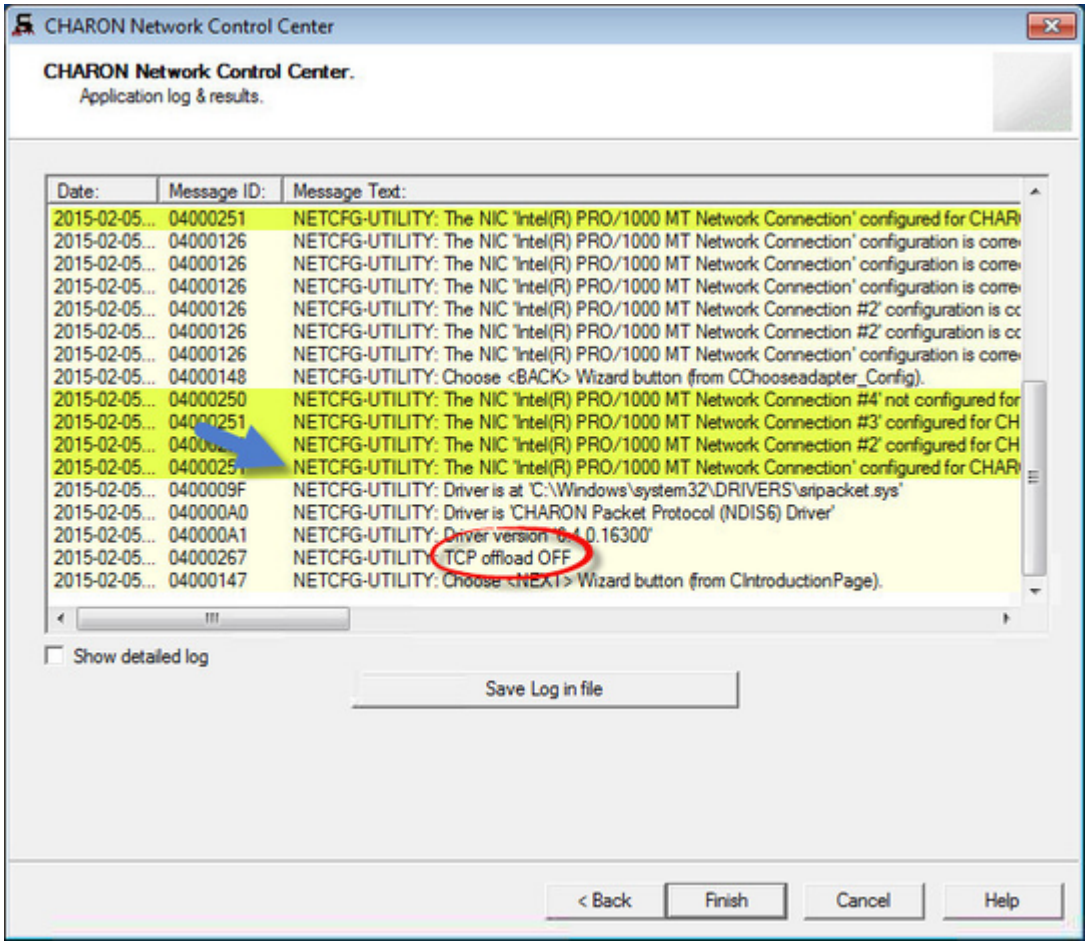
TCP Chimney Offload is a networking technology that helps transfer the workload from the CPU to a network adapter during network data transfer.

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

- Starting with CHARON-VAX V4.6 and CHARON-AXP V4.6, the Network Control Center Utility allows you to disable TCP Chimney offload:
  - 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. In the CHARON versions prior to 4.8 open the **Start** menu, select "**All Programs**" then "**CHARON**", select your Charon product version, "**Utilities**" then "**Network Controler Center version *x.xx***".
  - From the "**Network Control Center**" main page, select the "**Disable TCP chimney offload for CHARON**" option:



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



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

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

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

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

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

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

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

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

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

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

For older versions of Windows

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

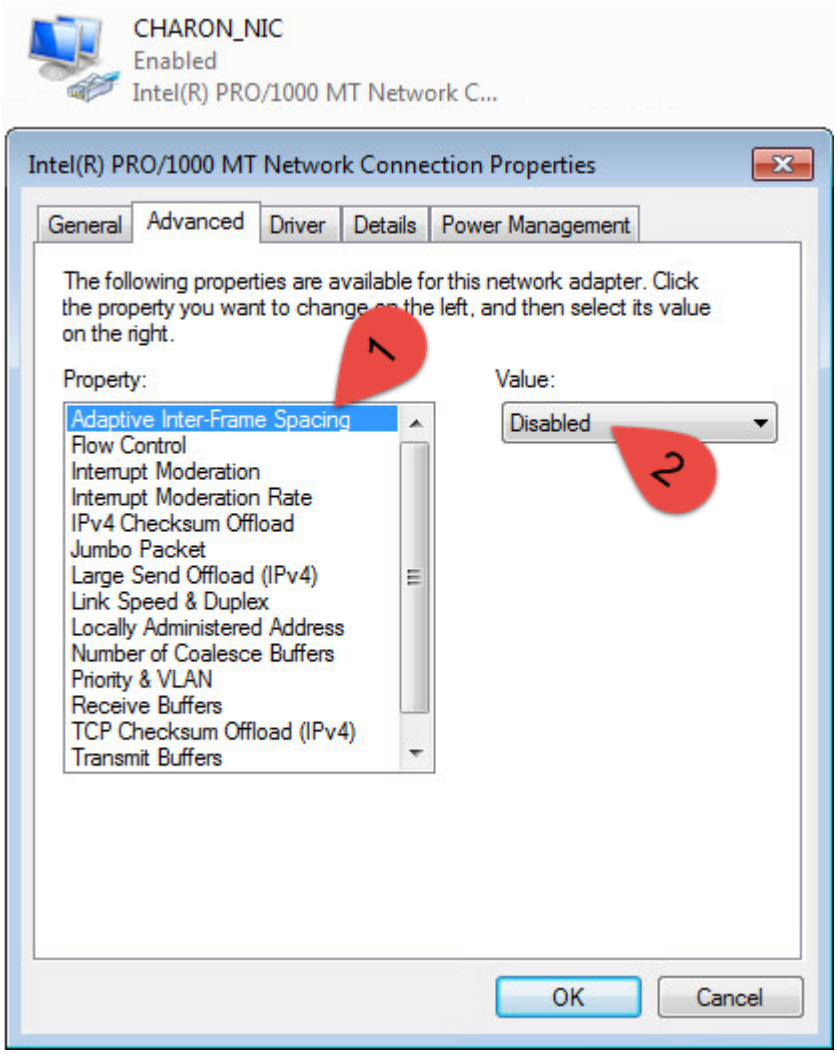
- Adaptive interframe spacing
- Flow control
- Interrupt moderation
- Interrupt moderation rate
- Ipv4 checksum offload
- Jumbo packet
- Large send offload
- Disable Priority and VLAN
- TCP checksum offload
- UDP checksum offload

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

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

© Stromasys, 2017

Example:



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

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

Back to Table of Contents

After installation

If you plan to use local system administrator ("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 CHARON host.

Otherwise it is possible to use 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.

Back to Table of Contents

© Stromasys, 2017

Running CHARON-VAX and CHARON-PDP for Windows

Table of Contents

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

Introduction

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

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

Example:

```
...
set session log="C:\Charon Instances\First\mv3k6.log"
set toy container="C:\Charon Instances\First\mv3k6.dat"

load RQDX3/RQDX3 DUA
set DUA container[0]="C:\Charon Instances\First\mv3k6_boot_disk.vdisk"
...
```

Please refer to the next chapters for more details concerning CHARON-VAX / CHARON-PDP configuration details.

Back to Table of Contents

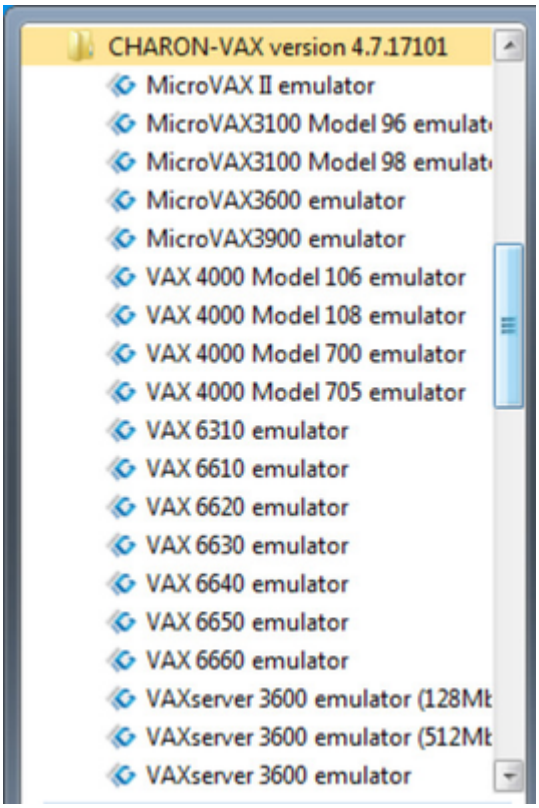
Running from a Desktop shortcut

The CHARON installation procedure creates shortcuts on the desktop for so-called "default" VAX/PDP11 model and shortcuts under "Start" menu for all VAX/PDP11 models included during installation.

© Stromasys, 2017

Desktop shortcut:

"Start" menu shortcuts:

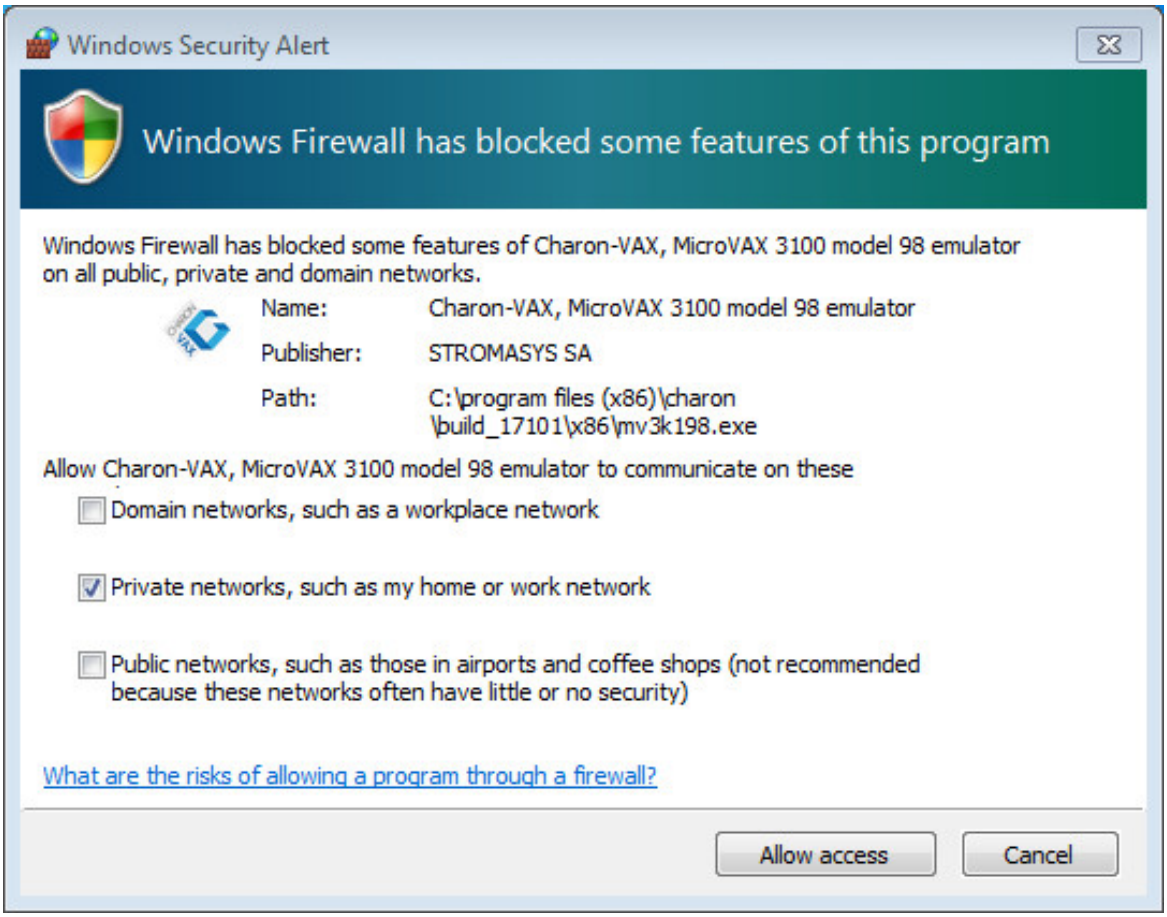


To run a certain CHARON-VAX / CHARON-PDP model, double click the target model's desktop shortcut (or click on "Start" menu shortcut).

> By default the CHARON sample configuration files do not contain any specification of disks and network; you need to update a copy of the sample configuration file to specify your specific disk and network configuration.

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

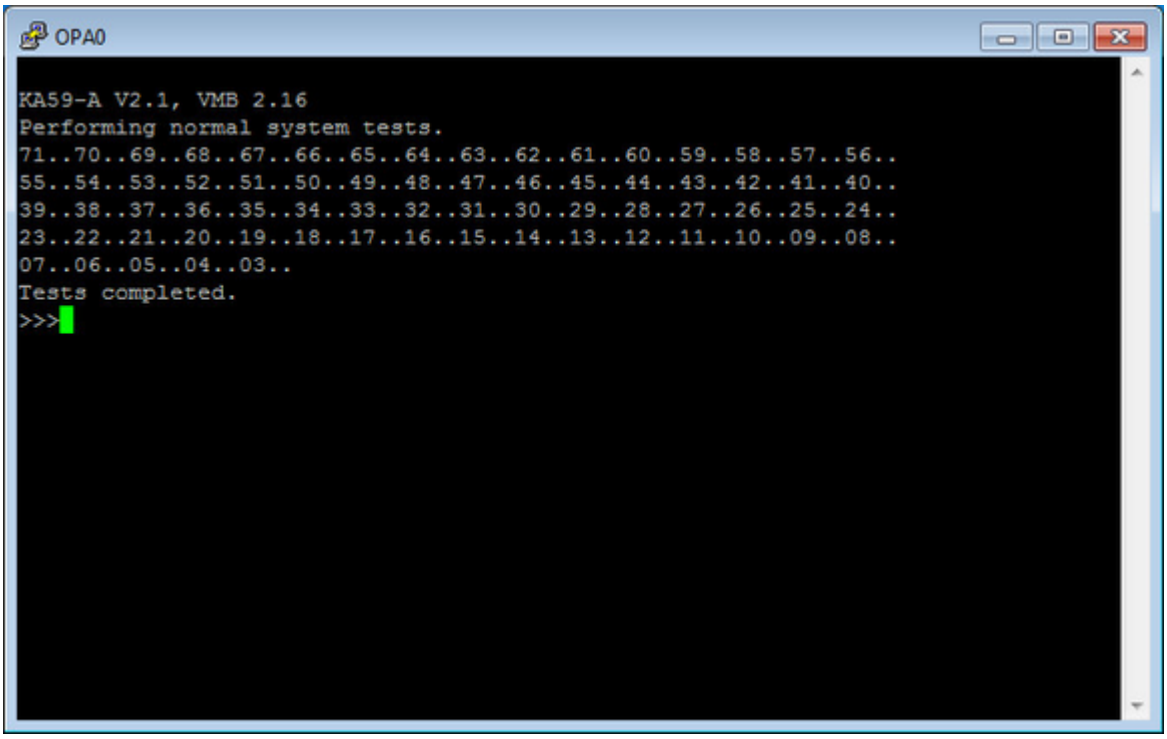
© Stromasys, 2017



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

The CHARON icon will appear in the system tray menu:

Next, the VAX/PDP11 emulated console will appear:



If the CHARON icon in the system tray menu disappears, it means a critical error happened at run-time. In this case you have to analyze the CHARON log file, by default stored beside CHARON executables. The default location for log files is "C:\Program Files (x86)\CHARON\Build_XXX\x86" (x64) or "C:\Program Files\CHARON\Build_XXX\x86" (x86). The name of the log file depends on the model of emulated VAX/PDP11 and the date and time, for example "MicroVAX_3100_Model_98-2015-08-05-14-40-51-000000000.log"

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

Generally it is not recommended to use template (sample) configuration files (used in shortcuts) for your configuration since those files are part of the installation and may be deleted by the CHARON installation procedure on modifications or re-installation. It is a good practice to copy the required configuration template from "C:\Program Files (x86)\CHARON\Build_XXX\x86" (x64) or "C:\Program Files\CHARON\Build_XXX\x86" (x86) directory to some local file and build the configuration from there.

Once the sample configuration files are copied, the CHARON default shortcut (created by the installation procedure) can be changed to use your target configuration file:

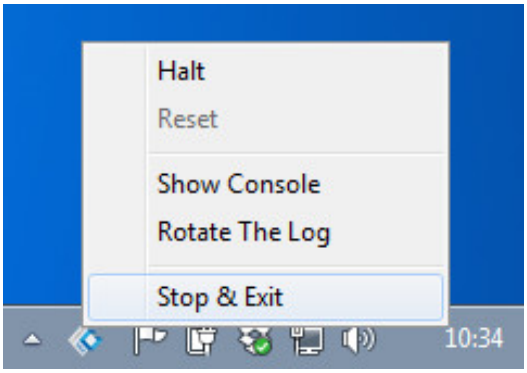
1. Right click on CHARON shortcut for target VAX/PDP11 model
2. Choose "Properties"
3. In the "Target" edit box, change the default template configuration file path and name to the local one. For example "C:\My configurations\my_mv3k6.cfg"
4. Press "Apply" button

The next step is to install or copy a system disk for CHARON operation. This can be either installation of a new VMS, RSX11, RT11, etc. system using a distributive provided by HP or a transfer of data from an existing VAX/PDP11 system. These possibilities will be discussed in detail in the next chapters.

© Stromasys, 2017

To exit from CHARON-VAX / CHARON-PDP emulator use the following methods:

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



- **Configuration file updated to enable "F6" button in configuration file to trigger exit from CHARON:**

```
#--------------------------------------------------------------------------
#
# Uncomment to allow 'F6' to terminate the running emulator.
#
#--------------------------------------------------------------------------

set OPA0 stop_on = F6
```

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

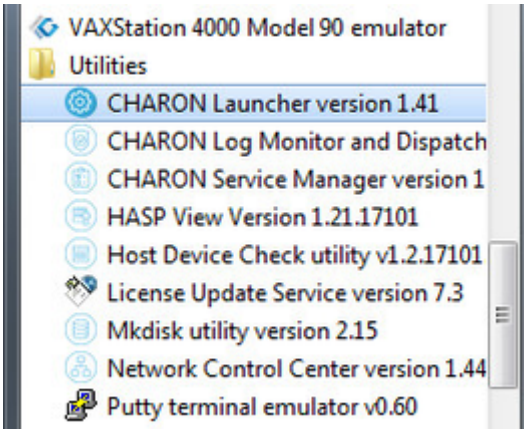
⚠ Before stopping the CHARON-VAX / CHARON-PDP service, shutdown the operating system running in CHARON-VAX / CHARON-PDP.

Back to Table of Contents

Running from the Launcher utility

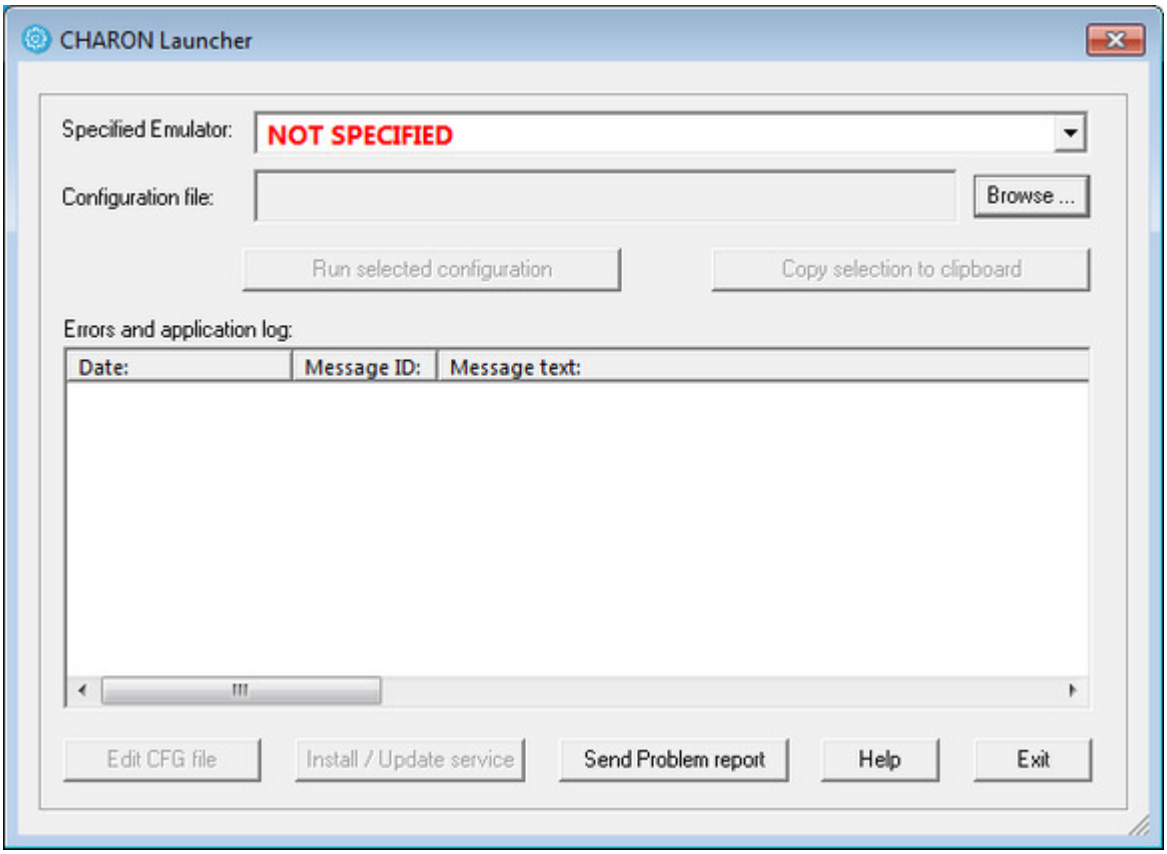
Running from the "CHARON Launcher" utility is very convenient during the debugging of your CHARON configuration. The utility shows CHARON log updates in real time. If RDP is used to connect to the CHARON host, the CHARON may not see the license; start CHARON as a Service instead. From the Launcher, CHARON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

Start the Launcher:

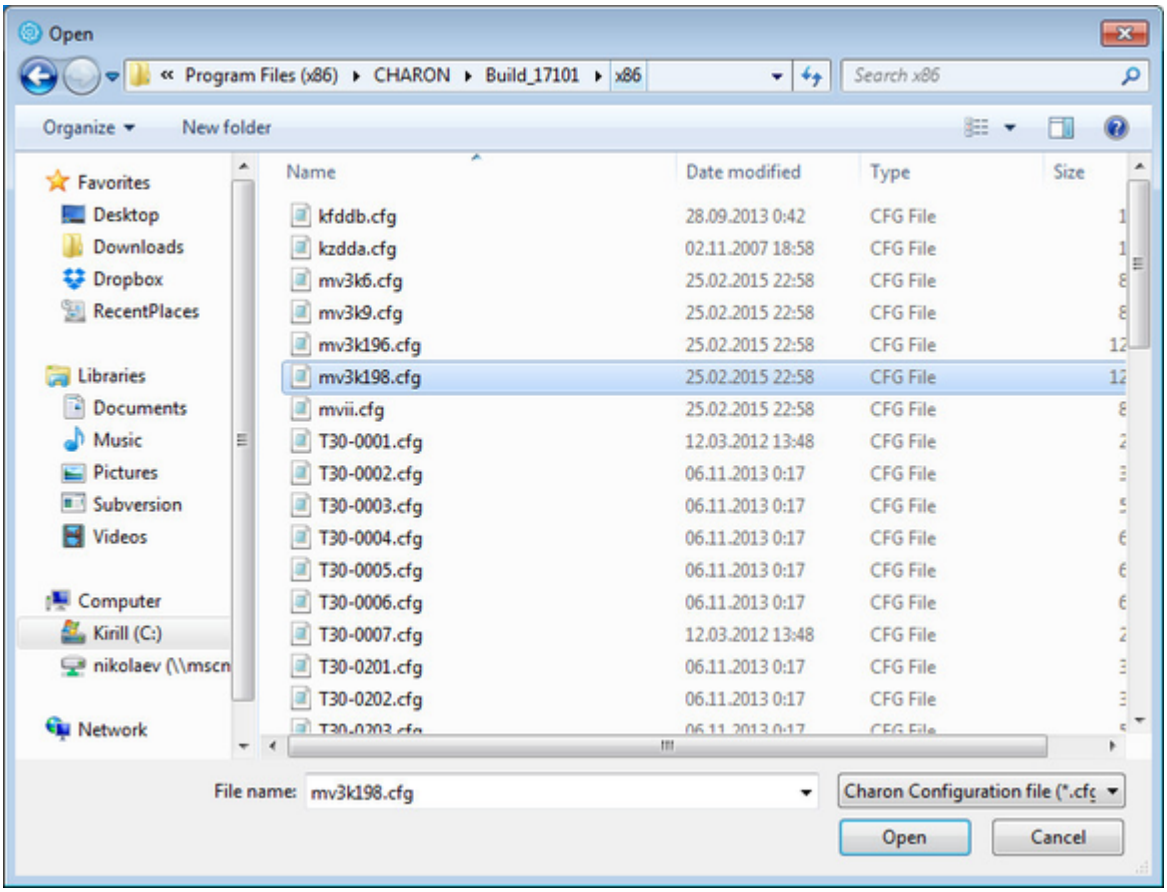


© Stromasys, 2017

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

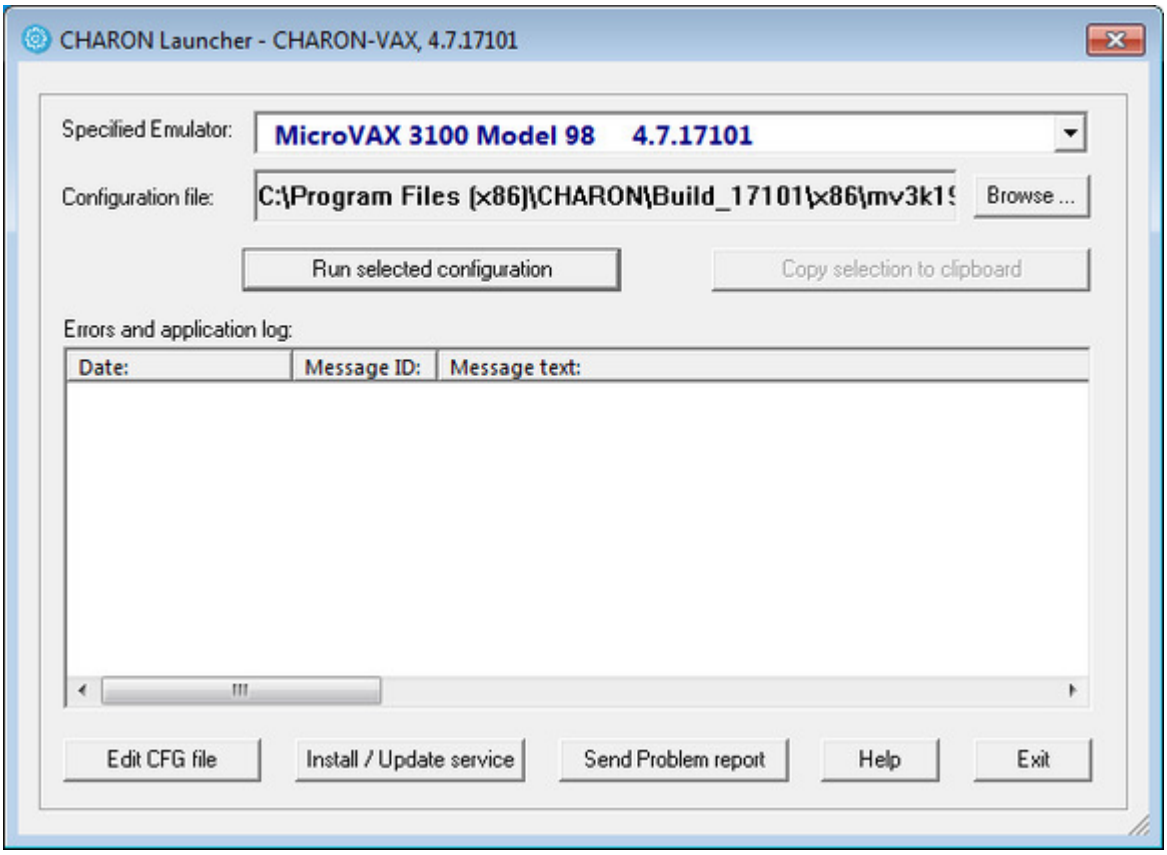


Select the desired configuration file:



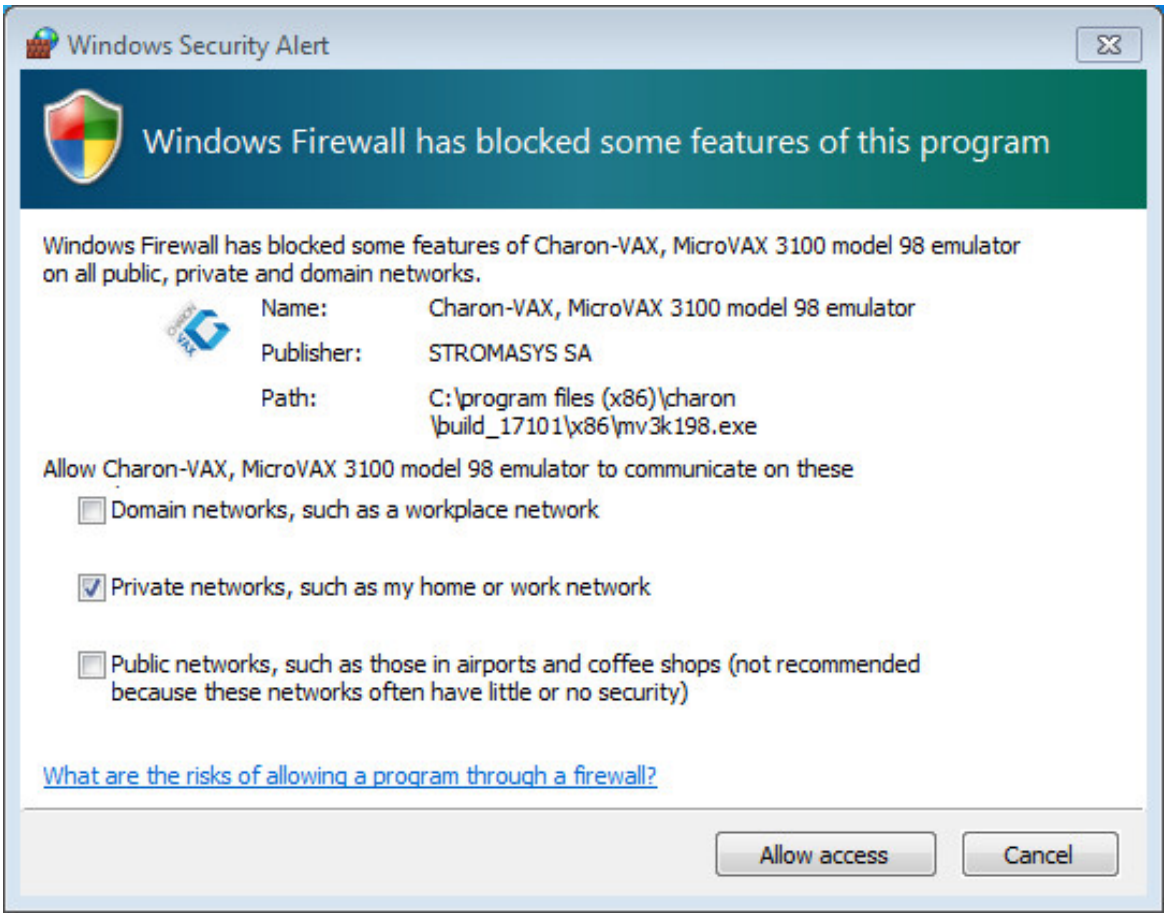
© Stromasys, 2017

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



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

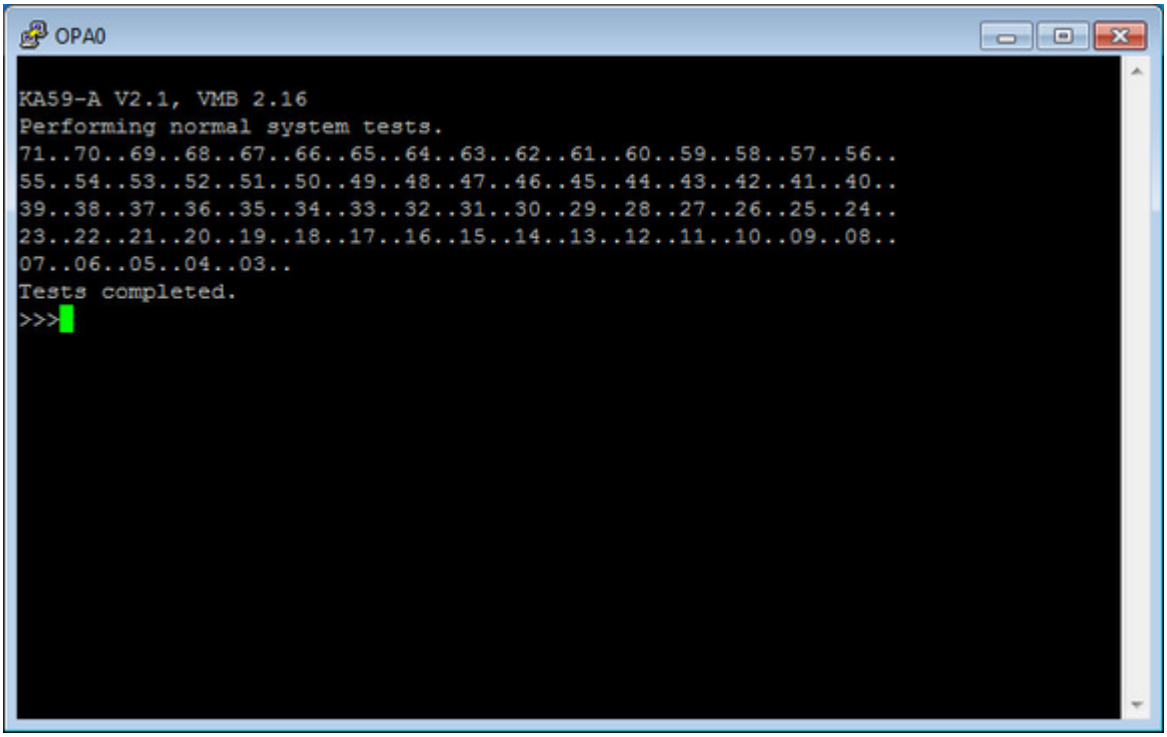
Depending on system settings, Windows may display a Firewall warning dialog asking to confirm CHARON access to different networks.



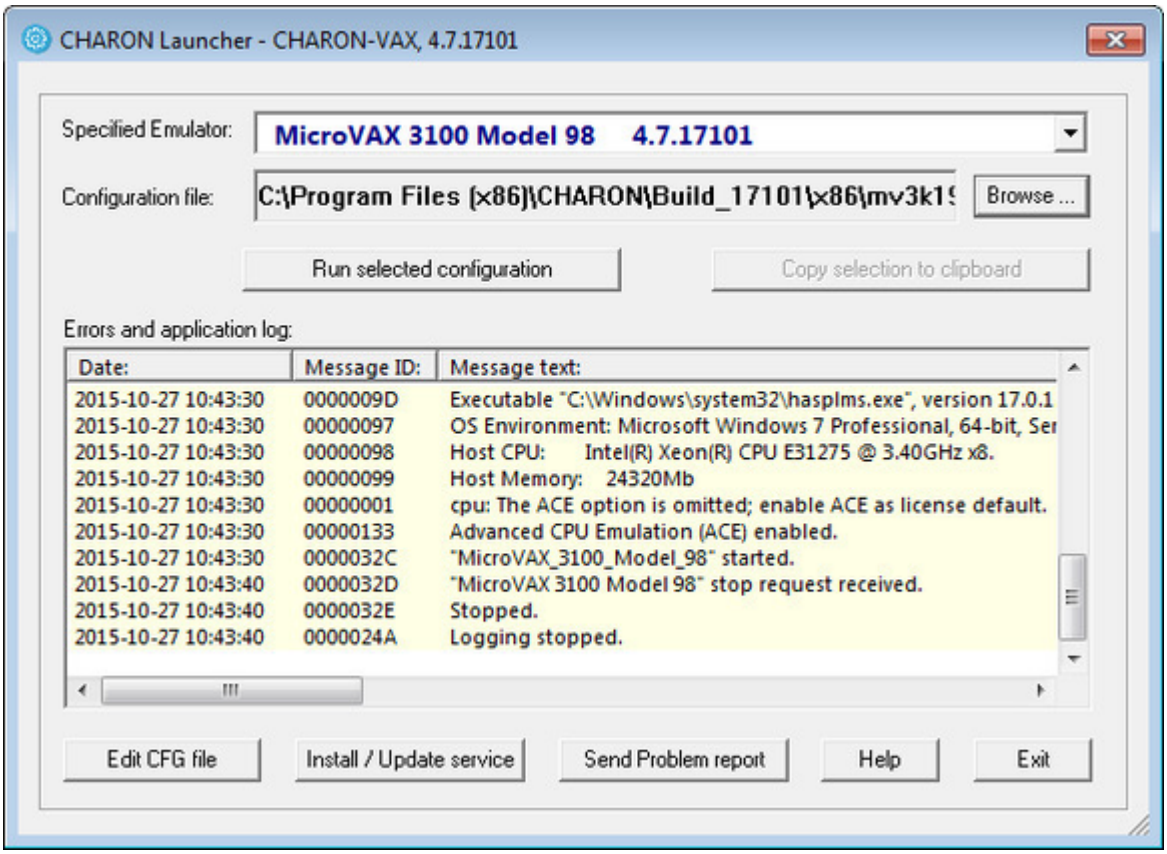
Check the desired options and press "Allow access" button.

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

Right after that the VAX/PDP11 emulated console will appear (in the default configuration. Some configurations may not assume console startup):



The Launcher will display CHARON log file:



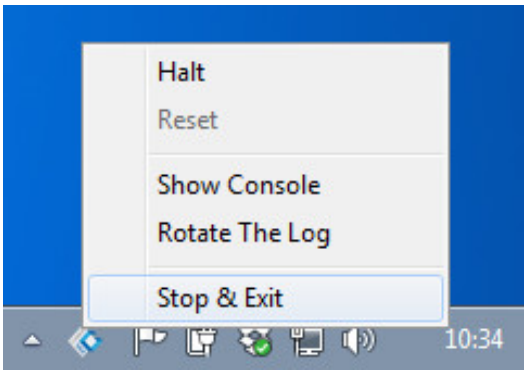
Log messages have the following colors:

- lightyellow means information
- cyan means warning
- red means error

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

To exit from CHARON-VAX / CHARON-PDP emulator use the following methods:

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



- Using this menu, it is also possible to halt the virtual VAX/PDP11, reset it or show its console if it has been closed

- **Configuration file updated to enable "F6" button in configuration file to trigger exit from CHARON:**

```
#--------------------------------------------------------------------------
#
# Uncomment to allow 'F6' to terminate the running emulator.
#
#--------------------------------------------------------------------------

set OPA0 stop_on = F6
```

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

Before stopping the CHARON-VAX / CHARON-PDP service, shutdown the operating system running in CHARON-VAX / CHARON-PDP.

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

Back to Table of Contents

Running as system service

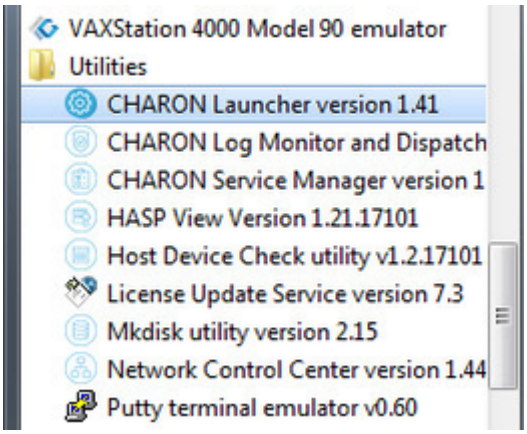
It is possible to run CHARON-VAX / CHARON-PDP as a Windows service.

Follow the description below to establish and run CHARON-VAX / CHARON-PDP as Windows service

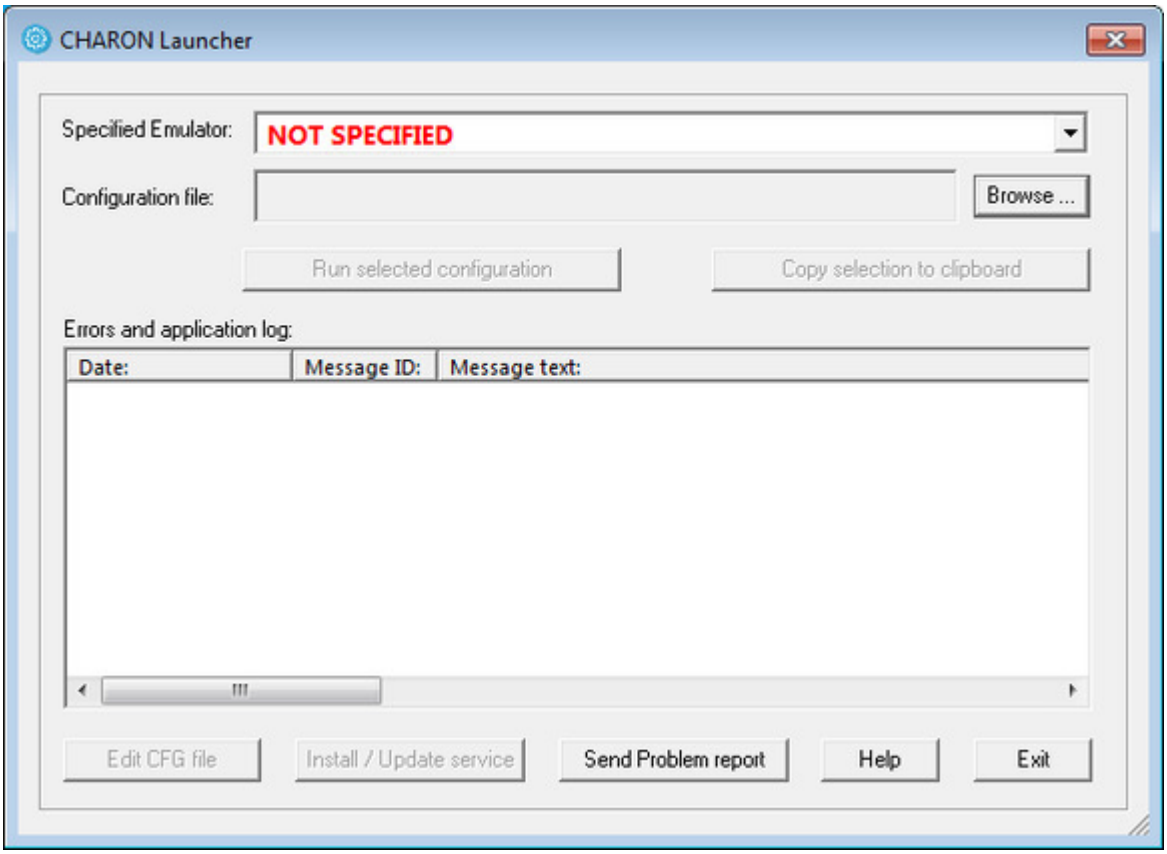
© Stromasys, 2017

Installation and start of CHARON-VAX / CHARON-PDP service

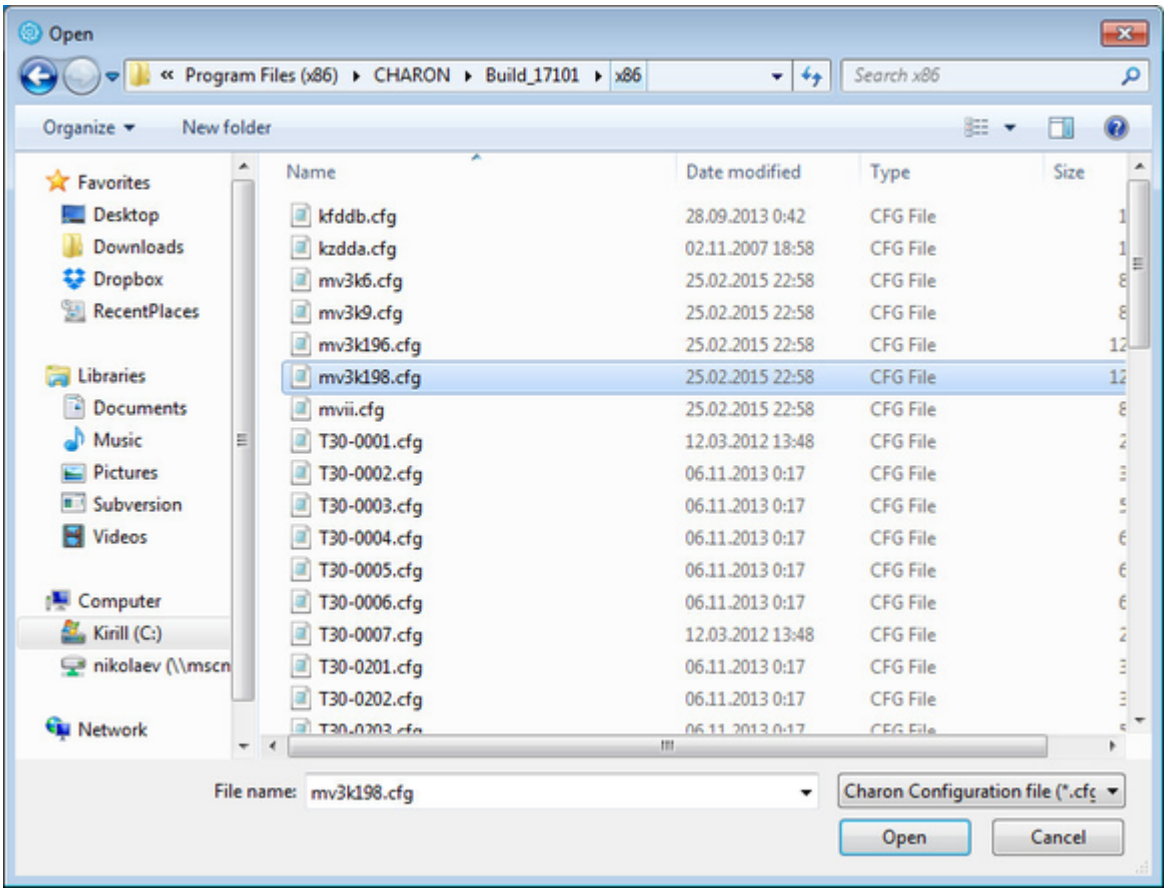
Open the "CHARON Launcher" utility from "Start" menu:



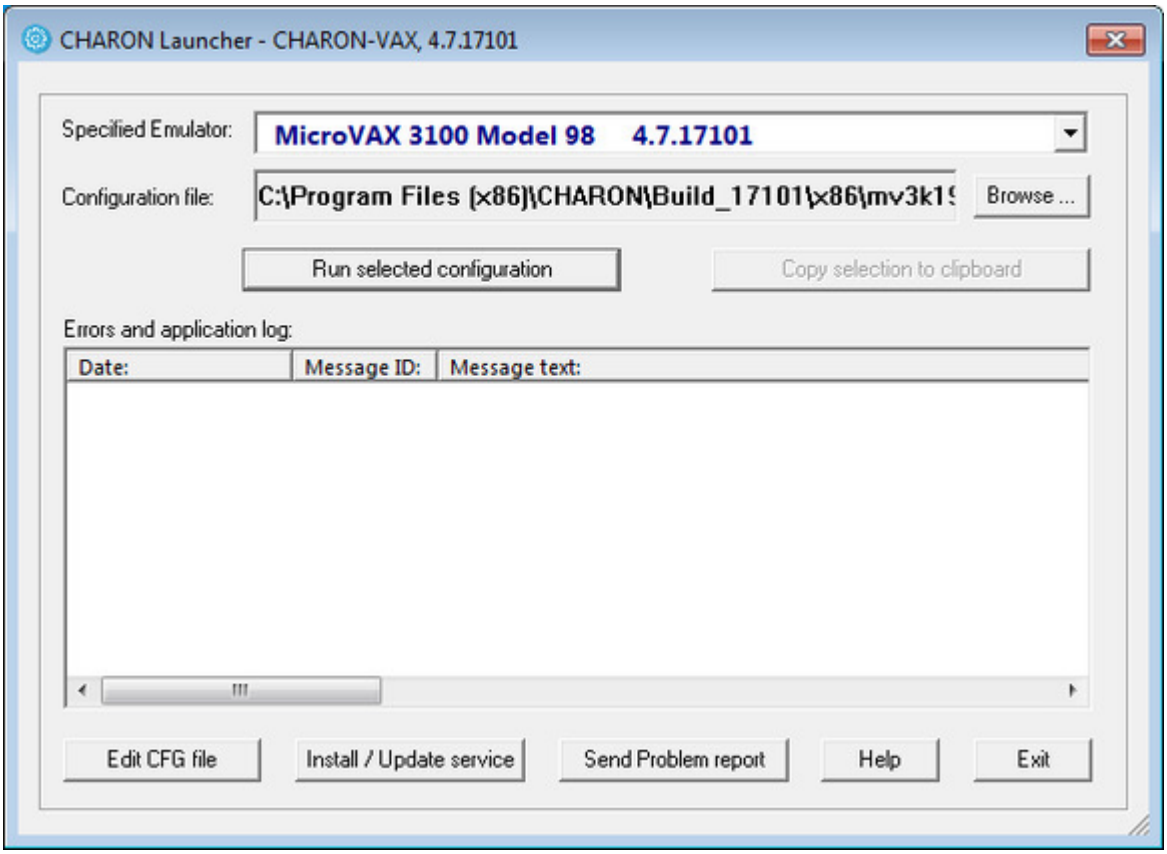
In the "CHARON Launcher" main interface press "Browse" button to select CHARON configuration to be used by CHARON running as Windows service:



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



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



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

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

```
...

set session log="C:\My Services\my_mv3k6.log"
set toy container="C:\My Services\my_mv3k6.dat"

load RQDX3/RQDX3 DUA
set DUA container[0]="C:\My Services\mv3k6_daemon_boot_disk.vdisk"
...
```

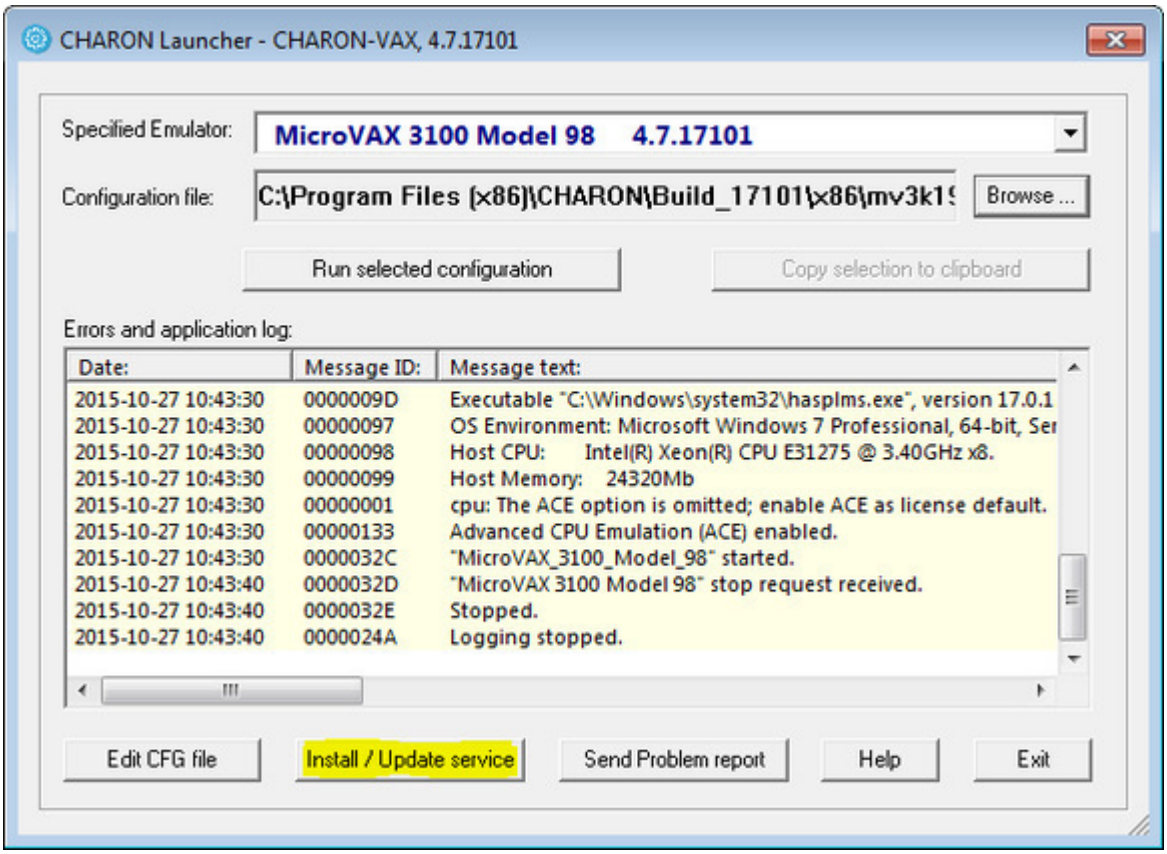
- Make sure the same physical devices are not used by other CHARON-VAX / CHARON-PDP services

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

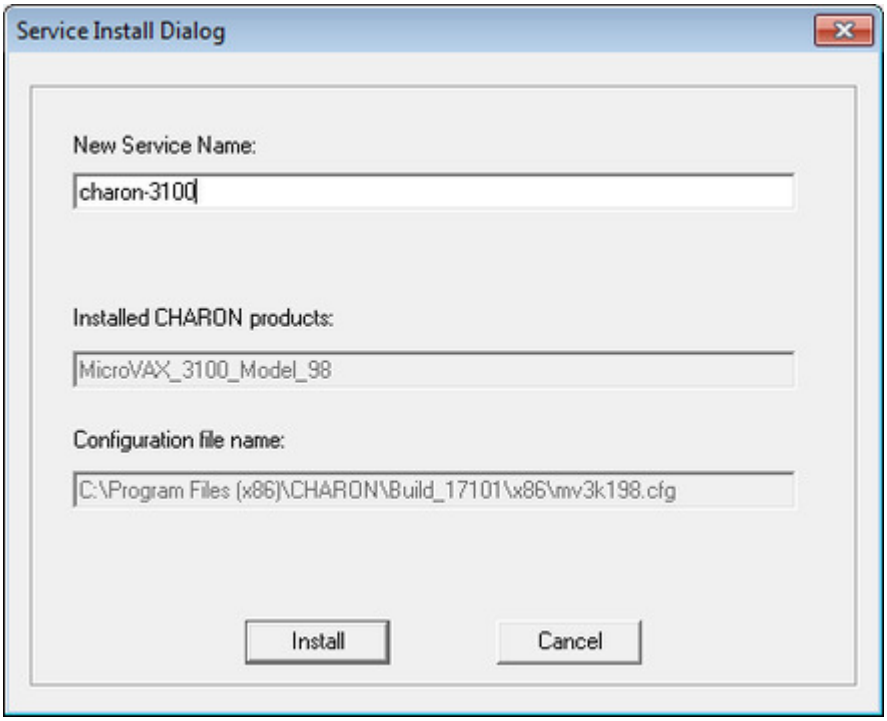
Run the configuration to make sure that it is ready to be installed as Windows service; stop CHARON (see above how to do that).

© Stromasys, 2017

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

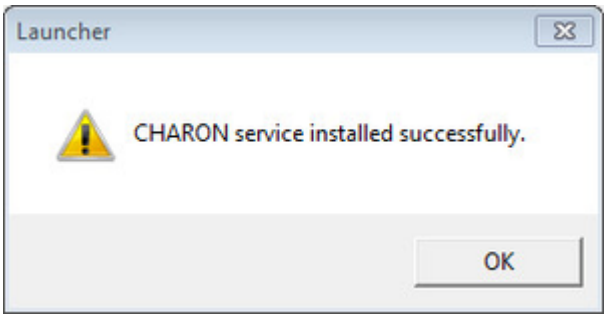


A special dialog for adding new CHARON service will appear:



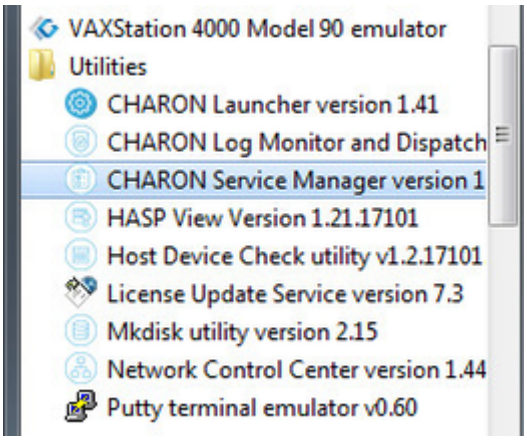
© Stromasys, 2017

Enter a new service name (in the example above it is "charon-3100") and press "Install" button. If the service has been installed correctly, a configration dialog box will appear:



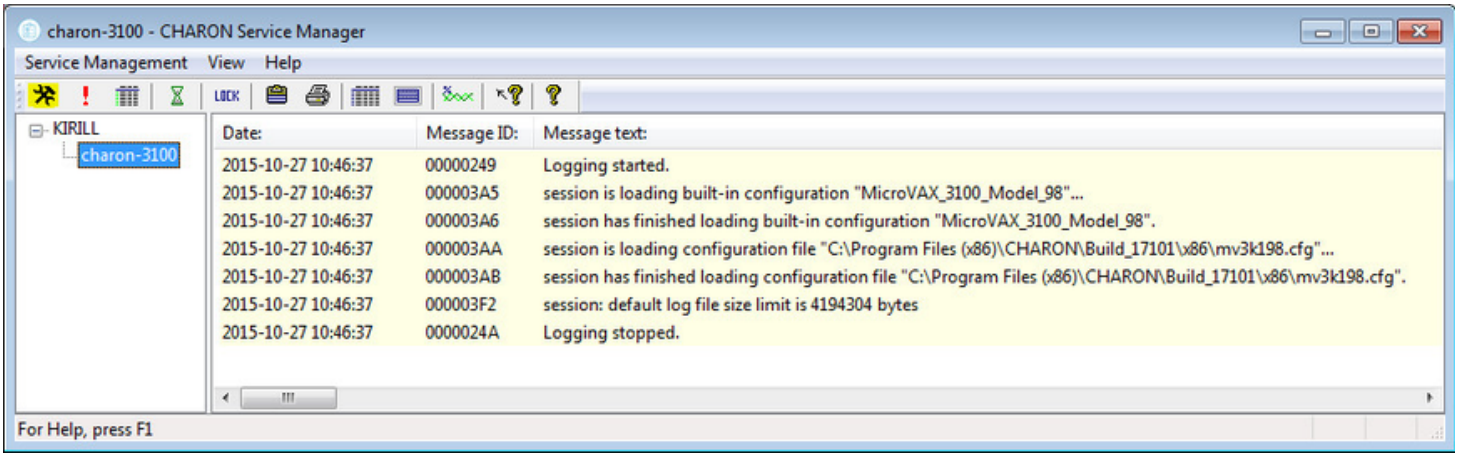
Press "Ok" button, then exit from "Launcher" by pressing "Exit" button.

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

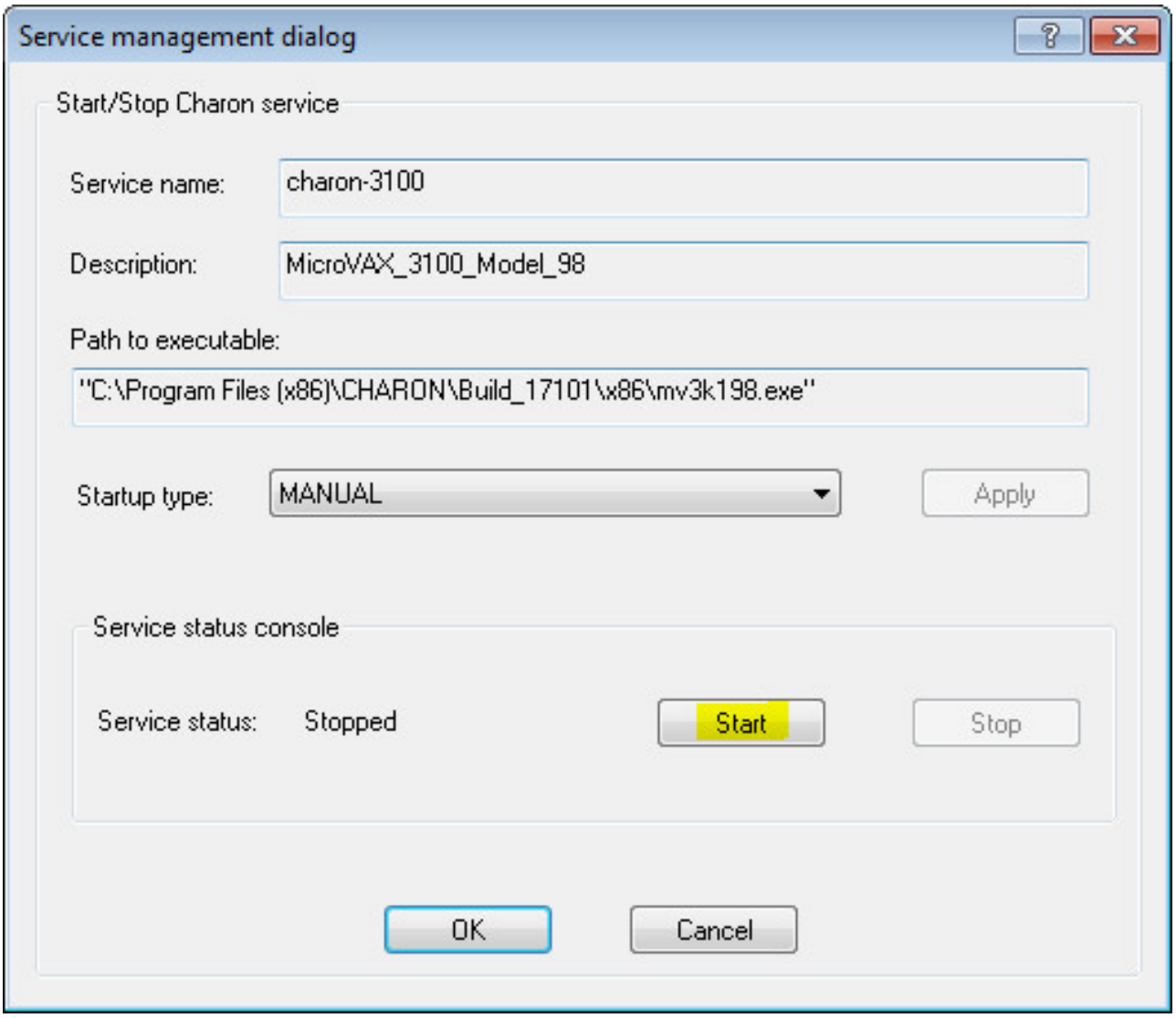


It is also possible to start the "CHARON Service Manager" utility from the system tray menu:

In the main window of the utility, select the target CHARON service ("charon-3100" in our example) and press the "Service" button (labeled with yellow on the screenshot below):



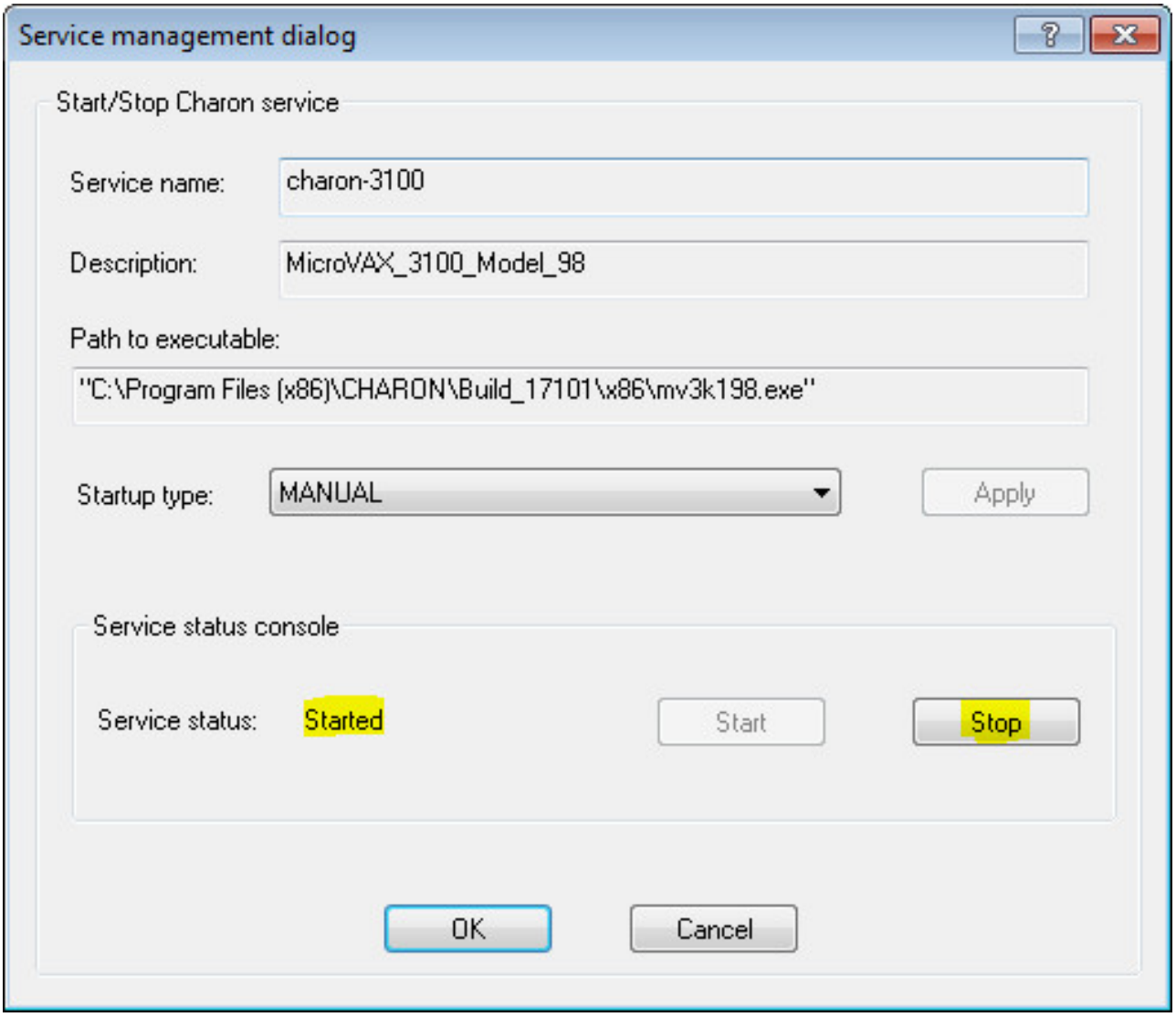
The "Service Management dialog" will appear:



The "Service Management dialog" shows the current status of the selected CHARON service and permits changes to its start up type and to start and stop it. The example screenshot above shows that the service "charon-3100" startup type is "MANUAL" (can be started / stopped only manually) and in the "Stopped" state - these are the default values.

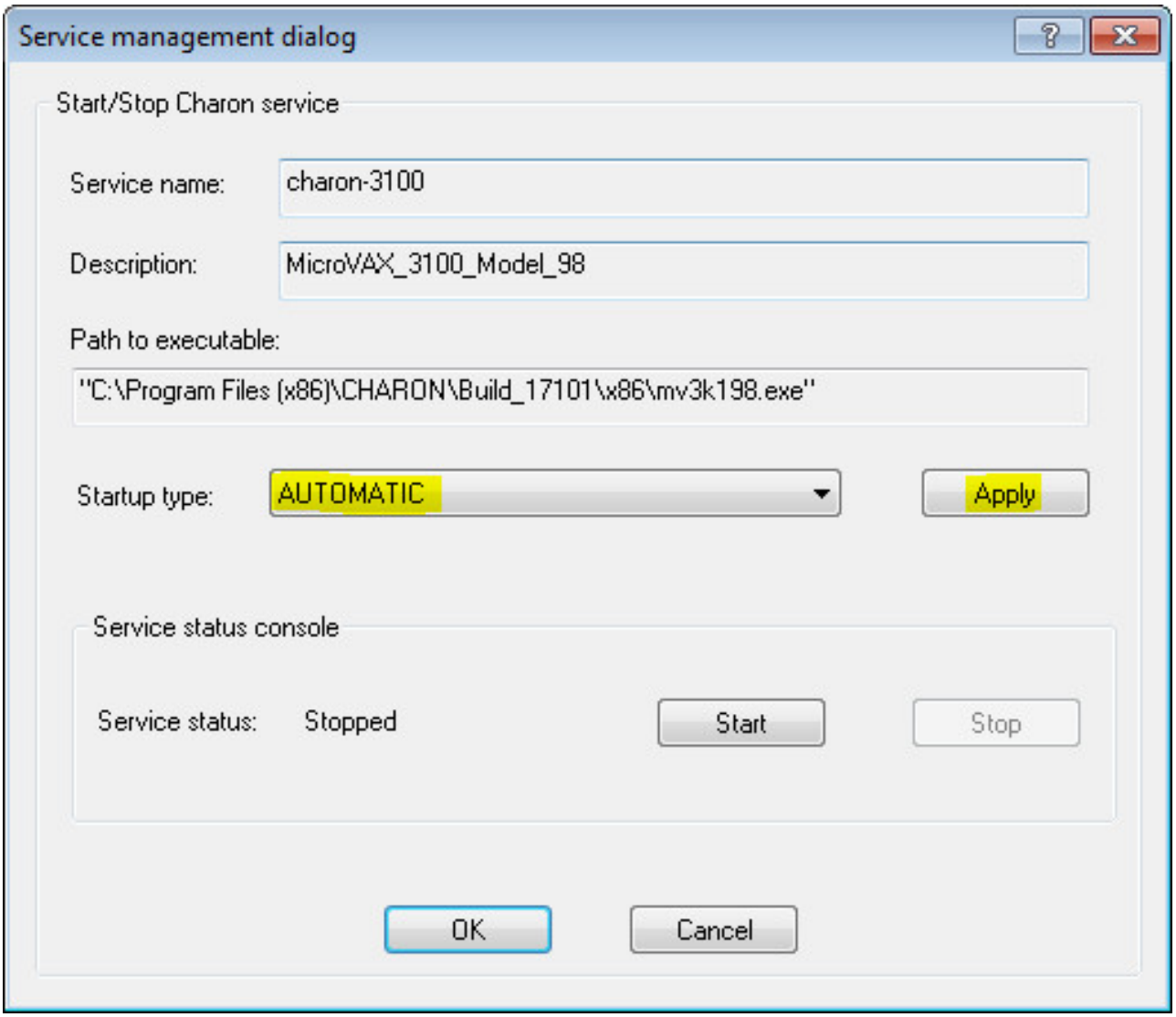
© Stromasys, 2017

Press the "Start" button to start the service:



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

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



Note that a certain delay may occur before the Sentinel Run-time process finds the network license on CHARON-VAX/PDP11 host system startup. If the CHARON-VAX/PDP11 service is starting automatically on the host system, it may report a "License not found" error and exit.

This problem can be avoided by specifying "license_key_lookup_retry" parameter in the configuration file:

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

where:

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

Example:

```
set session license_key_lookup_retry = 5
```

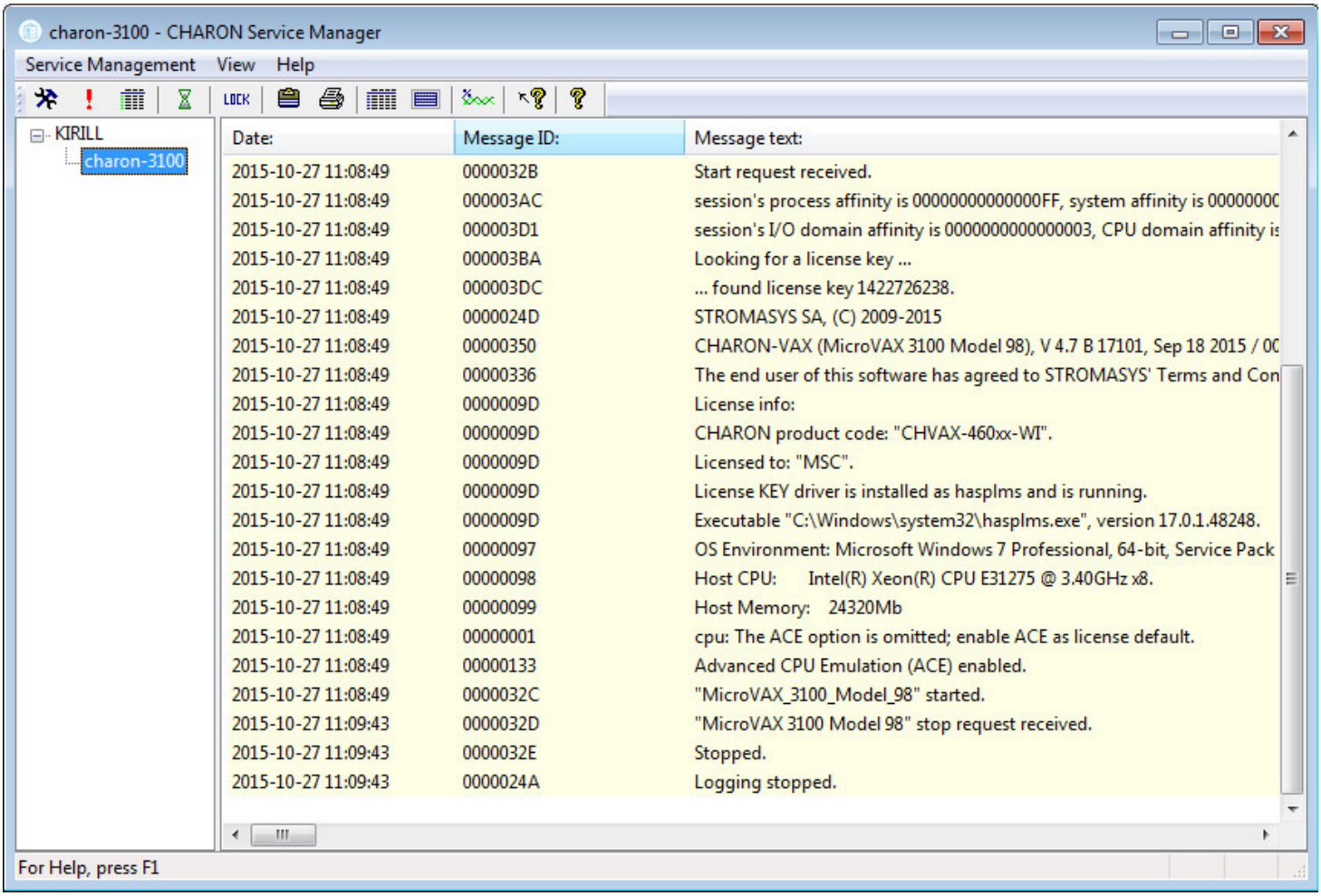
In this example, if the license key is not found during the initial scan, CHARON-VAX/PDP11 will scan 5 more times, waiting 60 seconds between them before it stops.

See the "General Settings" section for more details.

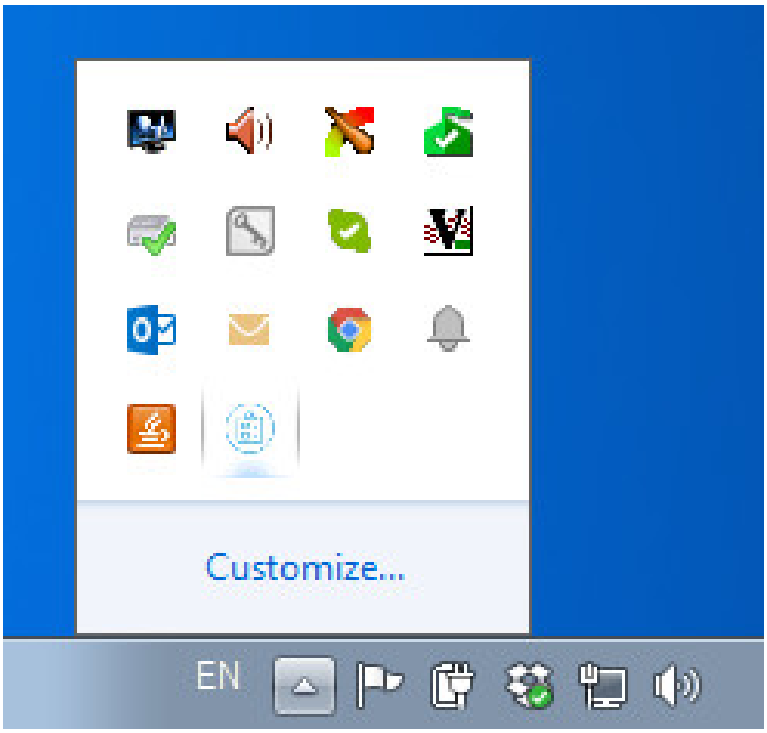
Back to Table of Contents

Management of CHARON-VAX / CHARON-PDP service

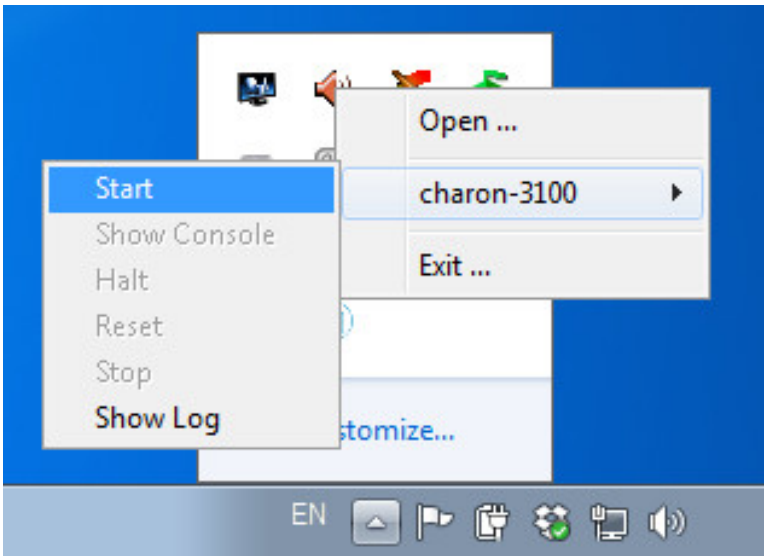
To see the execution log of a CHARON service, select the service ("charon-3100" in the example below) and the current log messages will be displayed in the right panel of the utility interface. You may need to scroll up to see startup/previous messages:



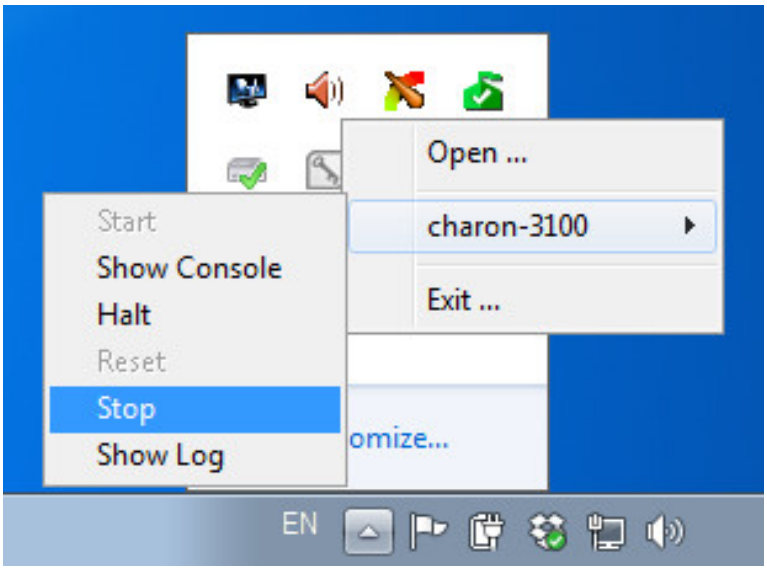
To control execution of CHARON services, open the "CHARON Service Manager" utility from the "Start" menu or use the system tray shortcut to manage the services w/o opening the main window of the utility:



Press the right button of the mouse to see a list of the installed CHARON services ("charon-3100" in our example):



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



Explanation:

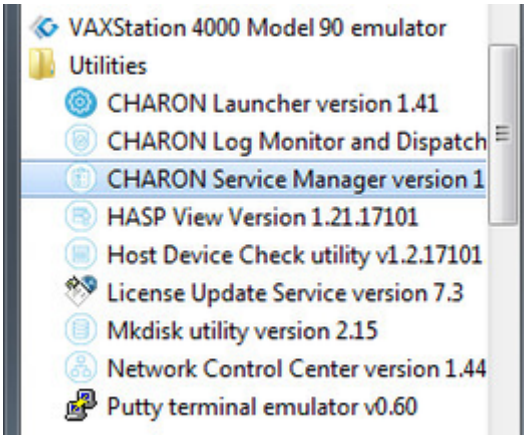
Menu Item	Function
Start	Start CHARON service
Show Console	Show CHARON service console if it has been closed or not active on the automatic service startup
Halt	Halt VAX/PDP11 emulator associated with the selected CHARON service
Reset	Reset the virtual VAX/PDP11 associated with the selected CHARON service
Stop	Stop CHARON service
Show Log	Show execution log of the selected CHARON service

Choose the desired action and click on it.

Back to Table of Contents

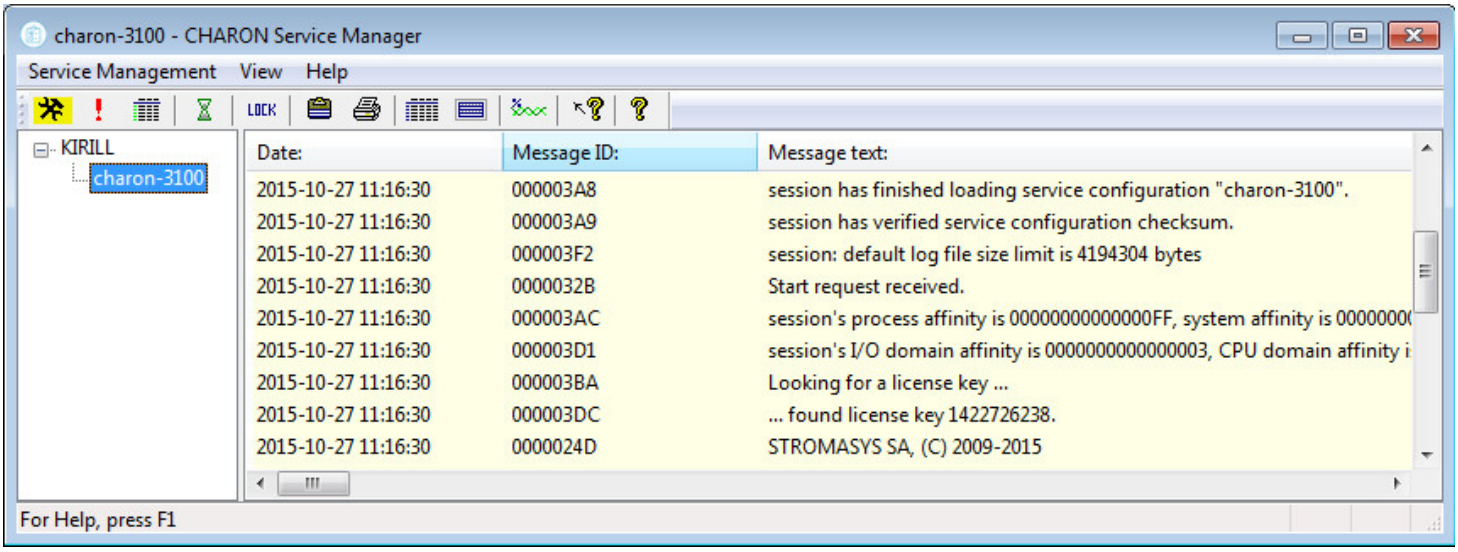
Stopping CHARON-VAX / CHARON-PDP service

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

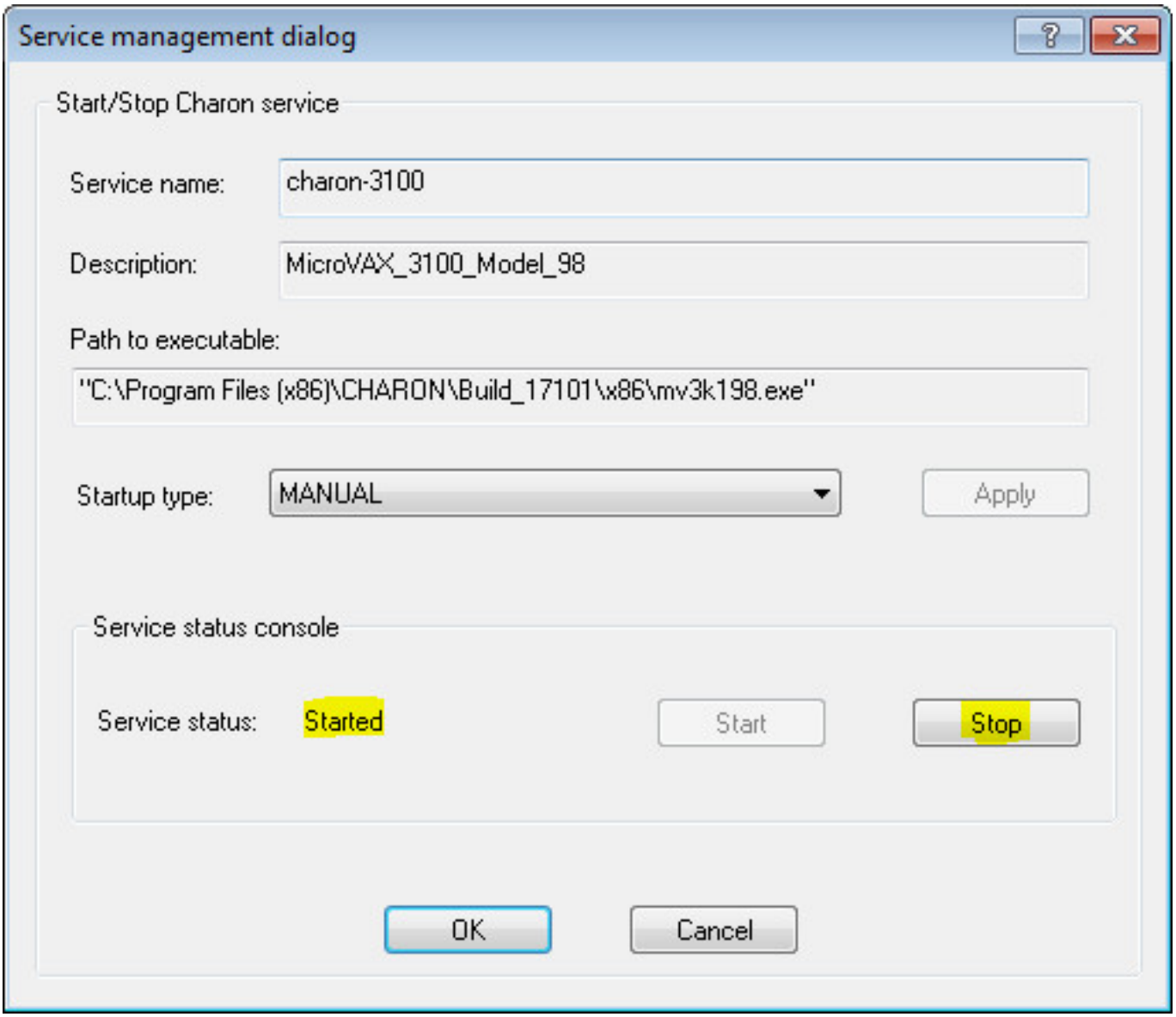


or from the system tray menu:

In the main window of the utility, select the target CHARON service ("charon-3100" in our example) and press the "Service" button (labeled with yellow on the screenshot below):

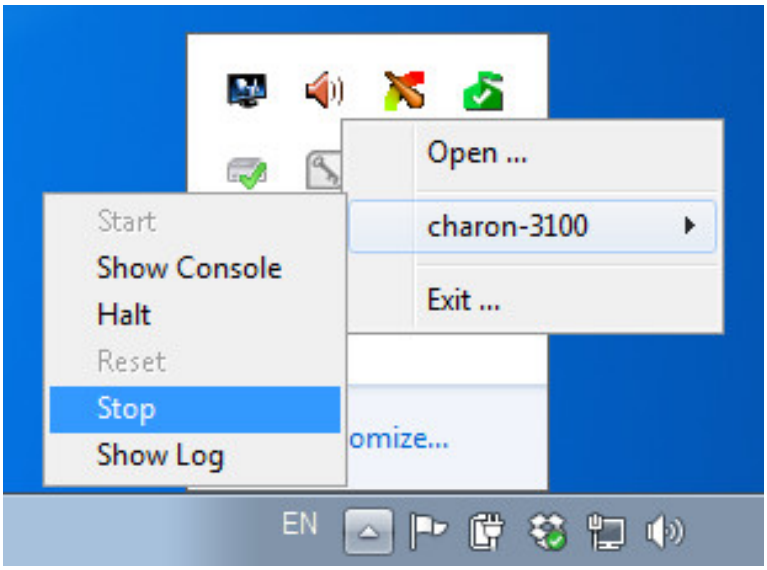


Press the "Stop" button in the resulting dialog box:



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

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



⚠ Please note: before stopping the CHARON-VAX/PDP11 service, shutdown the operating system running in CHARON-VAX/PDP11.

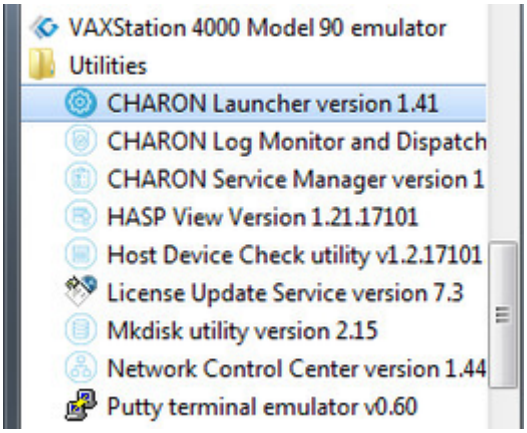
Back to Table of Contents

Modification of CHARON-VAX / CHARON-PDP service

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

Update the CHARON-VAX / CHARON-PDP configuration file for the CHARON service.

Start up the "CHARON Launcher" utility:



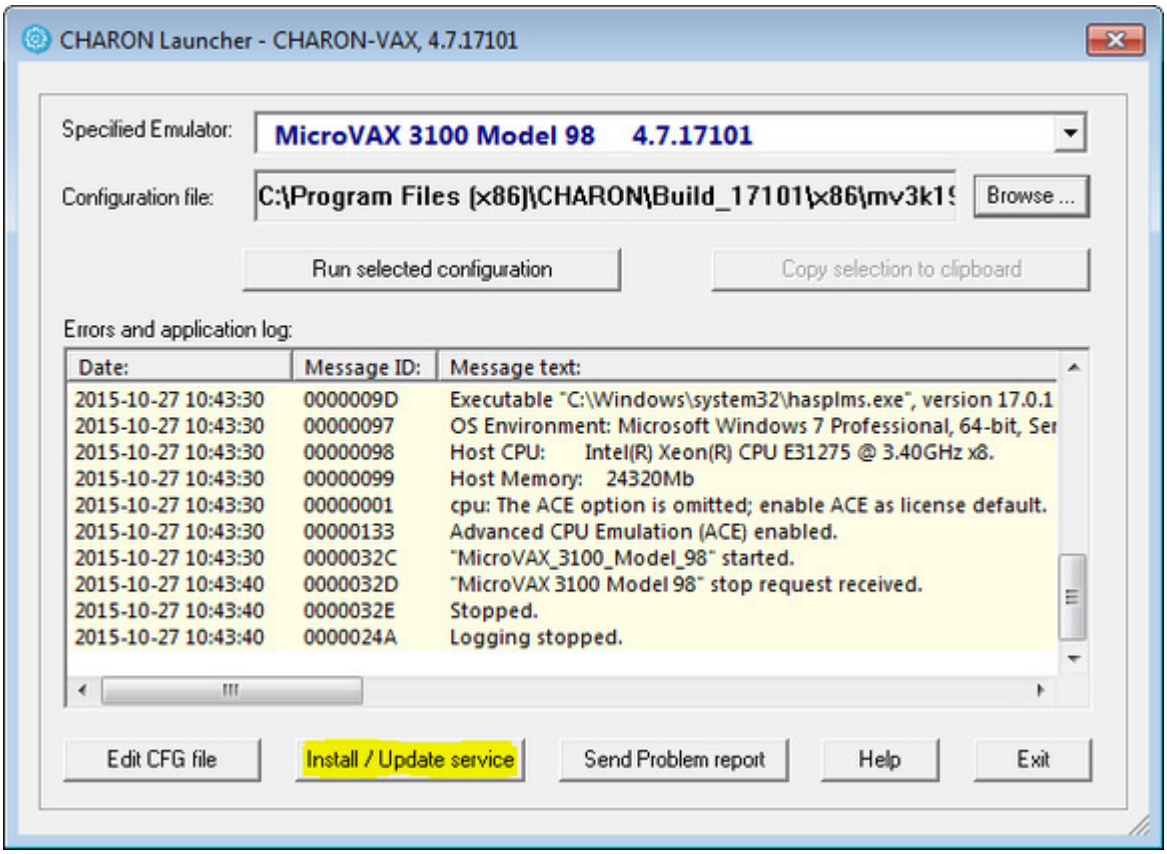
In the "CHARON Launcher" utility, select the modified configuration file and run it as described above.

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

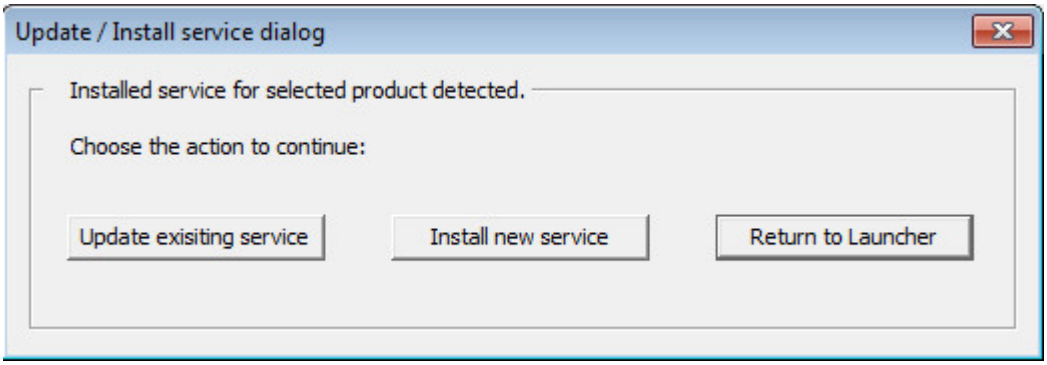
If errors appear, stop CHARON-VAX or CHARON-PDP, update its configuration file, start it again and make sure that the updated configuration works Ok this time, then stop CHARON-VAX / CHARON-PDP.

© Stromasys, 2017

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

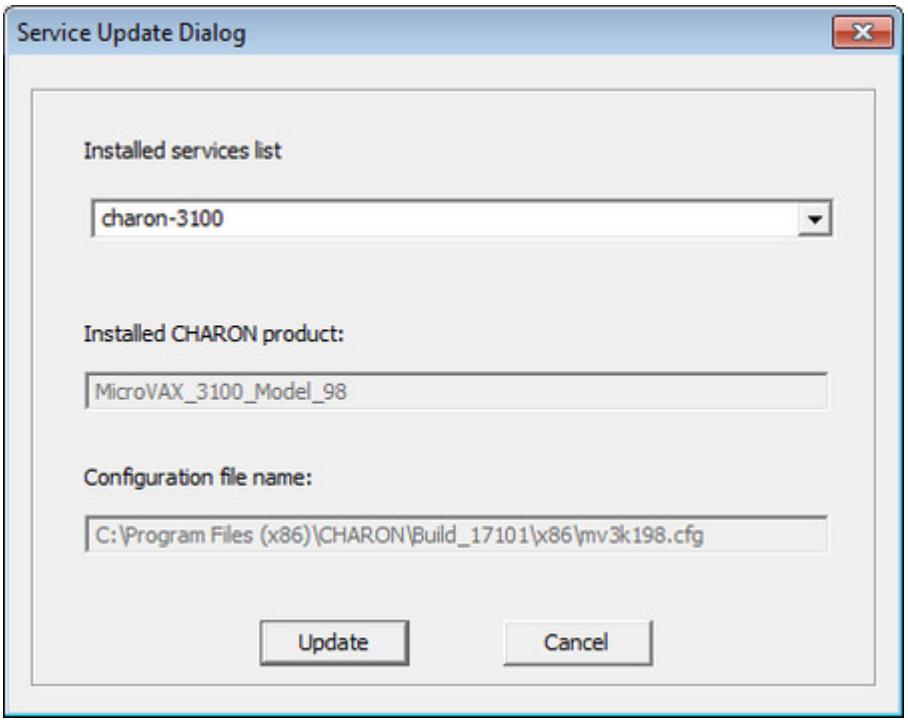


The install / update dialog will appear:



© Stromasys, 2017

Press "Update existing service" button:



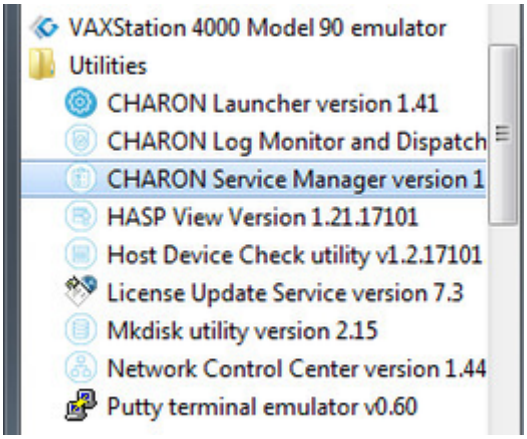
Select the service to update with the "Installed services list" drop-down box and press "Update" to apply.

⚠ Please note that before updating CHARON-VAX / CHARON-PDP service, one must stop it.

Back to Table of Contents

Removing CHARON-VAX / CHARON-PDP service

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

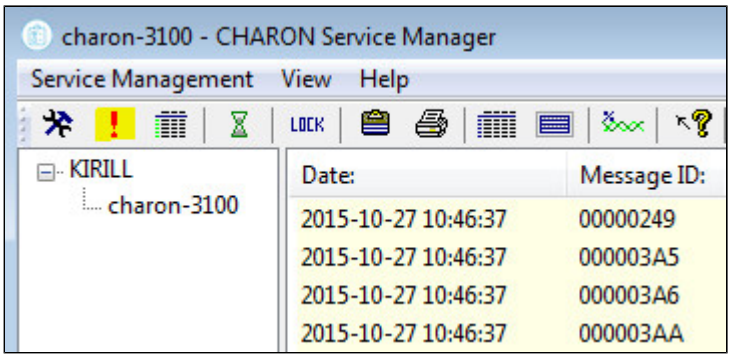


or from the system tray menu:

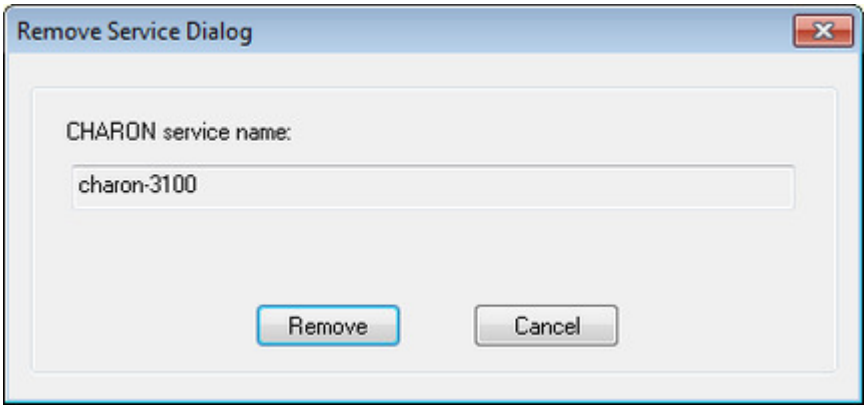
© Stromasys, 2017

Please note: before stopping the CHARON-VAX / CHARON-PDP service, shutdown the operating system running in CHARON-VAX / CHARON-PDP.

In the main window of the utility, select the CHARON service to remove ("charon-3100" in our example) and press the button with an exclamation sign on it:



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



Please refer to the next chapters for more details concerning CHARON-VAX / CHARON-PDP configuration details

Back to Table of Contents

© Stromasys, 2017

CHARON-VAX and CHARON-PDP for Windows configuration

Table of Contents

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

Creation of your own configuration file using a template

By default, all the CHARON templates are located in the "C:\Program Files (x86)\CHARON\Build_XXXXX\x86" (x64) or "C:\Program Files\CHARON\Build_XXXXX\x86" (x86) folder. Copy the appropriate template configuration file(s) to your home directory (or to any directory intended for CHARON-VAX / CHARON-PDP).

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

Once the file has been created you can open it in "Notepad" editor (or use the "CHARON Launcher" utility to open the configuration file) and proceed with modification to reflect the exact features of the system you are going to emulate.

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

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

Back to Table of Contents

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-VAX / CHARON-PDP configuration file from scratch it must be the very first uncommented line in the configuration file.

Back to Table of Contents

Configuration name

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

```
#set session configuration_name = MicroVAX_3600
```

© Stromasys, 2017

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

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

See example below with configuration name set to: `MicroVAX_3100_Model_98`

Back to Table of Contents

Log file parameters

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

Here is an example of a CHARON-VAX / CHARON-PDP log file:

```
20151027:112735:INFO :0:00000249:ethane.cxx(9611): Logging started.
20151027:112735:INFO :0:000003A5:ethane.cxx( 781): session is loading built-in configuration "MicroVAX_3600"...
20151027:112735:INFO :0:000003A6:ethane.cxx( 814): session has finished loading built-in configuration
"MicroVAX_3600".
20151027:112735:INFO :0:000003AA:ethane.cxx(1305): session is loading configuration file "mv3k6.cfg"...
20151027:112735:INFO :0:000003AB:ethane.cxx(1347): session has finished loading configuration file "mv3k6.cfg".
20151027:112735:INFO :0:000003F2:sesmgr.cxx(1394): session: default log file size limit is 4194304 bytes
20151027:112735:INFO :0:0000032B:ethane.cxx(2520): Start request received.
20151027:112735:INFO :0:000003AC:ethane.cxx( 695): session's process affinity is 00000000000000FF, system
affinity is 00000000000000FF.
20151027:112735:INFO :0:000003D1:ethane.cxx(1343): session's I/O domain affinity is 0000000000000003, CPU
domain affinity is 00000000000000FC
20151027:112735:INFO :0:000003BA:ll_sentine( 725): Looking for a license key ...
20151027:112735:INFO :0:000003DC:ll_sentine( 820): ... found license key 1422726238.
20151027:112735:INFO :0:0000024D:ethane.cxx(2651): STROMASYS SA, (C) 2009-2015
20151027:112735:INFO :0:00000350:ethane.cxx(2697): CHARON-VAX (MicroVAX 3600), V 4.7 B 17101, Sep 18 2015 /
000.msc.sanity.tests / 1422726238
20151027:112735:INFO :0:00000336:ethane.cxx(2724): 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
20151027:112735:INFO :0:0000009D:ethane.cxx(2801): License info:
CHARON product code: "CHVAX-460xx-WI".
Licensed to: "MSC".
License KEY driver is installed as hasplms and is running.
Executable "C:\Windows\system32\hasplms.exe", version 17.0.1.48248.
20151027:112735:INFO :0:00000097:ethane.cxx(2810): OS Environment: Microsoft Windows 7 Professional, 64-bit,
Service Pack 1 (Build 7601).
20151027:112735:INFO :0:00000098:ethane.cxx(2815): Host CPU: Intel(R) Xeon(R) CPU E31275 @ 3.40GHz x8.
20151027:112735:INFO :0:00000099:ethane.cxx(2820): Host Memory: 24320Mb
20151027:112735:INFO :0:00000001: tpool.cxx(1370): cpu: The ACE option is omitted; enable ACE as license
default.
20151027:112735:INFO :0:00000133: tpool.cxx(1602): Advanced CPU Emulation (ACE) enabled.
20151027:112735:INFO :0:0000032C:ethane.cxx(2556): "MicroVAX_3600" started.
20151027:112749:INFO :0:0000032D:ethane.cxx(2602): "MicroVAX 3600" stop request received.
20151027:112749:INFO :0:0000032E:ethane.cxx(2620): Stopped.
20151027:112749:INFO :0:0000024A:ethane.cxx( 365): Logging stopped.
```

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

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

Rotating log (default)

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

This mode is turned on if all the log parameters above are disabled (commented out) or the "session_log" parameter is pointing to a directory rather than

to a file. If a directory is specified, the log files will be created in that directory.

Names of the rotating log files are composed as follows:

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

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

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

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

Single log

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

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

Below is a specification of a CHARON-VAX / CHARON-PDP log file located in the "?:\My logs" directory which will be overwritten each time CHARON-VAX / CHARON-PDP starts:

```
set session log_method = overwrite
set session log = "C:\My logs\my_vax.log"
```

Back to Table of Contents

TOY, ROM and EEPROM containers

The next objects to be configured are TOY, ROM and EEPROM containers (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, CHARON-VAX / CHARON-PDP will create it. It is recommended to specify the path for each file so that time and console parameters will be kept whatever the current directory is when starting the guest.

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

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

The ROM container stores an intermediate state of the Flash ROM and some console parameters. So its container is also recommended to keep uncommented:

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

EEPROM stores the NVRAM content, so its container is also recommended to keep uncommented:

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

Back to Table of Contents

KW11 timer

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

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

Just uncomment the line corresponding to the frequency used in the RSX11/RT11 systems to be migrated.

Back to Table of Contents

ACE mode

The next setting indicates whether CHARON-VAX / CHARON-PDP will use Advanced CPU Emulation (ACE) mode:

© Stromasys, 2017

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

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.

If the "ace_mode" parameter is commented out, CHARON will set ACE mode according to your license settings. "Charon-*xx*/Plus" products have ACE mode enabled by default.

Back to Table of Contents

Emulated memory (RAM) size

The next parameter defines the amount of host memory the chosen CHARON-VAX / CHARON-PDP model 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.

© Stromasys, 2017

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.

Back to Table of Contents

© Stromasys, 2017

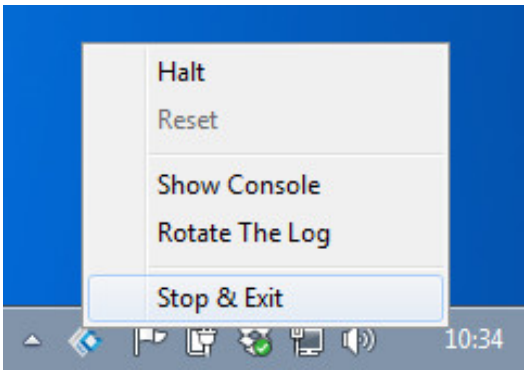
Console

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 CHARON-VAX / CHARON-PDP 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-VAX / CHARON-PDP host. Using this mapping it is possible to connect to CHARON-VAX / CHARON-PDP console and disconnect from it at any time.It is also possible to start up some chosen application to connect to the CHARON console IP port. **Parameter** / **Description** port — Specifies an IP port on the host to which the CHARON-VAX / CHARON-PDP console will be mapped application — A string specifying a name of an application to be started together with CHARON 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, CHARON-VAX / CHARON-PDP use the IP port "10003" for the VAX/PDP11 console and starts the "Putty" terminal emulator (included in each CHARON distributive) with the "OPA0" saved session directing it to connect to the IP port "10003".If the terminal emulator is closed, it is always possible to reopen it with the "Show Console" command from the CHARON system tray icon: 

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
```

In case of 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 there are a number of additional parameters for CHARON-VAX / CHARON-PDP serial lines configuration. Follow this link for details.

Exit on pressing F6 button

It is also recommended to set a hot key to stop the guest from the console in addition to the system tray icon:

```
set OPA0 stop_on = F6
```

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

Back to Table of Contents

Disk subsystem

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

CHARON-VAX / CHARON-PDP 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.

© Stromasys, 2017

MSCP disk controllers (RQDX3, KDB50, KDM70)

Below is a typical configuration sample for 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-VAX / CHARON-PDP 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-VAX / CHARON-PDP 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)"
    ```

© Stromasys, 2017

© Stromasys, 2017

- **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:"
    ```

Numbers in the square brackets represent 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 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.

Back to Table of Contents

© Stromasys, 2017

SCSI controller NCR53C94

The VAX 4000 and MicroVAX 3100 have an NCR53C94 SCSI controller onboard for 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 SCSI device not recognized by Windows
load physical_scsi_device pka_2 scsi_bus=pka scsi_id=2
set pka_2 container="\\.\Scsi<N>:<X>:<Y>:<Z>"

# 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:"
```

© Stromasys, 2017

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

<table>
<tr><th>Mapping Object</th><th>Description</th></tr>
<tr><td>virtual_scsi_disk</td><td>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-VAX / CHARON-PDP 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-VAX / CHARON-PDP 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.</td></tr>
<tr><td>physical_scsi_device</td><td>Mapping to a host SCSI device:
"\\.\Tape<N>" - mapping to a SCSI tape drive known to Windows
"\\.\ScsiN:X:Y:Z" - name of some SCSI device unknown to Windows for direct mapping, for example, a SCSI disk or tape reader. The values of N, X, Y and Z can be collected using special utility "Host Device Check" included in the CHARON distributive - or manually by investigation of the devices connected to CHARON host in the "Device Manager" applet.
<table><tr><th>Parameter</th><th>Description</th></tr><tr><td>N</td><td>A logical number assigned by host operating system (Microsoft Windows) to logical or host’s physical storage resource such as physical SCSI HBA</td></tr><tr><td>X</td><td>An internal SCSI bus number (usually 0) on host’s physical SCSI HBA</td></tr><tr><td>Y</td><td>A SCSI ID of physical SCSI target device attached to host’s physical SCSI HBA</td></tr><tr><td>Z</td><td>A logical unit number inside physical SCSI target device attached to host’s physical SCSI HBA</td></tr></table></td></tr>
<tr><td>virtual_scsi_cdrom</td><td>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: "<code>C:\My disks\vms_distributive.iso</code>"</td></tr>
<tr><td>virtual_scsi_tape</td><td>Mapping to a file representing tape (tape image). It may contain a path, for example: "<code>C:\My tapes\backup.vtape</code>"</td></tr>
<tr><td>floppy_scsi_device</td><td>Mapping to a host floppy drive:
"\\.\A:" or "\\.\B:" - name of host CD-ROM drive. This parameter can be omitted.</td></tr>
</table>

© Stromasys, 2017

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 "pkb_0" virtual SCSI device uses "PKB" controller by specifying a parameter "scsi_bus=pkb"

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

Follow this link for details of NCR53C94 SCSI controller controllers configuration.

Back to Table of Contents

Tape subsystem

Some MSCP and SCSI controllers support tape devices. CHARON-VAX / CHARON-PDP 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 TQK50 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 tape controller TQK50 with a name of 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

© Stromasys, 2017

the configuration sample above creates tape drive "MUA1". 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". Then configure the "TUK50" tape controller the same way as the TQK50.

Back to Table of Contents

Serial Lines

CHARON-VAX / CHARON-PDP 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 of the example above demonstrate loading serial line controllers of different types. The name of the controller (in this example) will be "TXA"

© Stromasys, 2017

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

<table>
<tr><th>Option</th><th>Description</th></tr>
<tr><td>physical_serial_line</td><td>Mapping to host "\\.\COM<N>" serial line</td></tr>
<tr><td>virtual_serial_line</td><td>Mapping to an IP port of CHARON-VAX / CHARON-PDP.
This mapping makes it possible to connect to and disconnect from the CHARON-VAX / CHARON-PDP 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><th>Parameter</th><th>Description</th></tr>
<tr><td>port</td><td>Specifies an IP port on the host to which the CHARON-VAX / CHARON-PDP serial line will be mapped</td></tr>
<tr><td>application</td><td>A string specifying a name of an application to be started together with CHARON to provide emulation of 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</td></tr>
</table>

Example:

<pre>load virtual_serial_line/chserial TXA0 port=10011 application="putty.exe -load TXA0"</pre>

In the example above, CHARON-VAX / CHARON-PDP uses the IP port "10011" for the VAX/PDP11 serial line and starts the "Putty" terminal emulator (included in each CHARON distributive) 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 by 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 host serial lines ("TXA<N>").

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

VAX 4000 and MicroVAX3100 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 exist for CHARON-VAX / CHARON-PDP serial lines configuration, follow this link for details.

Back to Table of Contents

© Stromasys, 2017

Networking

CHARON-VAX / CHARON-PDP 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). Note that 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 CHARON-VAX / CHARON-PDP 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 CHARON-VAX / CHARON-PDP 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-VAX / CHARON-PDP network controllers configuration.

Back to Table of Contents

Auto boot

CHARON-VAX / CHARON-PDP can be configured to automatically boot an operating system at start up.

MicroVAX 3100, VAX/PDP11 6310 and VAX/PDP11 4000 boot automatically if correct boot flags are set with VAX/PDP11 console:

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

Please check that the TOY, EEPROM and ROM containers (see above) are enabled so console command changes are saved between reboots.

The ROM of certain VAXes (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, some parameters in the configuration file can be used instead:

```
set bdr boot=auto
```

© Stromasys, 2017

CHARON-VAX 6000 models have a similar configuration setting:

```
set xmi boot=auto
```

CHARON-PDP models have the following setting:

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

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

Back to Table of Contents

Host load balance for SMP systems

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. “Affinity” overrides the default and allows explicit specification of which host CPUs will be used by the instance.Affinity does not reserve the CPU for exclusive use.	`set session affinity = "0, 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” overrides the default by specifying the number of I/O host CPUs explicitly.	`set session n_of_io_cpus = 2`

Back to Table of Contents

© Stromasys, 2017

Migration to CHARON-VAX and CHARON-PDP for Windows

Table of Contents

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

Introduction

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

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

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

Back to Table of Contents

© Stromasys, 2017

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-VAX configuration file.

Turn on your source VAX system. At the ">>>" prompt, issue "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)
```

© Stromasys, 2017

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.

Back to Table of Contents

© Stromasys, 2017

Creation of CHARON-VAX configuration file

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

```
...

#
# 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.

DESQA is mapped to a network interface having the name "Charon". This interface will be used for CHARON-VAX (see the Installation section for more details) on this particular host.

Back to Table of Contents

© Stromasys, 2017

Making disk images

In our example, possible mappings of RQDX3 and TQK50 tapes include physical devices and disk and tape images. Tape images should not be manually created, whereas you have to provision disk images, as described below.

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

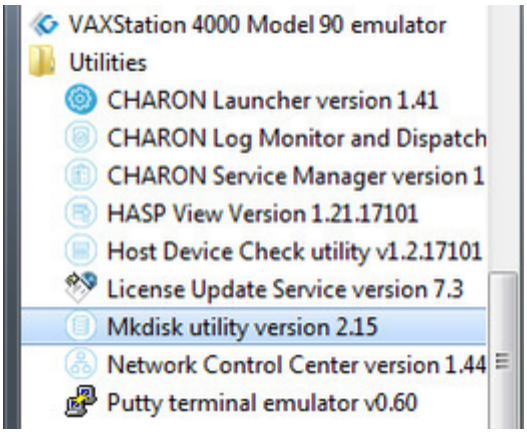
Create special directories for storing disk and tape images, as needed. Created directories are referenced in the sample configuration file above.

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

This operation can also be done using Windows Explorer.

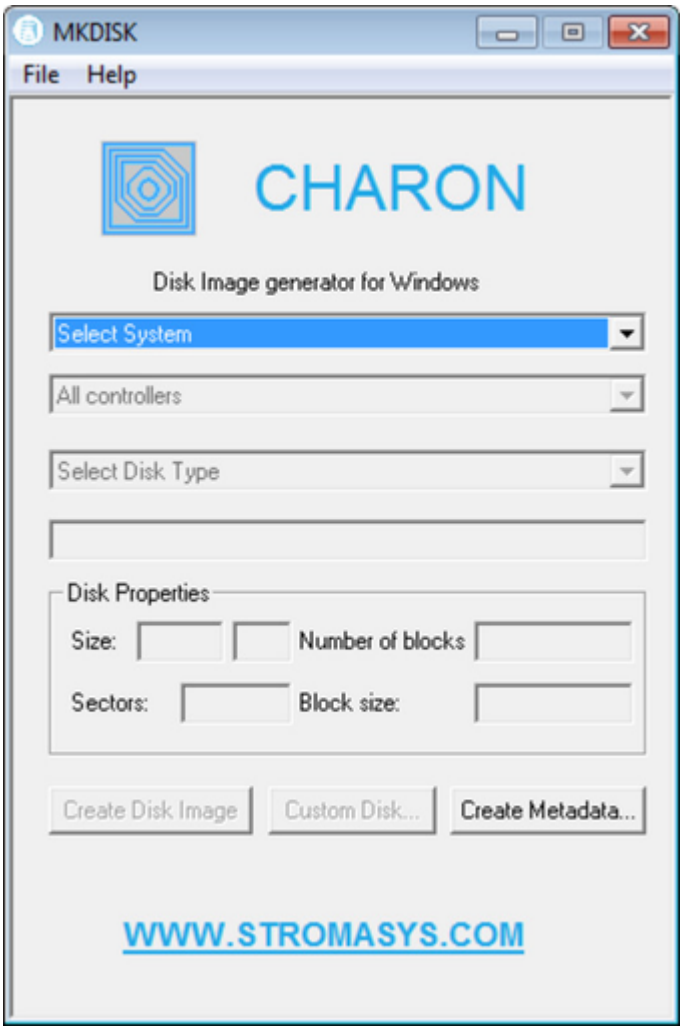
Next, create disk images using the "MkDisk" utility.

Start the "MkDisk" utility from the "Start" menu:

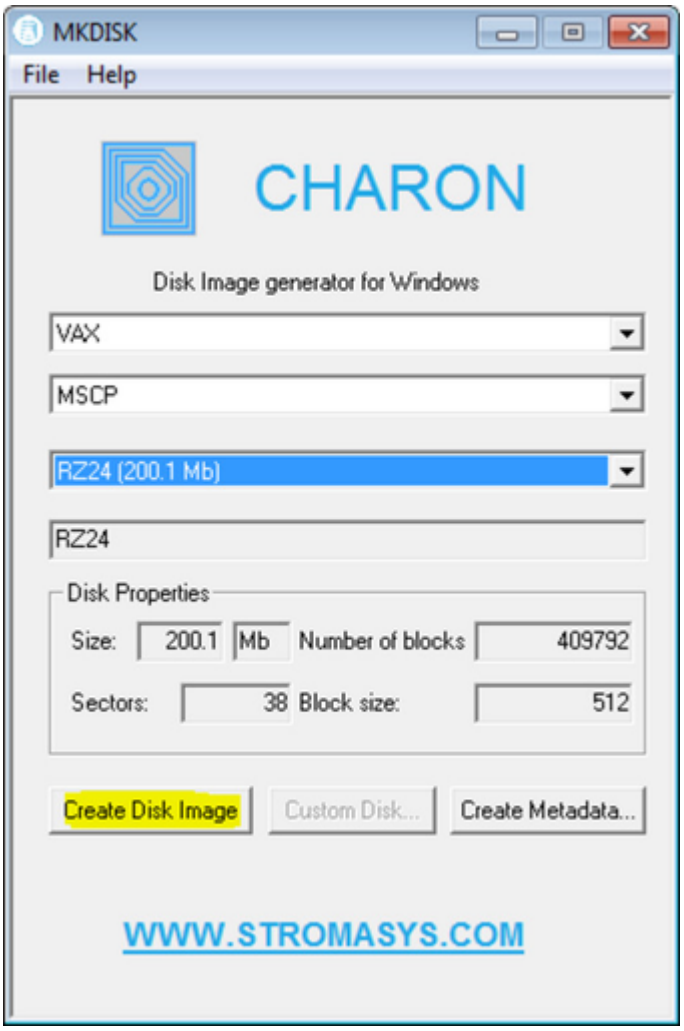


© Stromasys, 2017

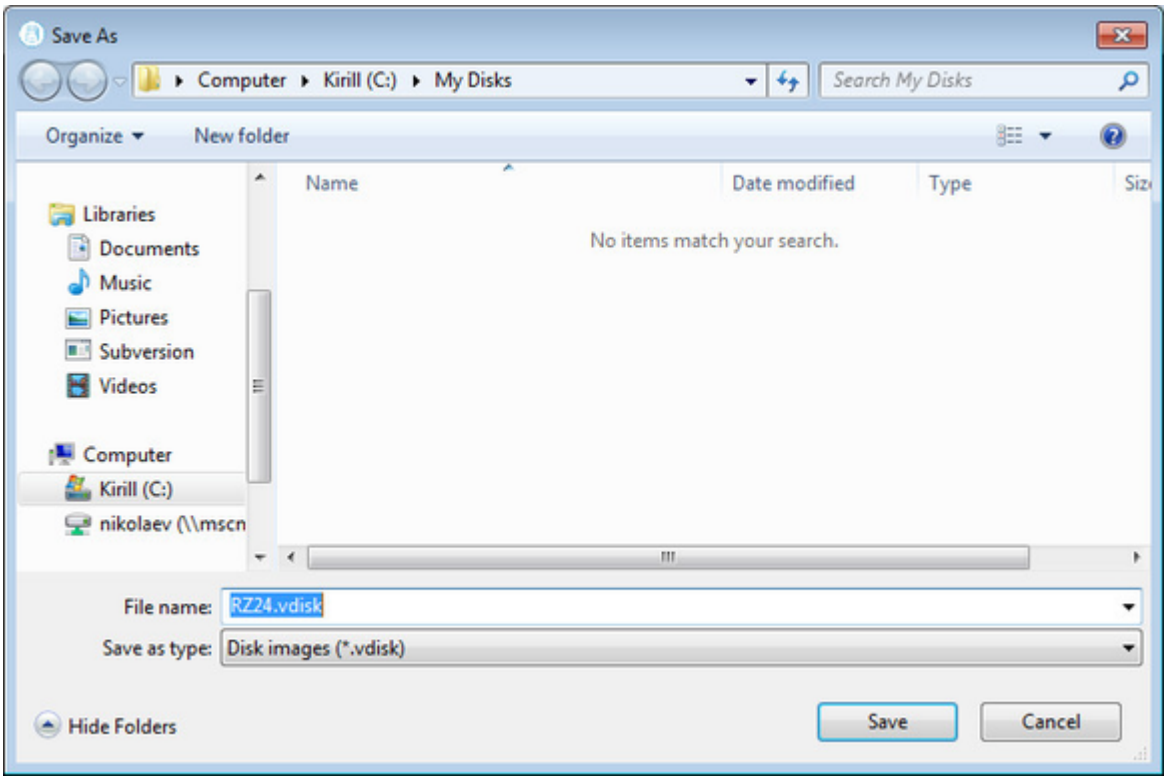
The main dialog of the utility will appear:



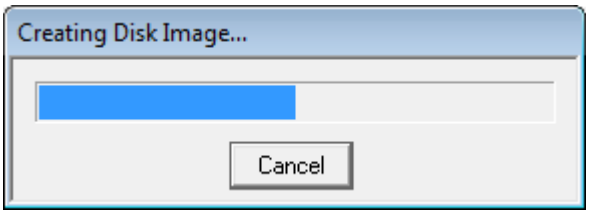
Select "VAX" in the "Select System" drop-down menu, "MSCP" in the "All Controllers" drop-down menu and "RZ24" (for example) in the "Select Disk Type" drop-down menu, then press "Create Disk Image" button:



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



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



Repeat this sequence for the disks "RZ23.vdisk", "RZ25.vdisk" and "RZ26.vdisk". Put them in the same "C:\My disks" folder.

Back to Table of Contents

Installation of VAX operating system

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

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 big 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 CHARON-VAX and boot from "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". DECnet address must belong to the same area as the source VAX system.

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

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

Back to Table of Contents

Making remote backups

Now we are ready to create disk backups from the source VAX system to CHARON-VAX.

Boot CHARON-VAX 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 application and close databases if any. The commands listed in SYS$MANAGER:SYSHUTDWN.COM may be helpful. The goal is to close as many files as possible. The system disk will have several files opened (pagefile, swapfile, etc.), but this is normal.

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

Issue (let's assume that the CHARON-VAX system is node 1.400 in this example) the following commands from the source VAX. Let's assume it's login 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
```

Once the backup procedure completes, the disk "DUA3" of CHARON-VAX will contain 2 savesets: "DUA0.BCK" and "DUA1.BCK"

Back to Table of Contents

© Stromasys, 2017

Restore backups to CHARON-VAX disks

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

```
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 CHARON-VAX and the original physical VAX on the network at the same time, you must change the network identity of one (usually the CHARON-VAX). The easiest way is to boot the CHARON-VAX on the restored system disk with the network Disabled and configure new addresses, as needed. The NIC can be disabled with a "connection:disabled" statement in the CHARON cfg file.

Then Enable the network and reboot.

Back to Table of Contents

Alternative ways of data transfer

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

- Connect a SCSI tape drive to CHARON-VAX / CHARON-PDP host via a PCI card
  - Map the tape drive in the CHARON-VAX / CHARON-PDP configuration file
    a. Restore source VAX system backups from tape to disk images via VMS/RSX11/RT11 running on CHARON-VAX / CHARON-PDP
    b. Boot from standalone backups and restore its content to CHARON-VAX / CHARON-PDP virtual disks
  - Dump source VAX/PDP11 system backups to tape images with "mtd" utility and:
    a. Boot from freshly installed VMS/RSX11/RT11 system and restore the tape images to CHARON-VAX / CHARON-PDP virtual disks
    b. Boot from standalone backups and restore its content to CHARON-VAX / CHARON-PDP virtual disks
- Create a network cluster between the source VAX/PDP11 system and CHARON-VAX / CHARON-PDP (it is possible to use the source system as boot server); then simple backup from one disk to another:

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

Back to Table of Contents

© Stromasys, 2017

CHARON-VAX for Windows DSSI cluster

Table of Contents

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

Introduction

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

Back to Table of Contents

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.

Back to Table of Contents

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-VAX system. 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-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. Disks on a multiport SCSI switch are not acceptable, as a SCSI switch does not provide true simultaneous access to multiple nodes.

© Stromasys, 2017

Steps to configure DSSI cluster:

1. Set unique ID for SHAC controller of each node

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 VAX nodes 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 node "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 nodes. 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.

© Stromasys, 2017

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 VAX nodes 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

Back to Table of Contents

© Stromasys, 2017

Example 1: Dual node DSSI cluster with 4 shared disks

To setup two emulated VAX 4000 Model 108 nodes, 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-VAX 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 impo rtant 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-VAX tries to locate the host adapter automatically. This automatic lookup works only if the CHARON-VAX 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-VAX configuration files abobe.

Back to Table of Contents

© Stromasys, 2017

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-VAX 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-VAX 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-VAX 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-VAX 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
...
```

Back to Table of Contents

© Stromasys, 2017

CHARON-VAX for Windows CI cluster

Table of Contents

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

Introduction

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

Back to Table of Contents

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-VAX/66x0 instances 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 nodes 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.

Back to Table of Contents

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-VAX system. 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-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. 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

Back to Table of Contents

© Stromasys, 2017

Example 1: Dual node CI cluster with 4 shared disks

In this example, setting up two emulated VAX 6610 nodes, 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-VAX 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-VAX tries to locate the host adapter automatically. This automatic lookup works only if the CHARON-VAX 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.

Back to Table of Contents

© Stromasys, 2017

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-VAX 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-VAX 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-VAX 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-VAX 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
...
```

Back to Table of Contents

© Stromasys, 2017

CHARON-VAX and CHARON-PDP for Windows licensing

Table of Contents

- General description
- Parameters defined by CHARON-VAX / CHARON-PDP license
- CHARON-VAX / CHARON-PDP licensing models
  - Regular Sentinel HASP keys
  - Network Sentinel HASP keys
  - Software licenses
- Multiple licenses configuration and backup license
- License installation
  - Installation from scratch
  - Replacement of currently installed Sentinel run-time
  - Installation and update of CHARON-VAX/PDP Software License or HL/HASP dongle License
- License management
  - Sentinel Admin Control Center
    - General Description
    - Disable remote keys access
  - License management utility
- Transferring and removing CHARON-VAX / CHARON-PDP software licenses
  - Software Licenses Transfer
  - Software License Removal
  - Cloned Software License Removal
- License Deinstallation
- Special "backup" license keys

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. Please be careful with your license key, in case of loss or damage, the CHARON-VAX / CHARON-PDP instances 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-VAX / CHARON-PDP configuration file to prevent CHARON-VAX / CHARON-PDP 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-VAX / CHARON-PDP and at a specified interval (defined by the license content) during the emulated system execution (default is 1 hour). If CHARON-VAX / CHARON-PDP detects the absence (or malfunction) of the license key or software license, CHARON-VAX / CHARON-PDP will try to use a backup license (if specified in the configuration file). If the license is not available or not specified, CHARON-VAX / CHARON-PDP 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-VAX / CHARON-PDP 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, CHARON-VAX / CHARON-PDP tries to find its replacement automatically and, if found, CHARON-VAX / CHARON-PDP proceeds using the replacement license.

> The present CHARON-VAX / CHARON-PDP implementation requires that the expired license be removed to allow the running CHARON-VAX / CHARON-PDP instance to switch to a valid one.
>
> The CHARON-VAX / CHARON-PDP software license is not distributed for Proof-of-Concept and evaluation installations. Only hardware dongles are used in this cases.
>
> It is important to connect the HASP license keys to the computer, even if CHARON-VAX / CHARON-PDP is not running, because the keys contain a built-in accumulator that needs to be charged. If the accumulator is completely discharged, the license key can be irreparably damaged.

Update of the CHARON-VAX / CHARON-PDP license can be performed on the fly without stopping CHARON-VAX / CHARON-PDP. All the parameters including the expiration date and time can be updated w/o any restrictions.

At the next license check, CHARON-VAX / CHARON-PDP will use the updated license normally.

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

© Stromasys, 2017

Back to Table of Contents

Parameters defined by CHARON-VAX / CHARON-PDP license

The following table represents all the parameters defined by the CHARON-VAX / CHARON-PDP 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-VAX / CHARON-PDP 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-VAX / CHARON-PDP 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-VAX / CHARON-PDP 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-VAX / CHARON-PDP license checking during CHARON-VAX / CHARON-PDP 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)

Back to Table of Contents

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-VAX / CHARON-PDP instances allowed to run on a particular host may be restricted by the license content (see above).

Network Sentinel HASP keys

The Network Sentinel HASP key (red dongle) can be shared between several hosts running CHARON-VAX / CHARON-PDP (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 to CHARON-VAX / CHARON-PDP instances running on other hosts. In this case the Sentinel driver must be installed on the host manually.

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

© Stromasys, 2017

for details.

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

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

Software licenses

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

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

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

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

Back to Table of Contents

Multiple licenses configuration and backup license

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

The "HASP View" utility displays all the available licenses. The utility provides the licenses number and ID / IP addresses of the hosts where the licenses are installed.

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

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

© Stromasys, 2017

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

Parameter	Type	Value

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

Back to Table of Contents

License installation

Installation from scratch

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

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

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

Installation of a CHARON-VAX / CHARON-PDP regular or network licenses 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 system fingerprint (*.c2v file), sending it to STROMASYS and applying update (*.v2c file) in case of software license. See the details below.

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

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

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

> Note that the following operations have to be performed on installation phase for network licenses:
>
> - *On server side (where network license will reside)*: open port 1947 for both TCP and UDP
> - *On clients side*: open UDP ports 30000-65535
> - *Both on server and client sides*: setup default gateway
>
> Please consult with your Windows User's Guide on details.
>
> If stricter firewall rules are required, it is possible to open the ports 30000-65535 and 1947 only for the "Sentinel HASP License Manager" (hasplms.exe) service (it will be installed by CHARON-VAX / CHARON-PDP).

Back to Table of Contents

Replacement of currently installed Sentinel run-time

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

- Login as Administrator
- Extract content of the unique ZIP file residing in “hasp_install” directory of the CHARON distribution to the same folder.
- Run “cmd.exe” from “Start” menu of host computer
- “cd” to the “hasp_install” directory
- Issue the following command:

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

Install other Sentinel run-time:

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

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

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

Back to Table of Contents

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

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

- Install CHARON-VAX / CHARON-PDP together with Sentinel run-time (Sentinel run-time is an essential part of CHARON-VAX / CHARON-PDP

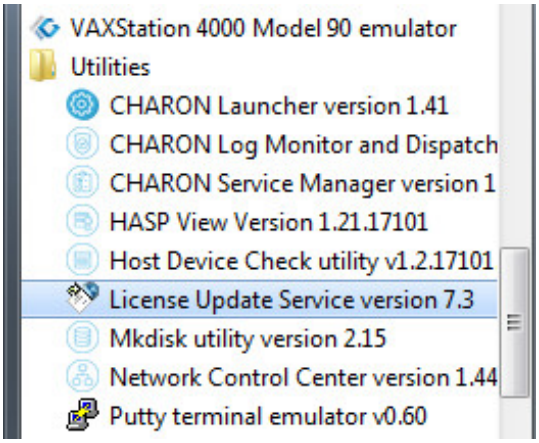
for Windows distribution)
- Reboot host system
- In case of Software License installation and if there are already installed network-wide SL's in local network disable access to network licenses in the following way:
    - Go to http://localhost:1947 to access the "Sentinel HASP Admin Control Center"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

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

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

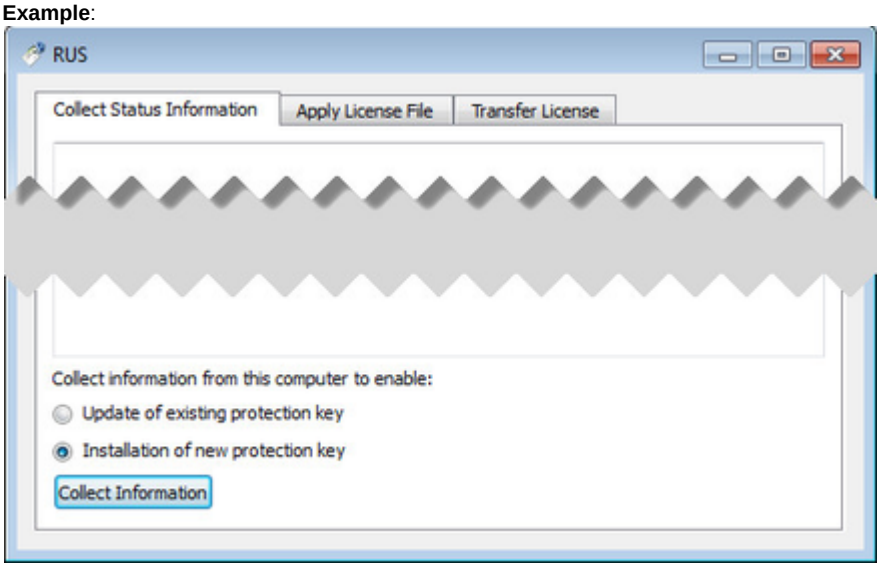
© Stromasys, 2017

Open "License Update Service" utility:



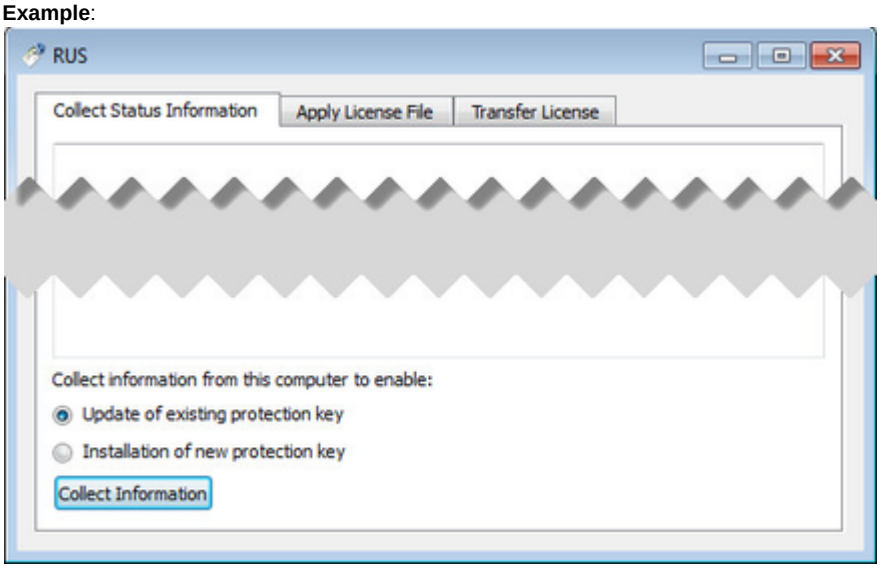
- In the utility dialog popup open the "Collect Status Information" tab, then select:
  - "Installation of new protection key" (in case of Software License if no Software License has been installed on the host):

    Example:

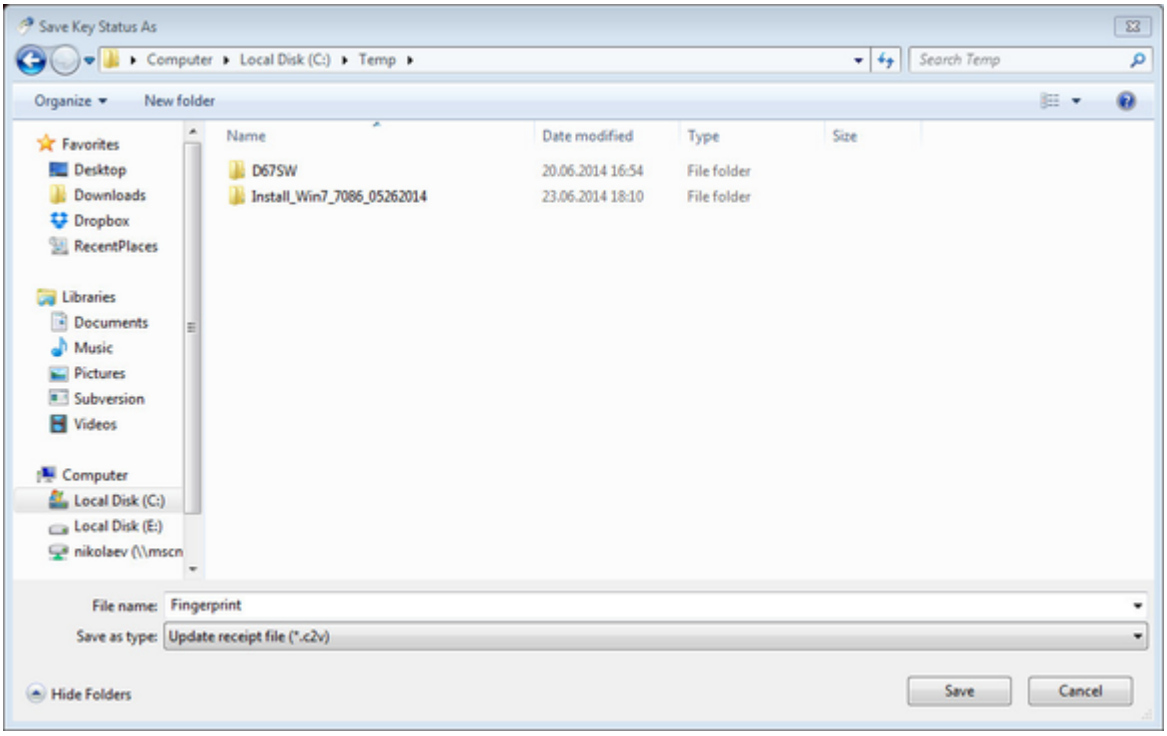
  - "Update of existing protection key" (in case of HL/HASP dongle or Software License that has been already installed on the host and needs updating)

Example:

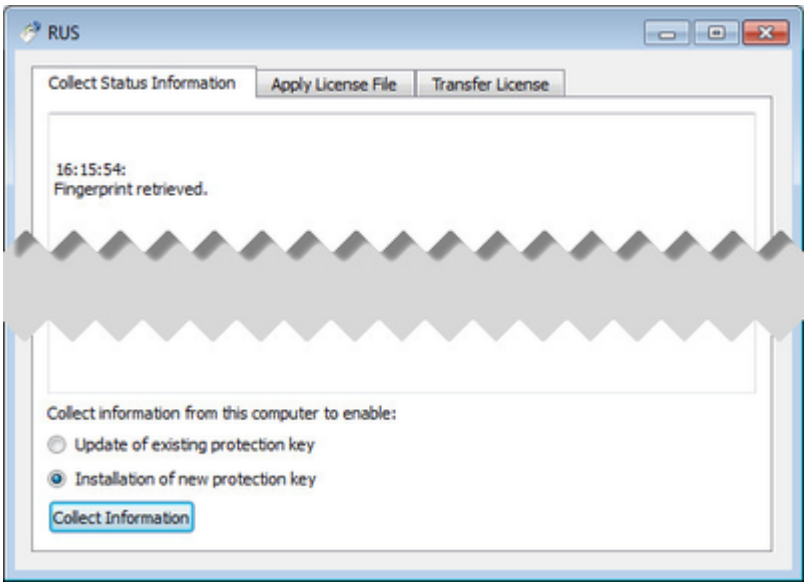


> In case of HASP dongles use only "Update of existing protection key" option!
>
> In case of Software License use "Installation of new protection key" option if host does not have Software License installed and "Update of existing protection key" option if already installed Software License has to be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

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

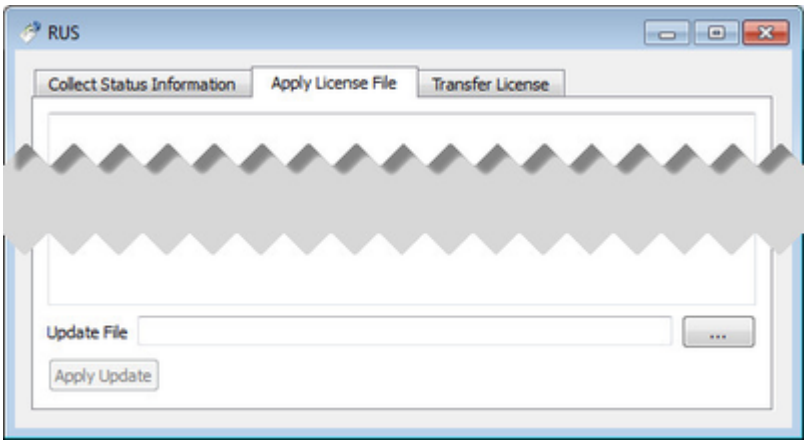


- A message should appear similar to this example, confirming the fingerprint has been collected successfully.

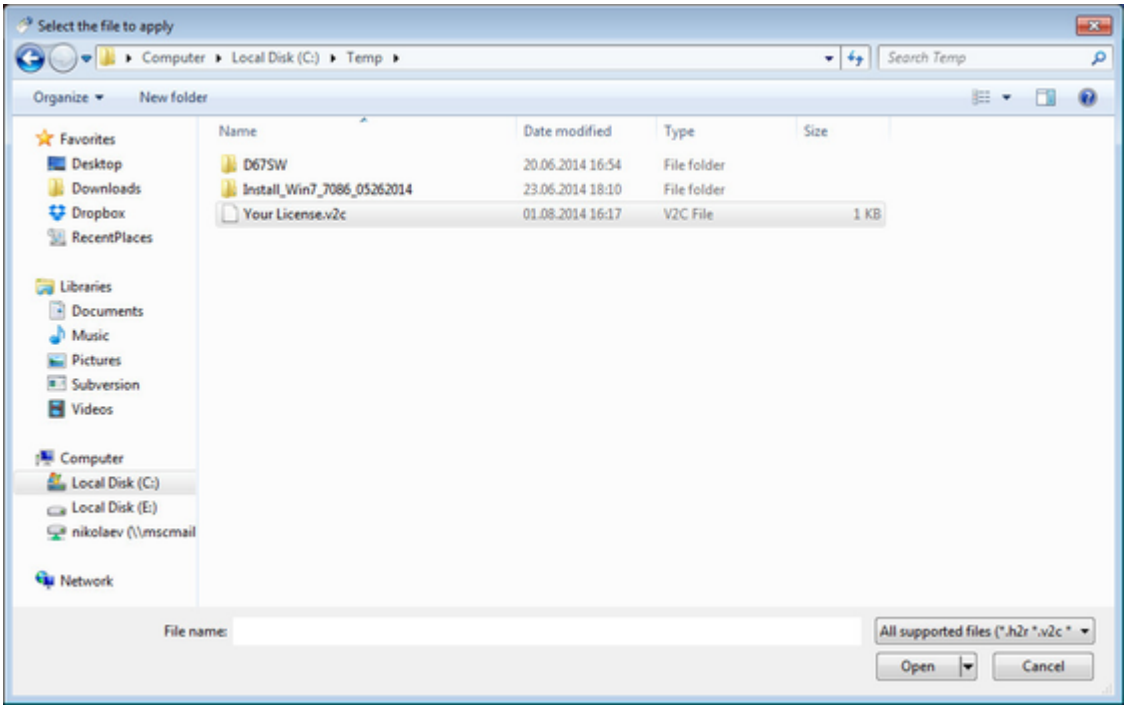


- Send the ".c2v" file ("*Fingerprint.c2v*" in the example above) to STROMASYS

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



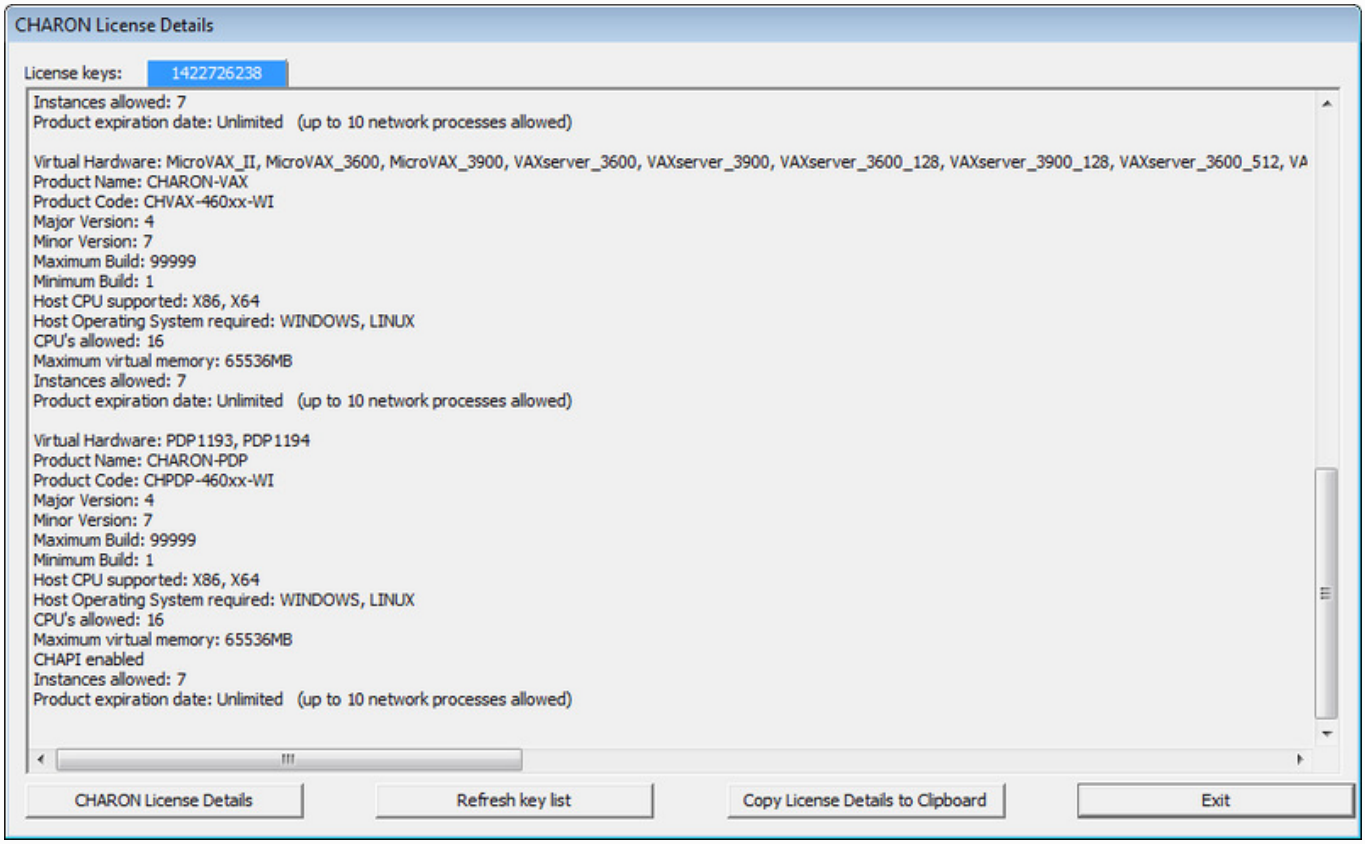
Press "..." button beside the "Update File" edit-box. In the popup select the license file received from STROMASYS:



Press the "Open" button and apply the license.

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

The content of the installed software license is not shown by the Sentinel HASP Admin Control Center. To see it, run the "HASP View" utility from the CHARON-VAX / CHARON-PDP "Utilities" menu (see above):



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

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

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

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

Back to Table of Contents

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.

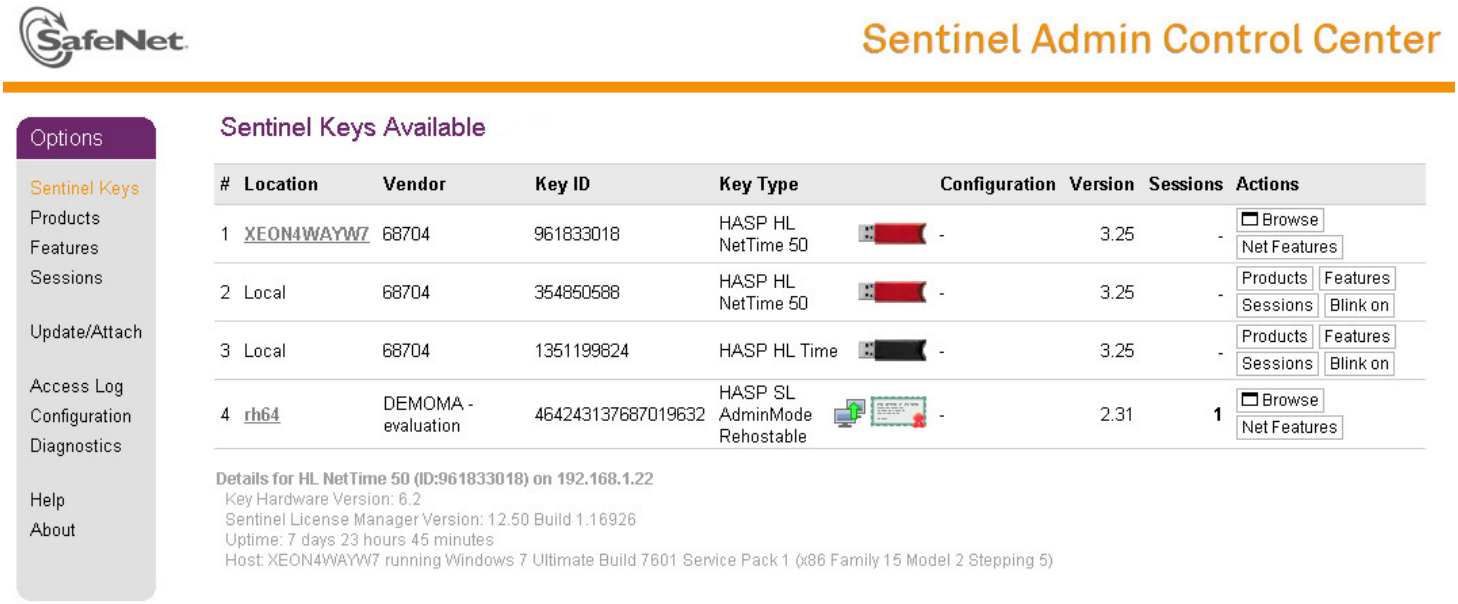
Sentinel Admin Control Center is not able to display CHARON-VAX / CHARON-PDP licenses - to view key contents, use the "HASP View" utility.

© Stromasys, 2017

© Stromasys, 2017

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

Example:



This example demonstrates that 4 license keys are available:

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

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

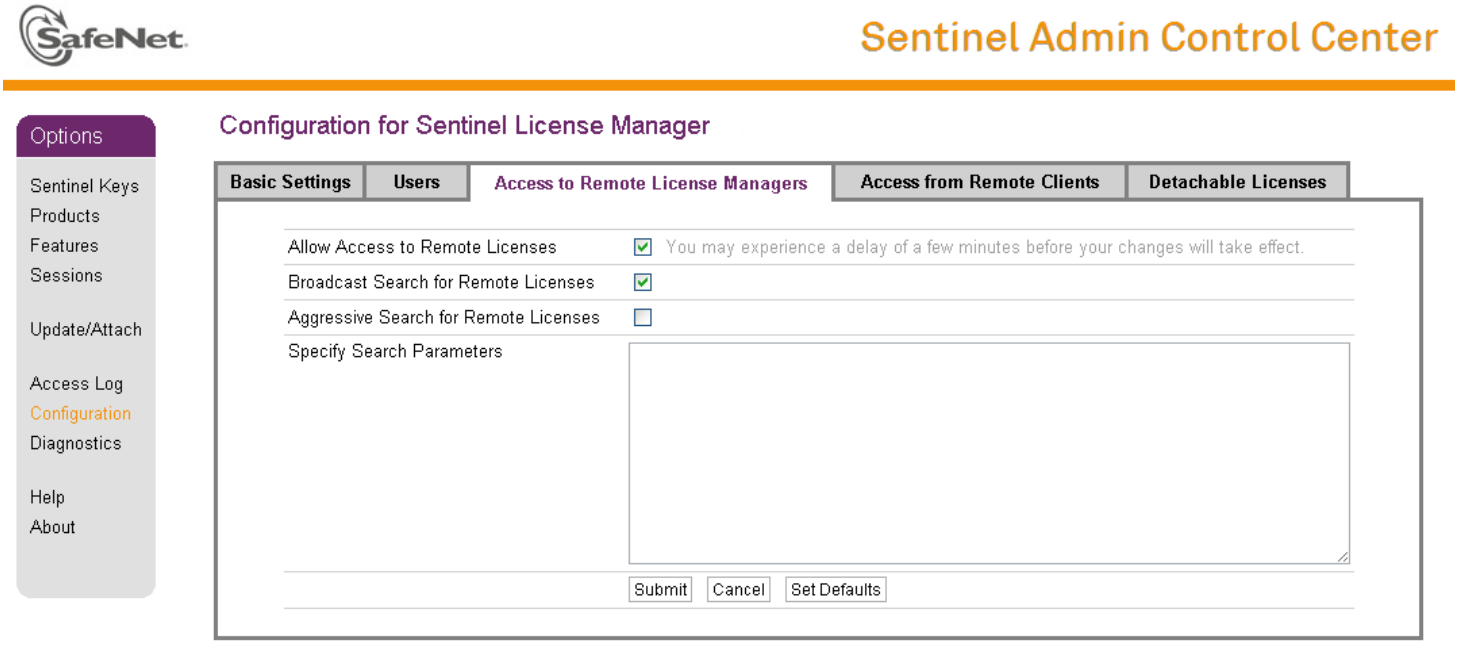
For a more detailed description of Sentinel Admin Control Center, please refer to its "Help" section.

Back to Table of Contents

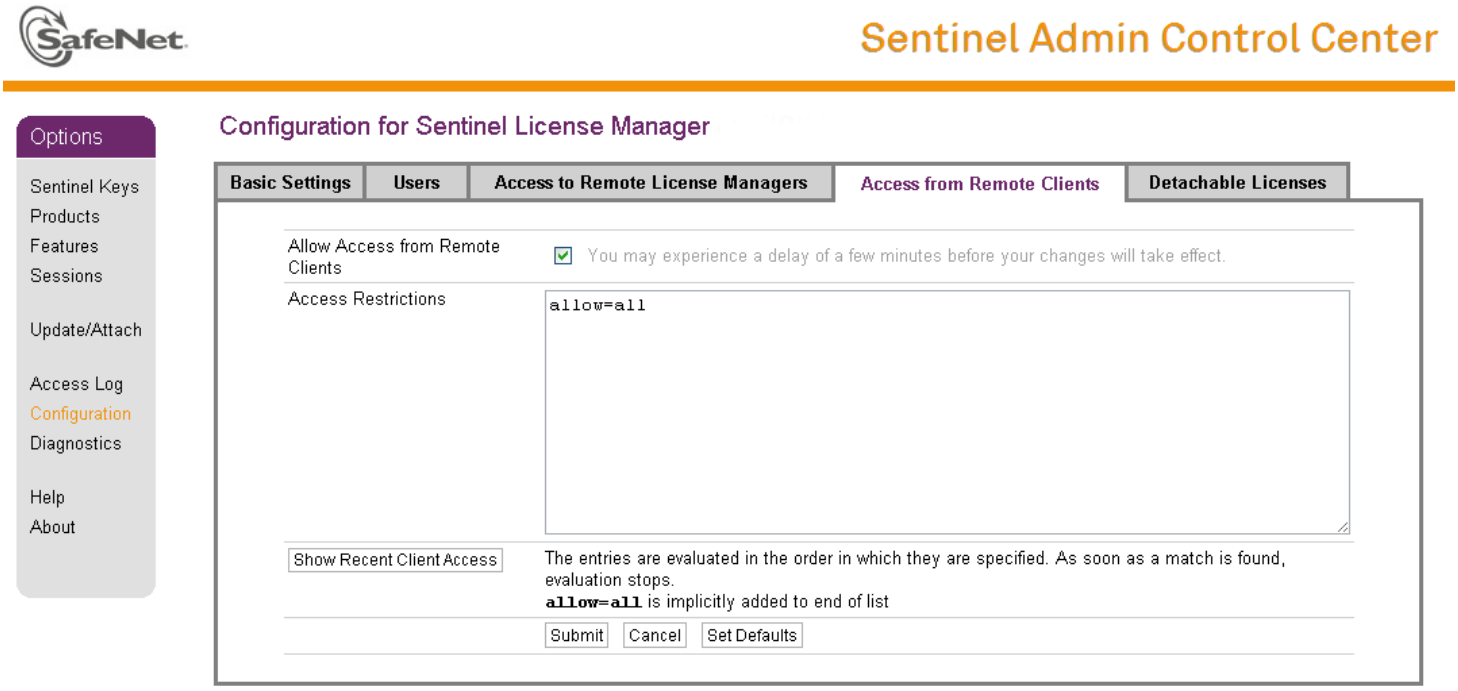
Disable remote keys access

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

To disable access to remote keys switch to the "Access to Remote License Managers" tab and uncheck the "Allow Access to Remote Licenses" 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 and uncheck the "Allow Access from Remote Clients" check box. Then press "Submit" button to apply this setting:



Back to Table of Contents

License management utility

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

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

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

Back to Table of Contents

Transferring and removing CHARON-VAX / CHARON-PDP software licenses

Software Licenses Transfer

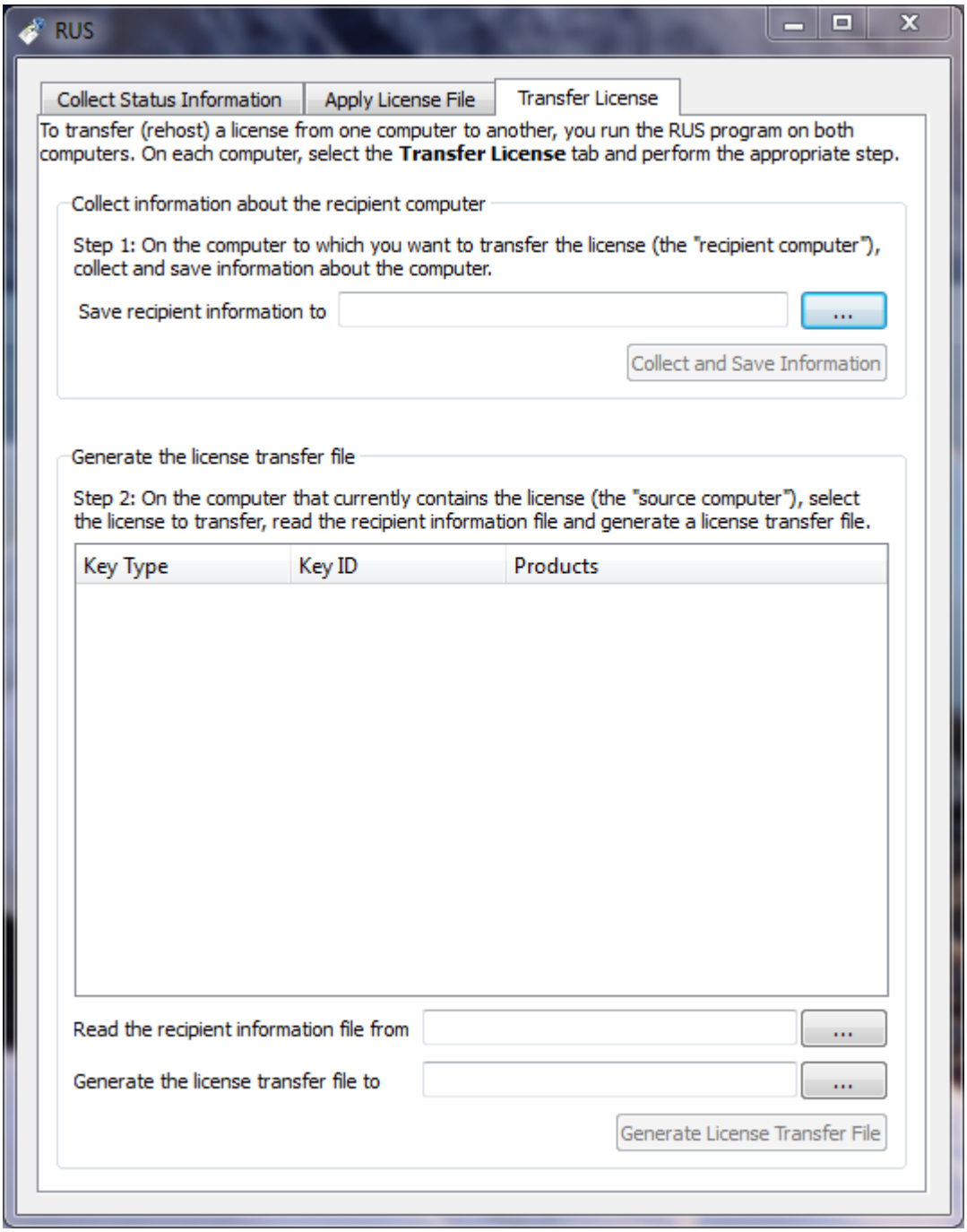
Software Licenses (SL) can be transferred from one host to another using the "License Update Service" utility.

The following example demonstrates the transfer procedure.

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

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

© Stromasys, 2017



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

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

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

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

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

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

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

Back to Table of Contents

Software License Removal

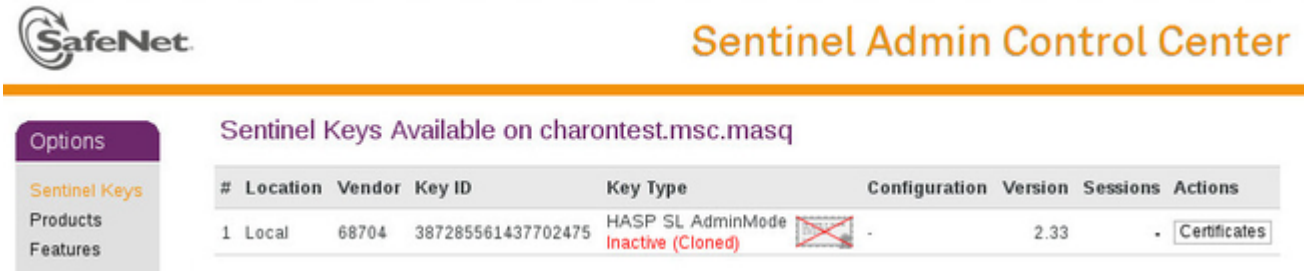
It is also possible to remove a Software License completely from a host, the license will then be stored in a specific transfer license file "*.h2h", so it can be re-applied if needed.To remove a Software License completely from a host, follow the license transfer procedure described above. It is possible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

Back to Table of Contents

Cloned Software License Removal

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

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



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

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

Back to Table of Contents

© Stromasys, 2017

License Deinstallation

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

- Login as Administrator
- Open the 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
- Issue:

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

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

Back to Table of Contents

Special "backup" license keys

Backup keys are provided by STROMASYS along with standard license dongles. It is strongly recommended to order a backup key to recover immediately from damage or loss of the main license key.Backup keys use a counter (integer) value hardcoded inside the key. This integer value is a number of hours CHARON-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 backup keys have restricted functionality:

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

Back to Table of Contents

© Stromasys, 2017

CHARON-VAX and CHARON-PDP for Windows utilities

Table of Contents

- General description
- Details of CHARON-VAX / CHARON-PDP Utilities

General description

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

Utility	Description
CHARON Launcher	Used to start CHARON and install it as a Windows service.
CHARON Service Manager	Used to manage CHARON services.
Host Device Check	Used to review system resources that can be mapped to CHARON.
Network Control Center	Used to configure a CHARON network.
MkDisk	GUI-based utility used to create custom or standard CHARON virtual disk containers.
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.
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.
Performance Improvement package	Used to improve and manage CHARON performance.
HOSTPrint	Used to print CHARON output to Windows printers.
VT-30TV	TV-30TV screen emulation.

Back to Table of Contents

Details of CHARON-VAX / CHARON-PDP Utilities

- CHARON Launcher
- CHARON Service Manager
- Host Device Check
- Network Control Center
- MkDisk
- MkDskCmd
- HASP View
- License Update Service
- mtd
- CHARON Log Monitor and Dispatcher
- Performance Improvement Package
- HOSTPrint
- VT-30TV

© Stromasys, 2017

CHARON Launcher

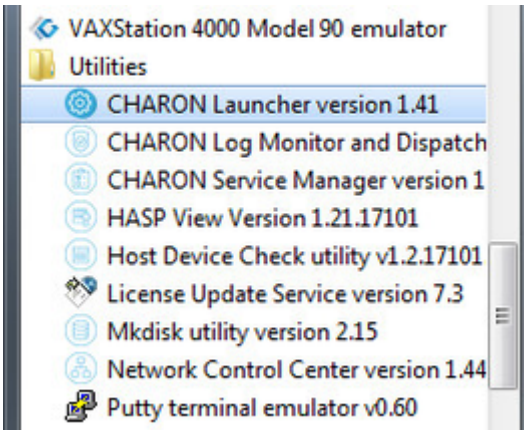
Table of Contents

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

Back to Table of Contents

General Description

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

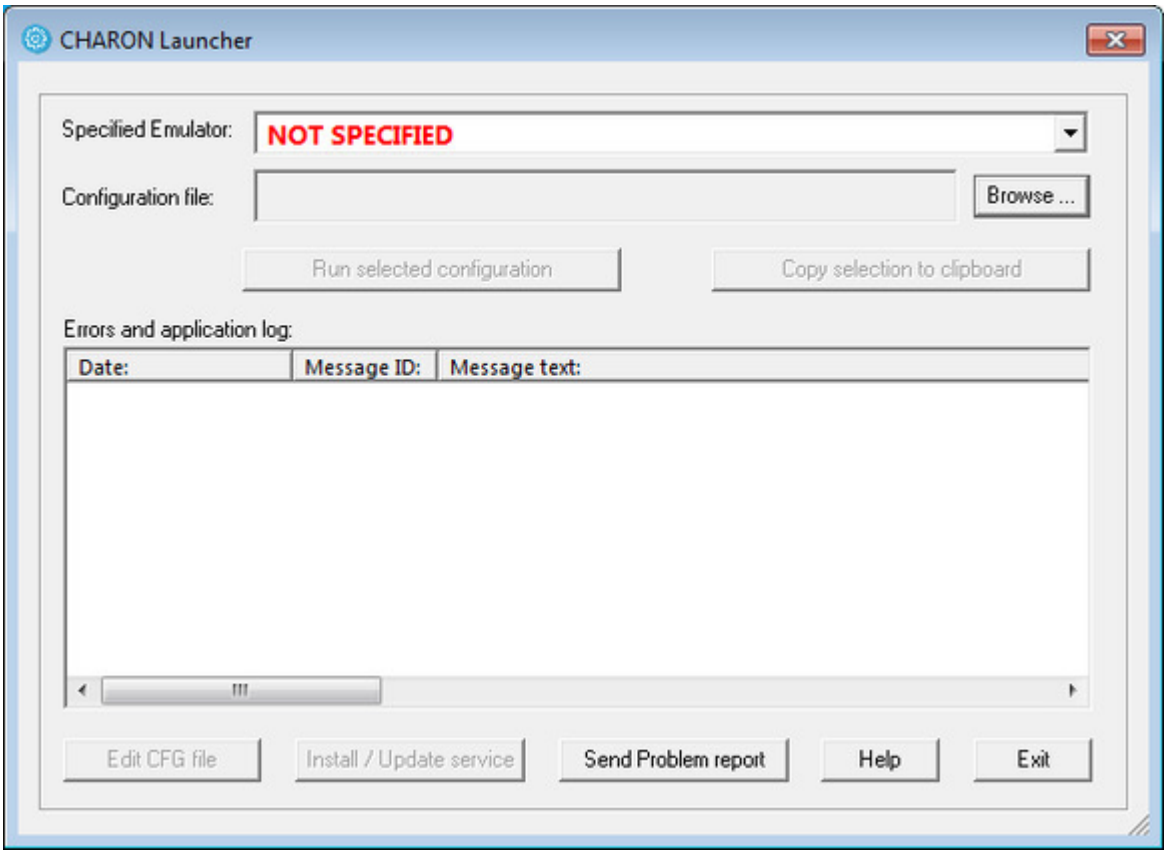


Back to Table of Contents

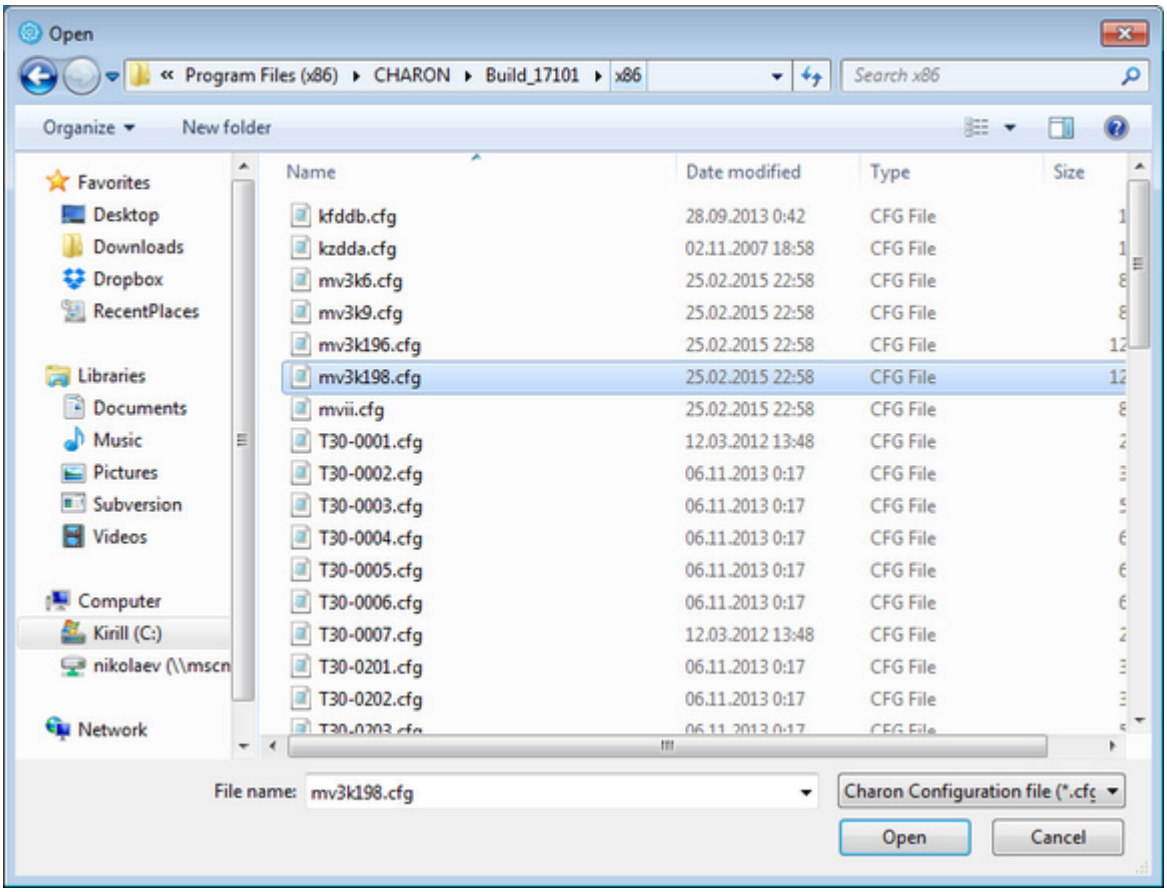
© Stromasys, 2017

Running Charon

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

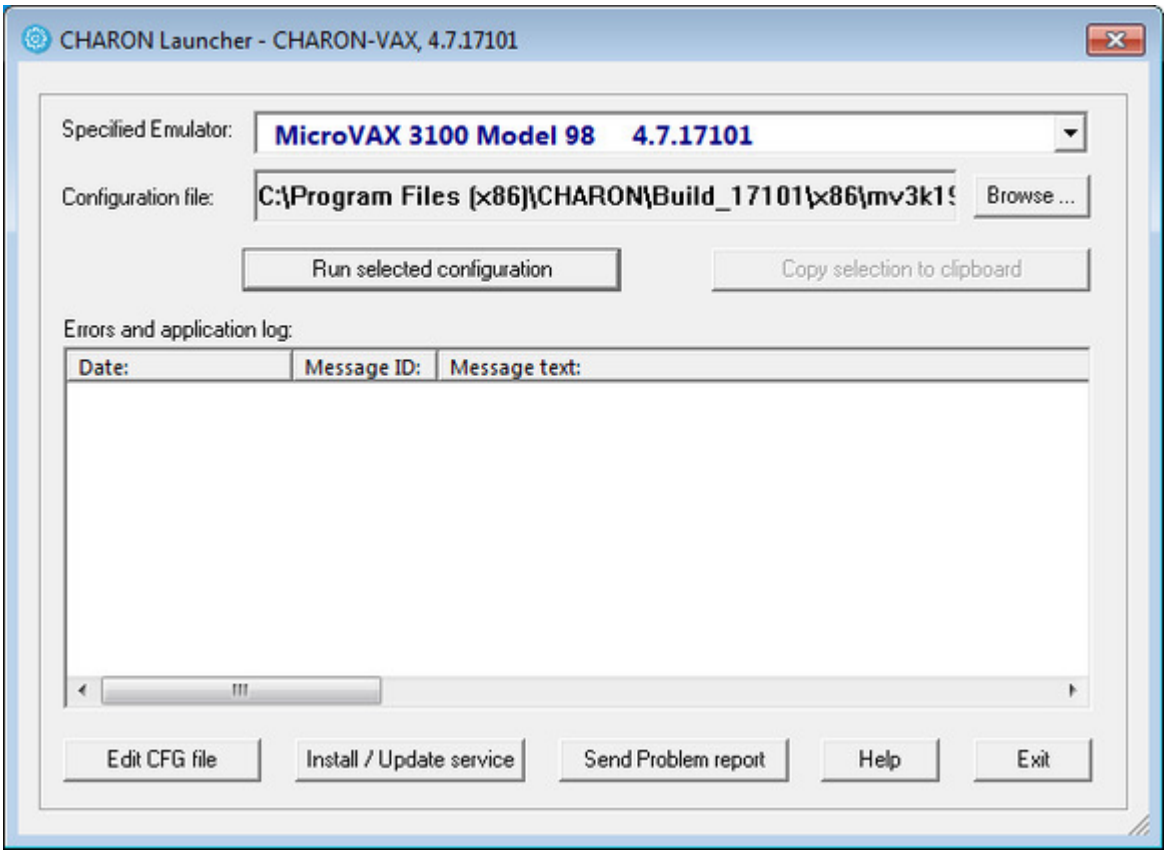


Select the desired configuration file:



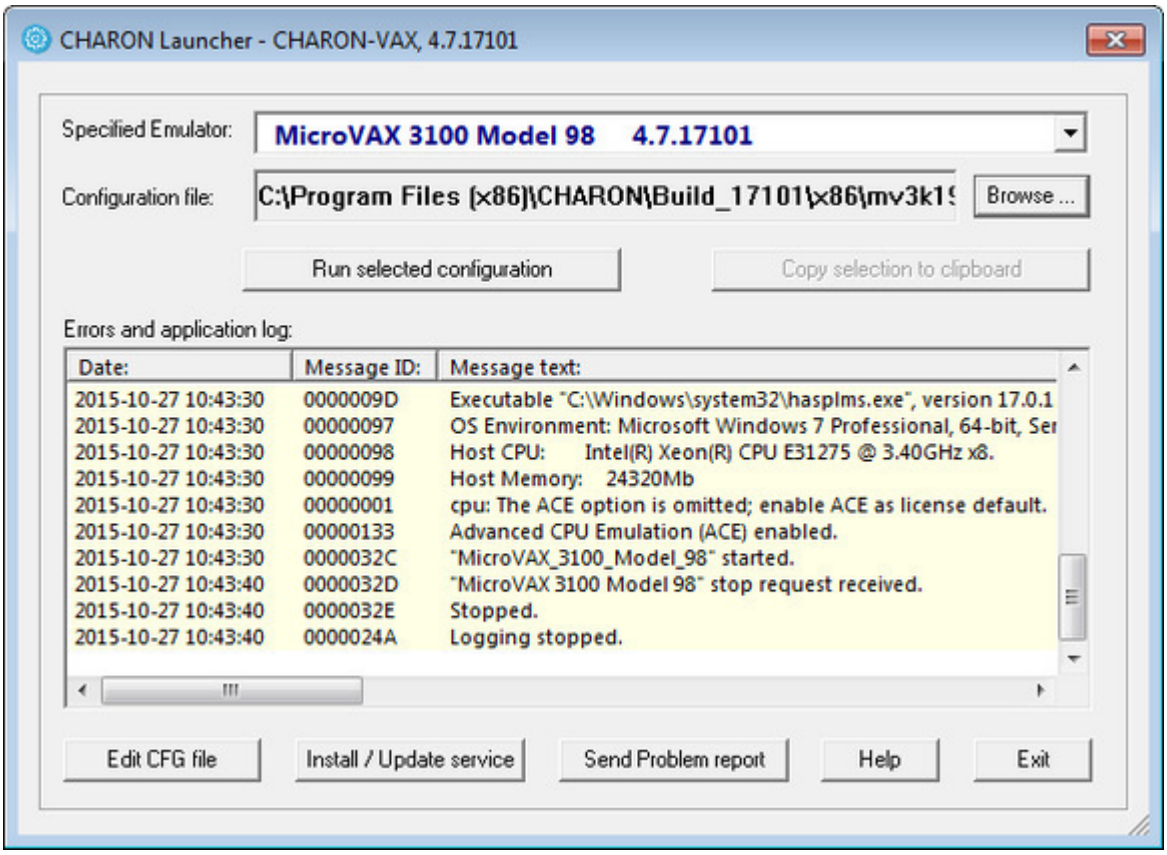
© Stromasys, 2017

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



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

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



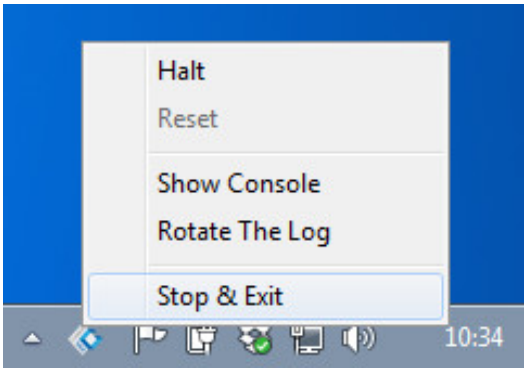
Log messages have the following colors:

Color	Error level
lightyellow	Information
cyan	Warning
red	Error

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

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

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



  - Using this menu, it is also possible to halt, reset or show the virtual VAX/PDP11 console, if it has been closed.
- **Configuration file updated to enable "F6" button in configuration file to trigger exit from CHARON:**

```
#------------------------------------------------------------------------------
#
# Uncomment to allow 'F6' to terminate the running emulator.
#
#------------------------------------------------------------------------------

set OPA0 stop_on = F6
```

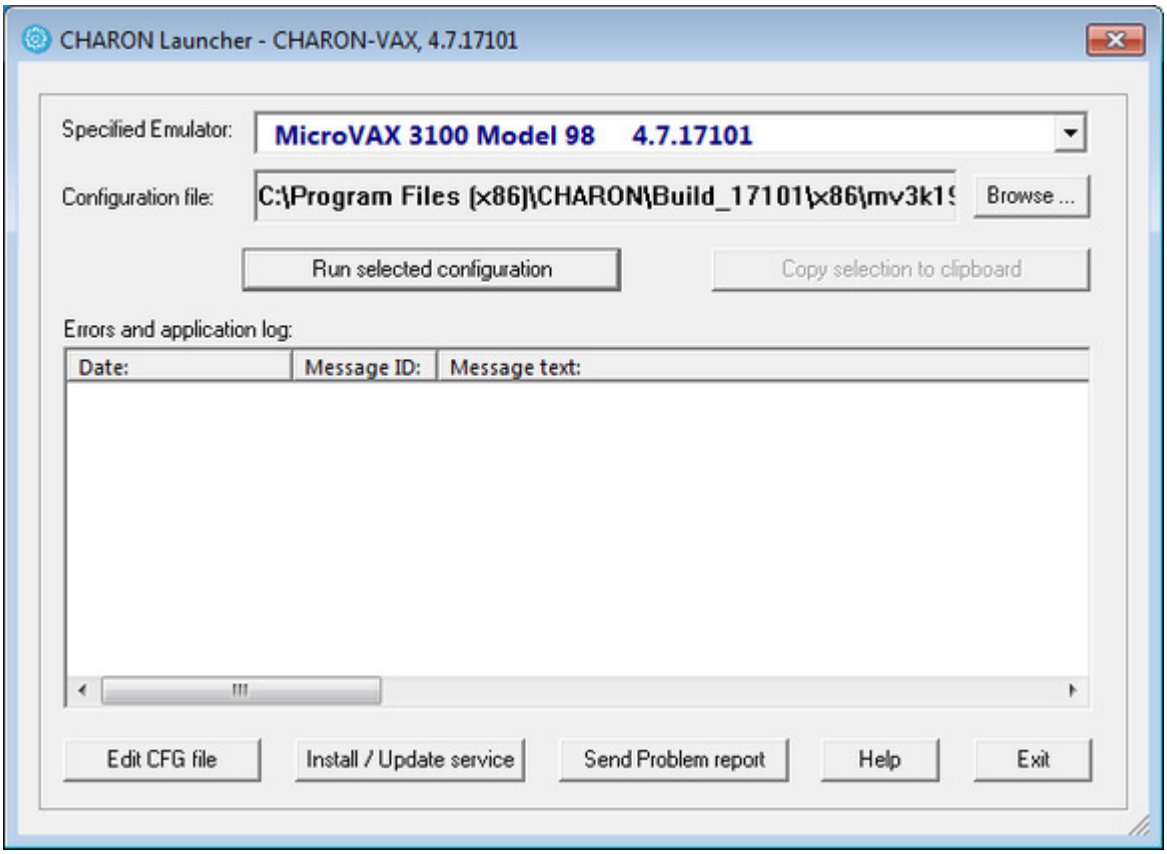
  - To exit, press "**F6**" in the console window
  - Use CHARON icon tray menu as shown above

⚠ Before stopping the CHARON-VAX / CHARON-PDP service, shutdown the operating system running in CHARON-VAX / CHARON-PDP.

Back to Table of Contents

Installation of CHARON as a service

Load the CHARON configuration file as it is described above:



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

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

```
...

set session log="C:\My Services\my_mv3k6.log"
set toy container="C:\My Services\my_mv3k6.dat"

load RQDX3/RQDX3 DUA
set DUA container[0]="C:\My Services\mv3k6_daemon_boot_disk.vdisk"
...
```

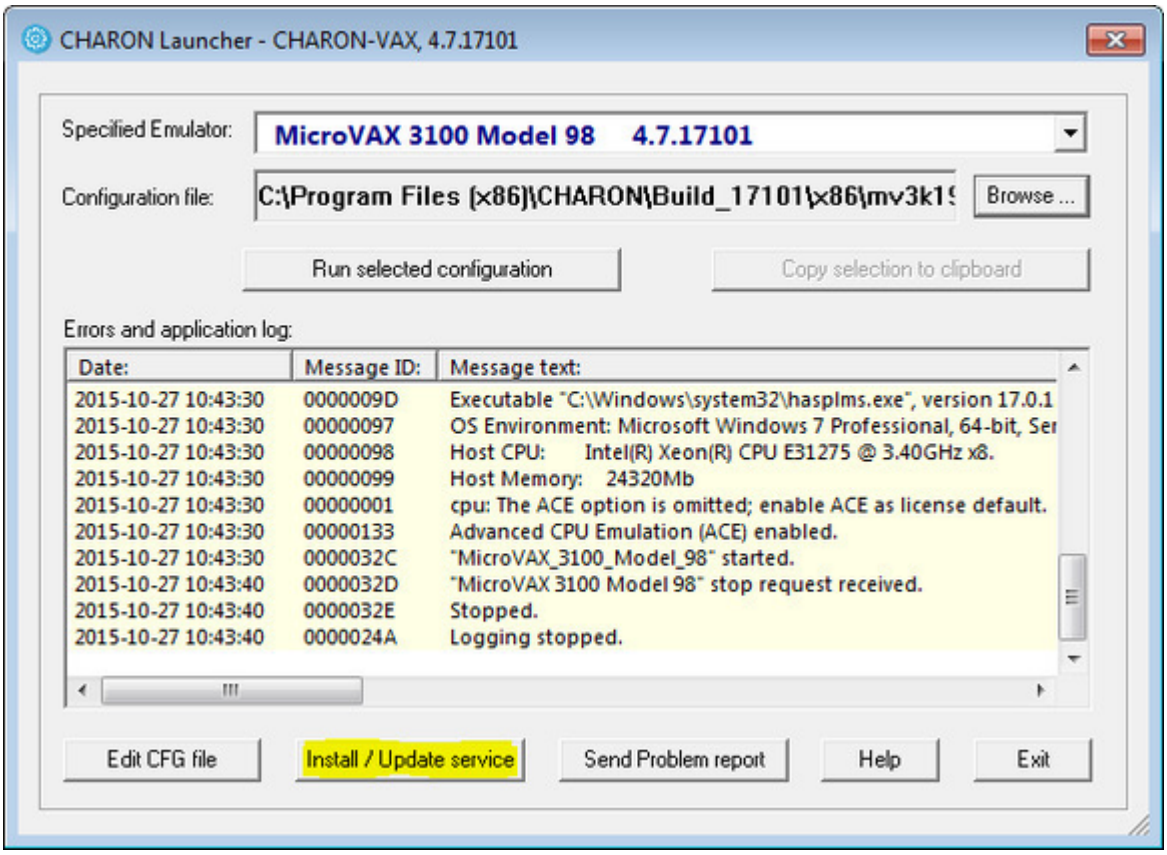
- Make sure the same physical devices are not used by other CHARON-VAX / CHARON-PDP services

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

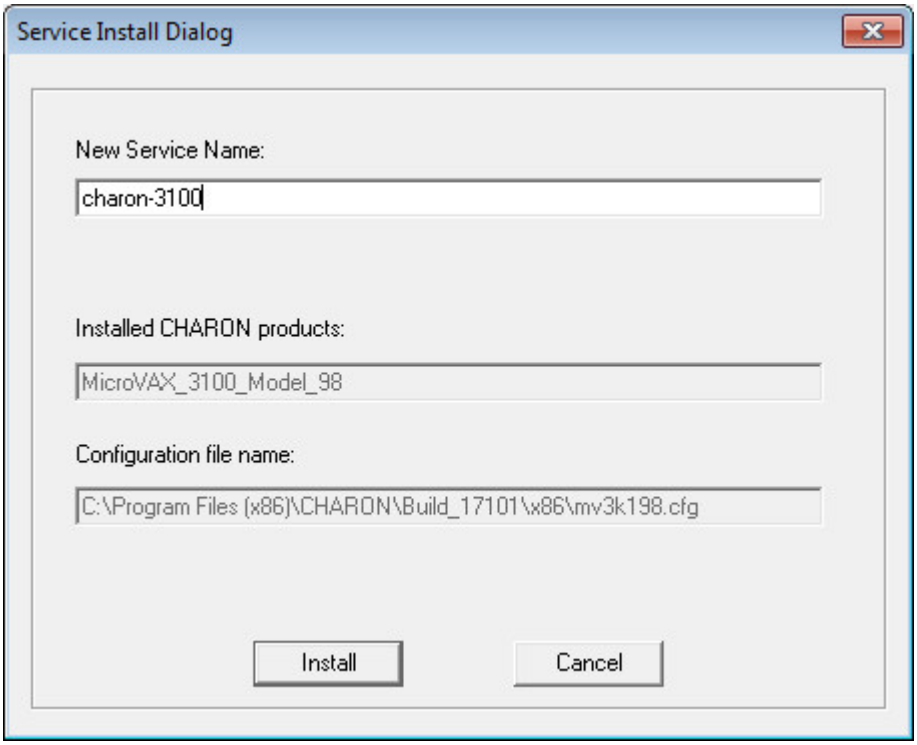
Run the configuration to confirm it is ready to be installed as a Windows service; stop CHARON (see above).

© Stromasys, 2017

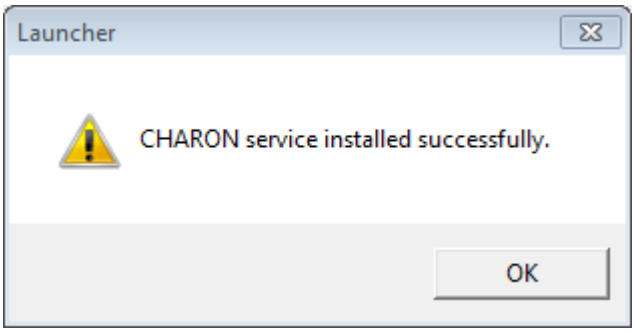
Press the "Install / Update service" button in the Launcher:



A special dialog for adding new CHARON services will appear:



Enter a new service name (in the example above it is "charon-3100") and press the "Install" button. If the service has been installed correctly, a configration dialog box will appear:

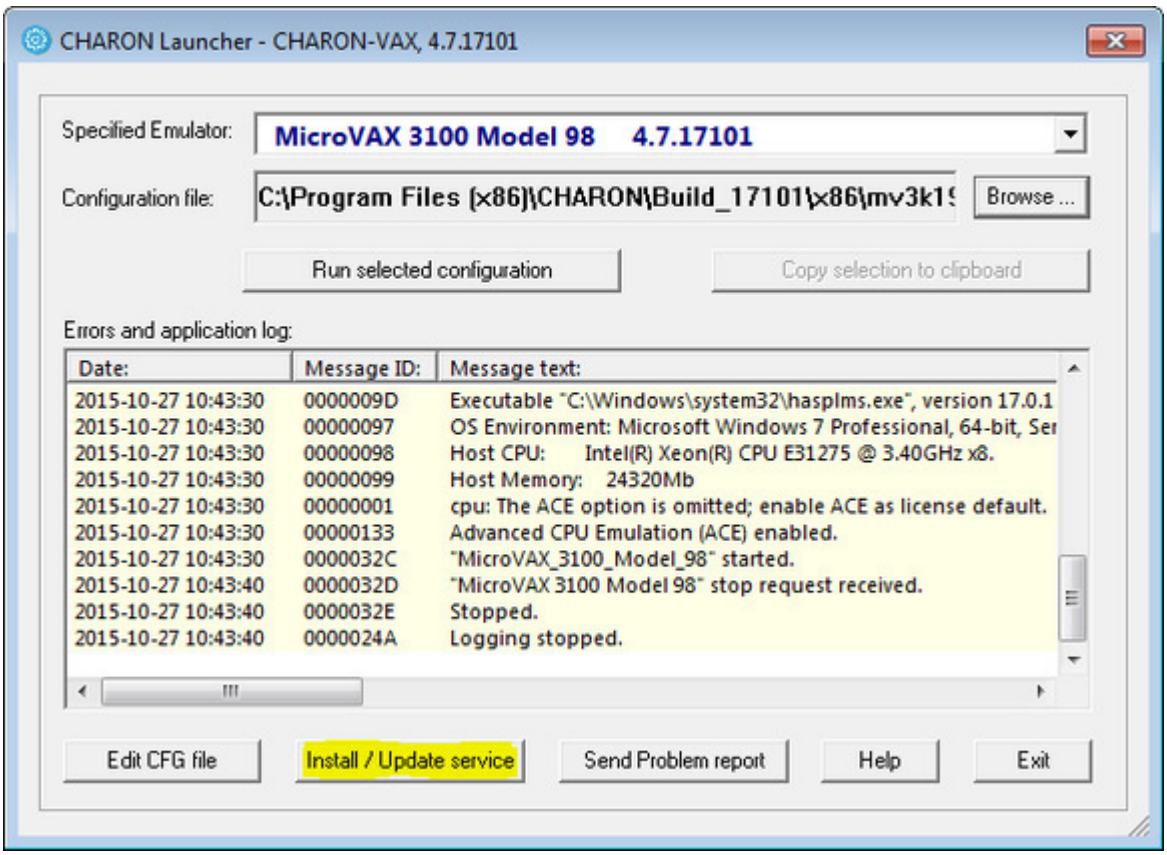


Back to Table of Contents

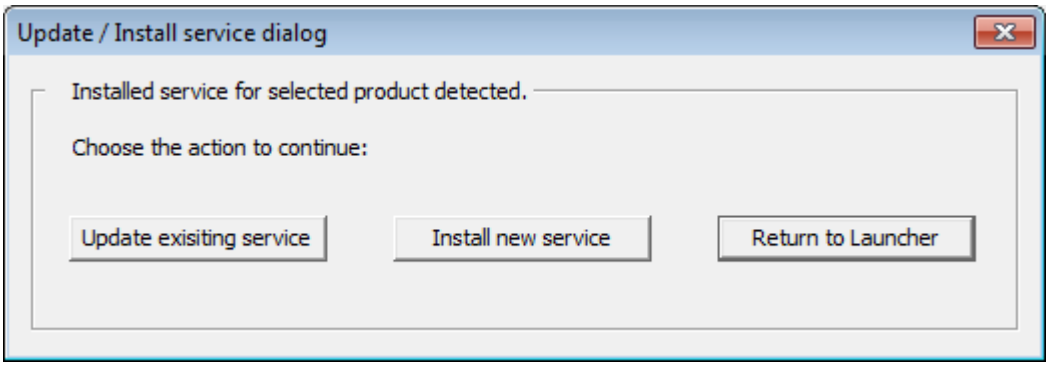
Modification of CHARON service

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

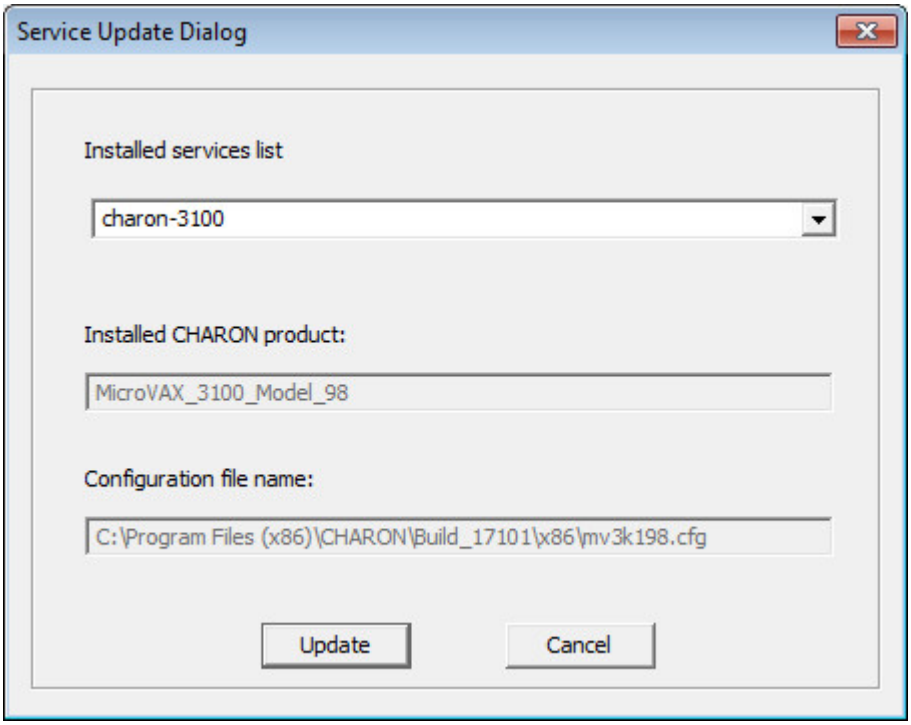
Press the "Install / Update service" button:



The install / update dialog will appear:



Press "Update existing service" button:



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

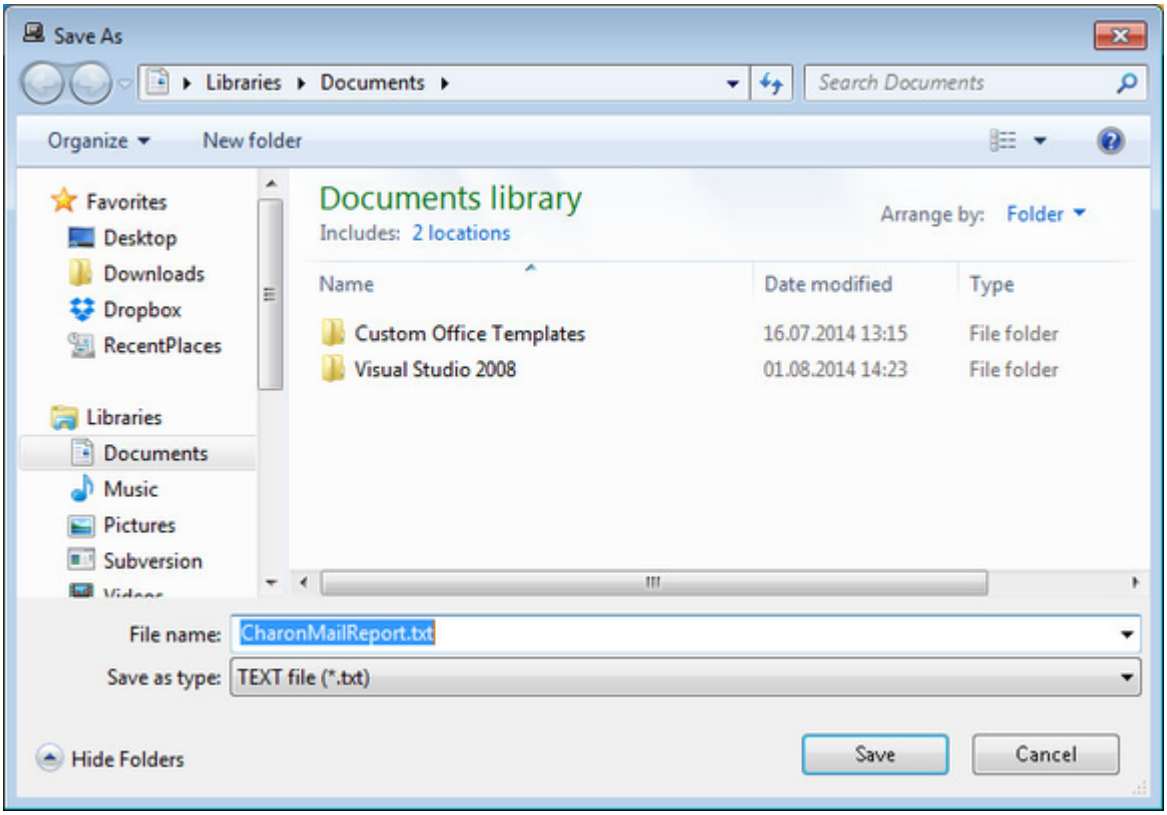
⚠ Please note that before updating a CHARON-VAX / CHARON-PDP service, it must first be stopped

Back to Table of Contents

© Stromasys, 2017

Sending problem reports

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



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

Back to Table of Contents

© Stromasys, 2017

CHARON Service Manager

Table of Contents

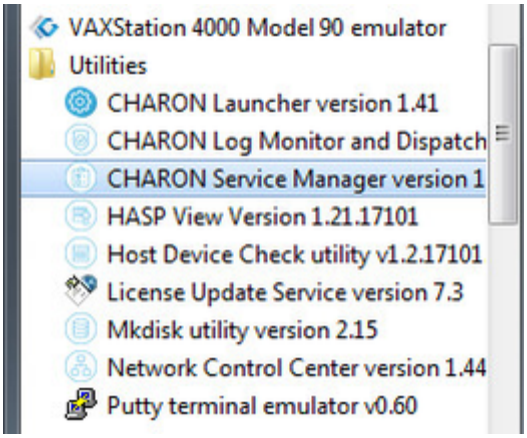
- General Description
- Starting CHARON service
- Managing CHARON service
- Stopping CHARON service
- Removing CHARON service
- Scheduling CHARON service
- Security settings
- Sending problem report
- Printing the CHARON service log file
- Reviewing the CHARON service configuration
- Tracing system resources

Back to Table of Contents

General Description

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

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



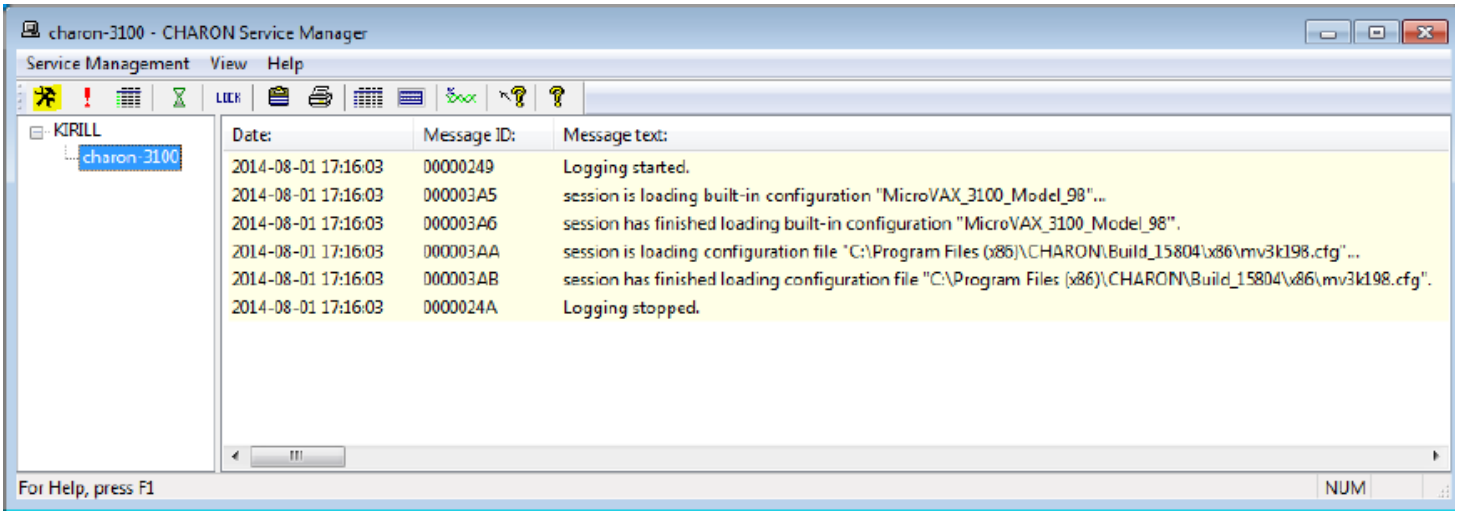
It is also possible to start the "CHARON Service Manager" utility from the system tray menu:

Back to Table of Contents

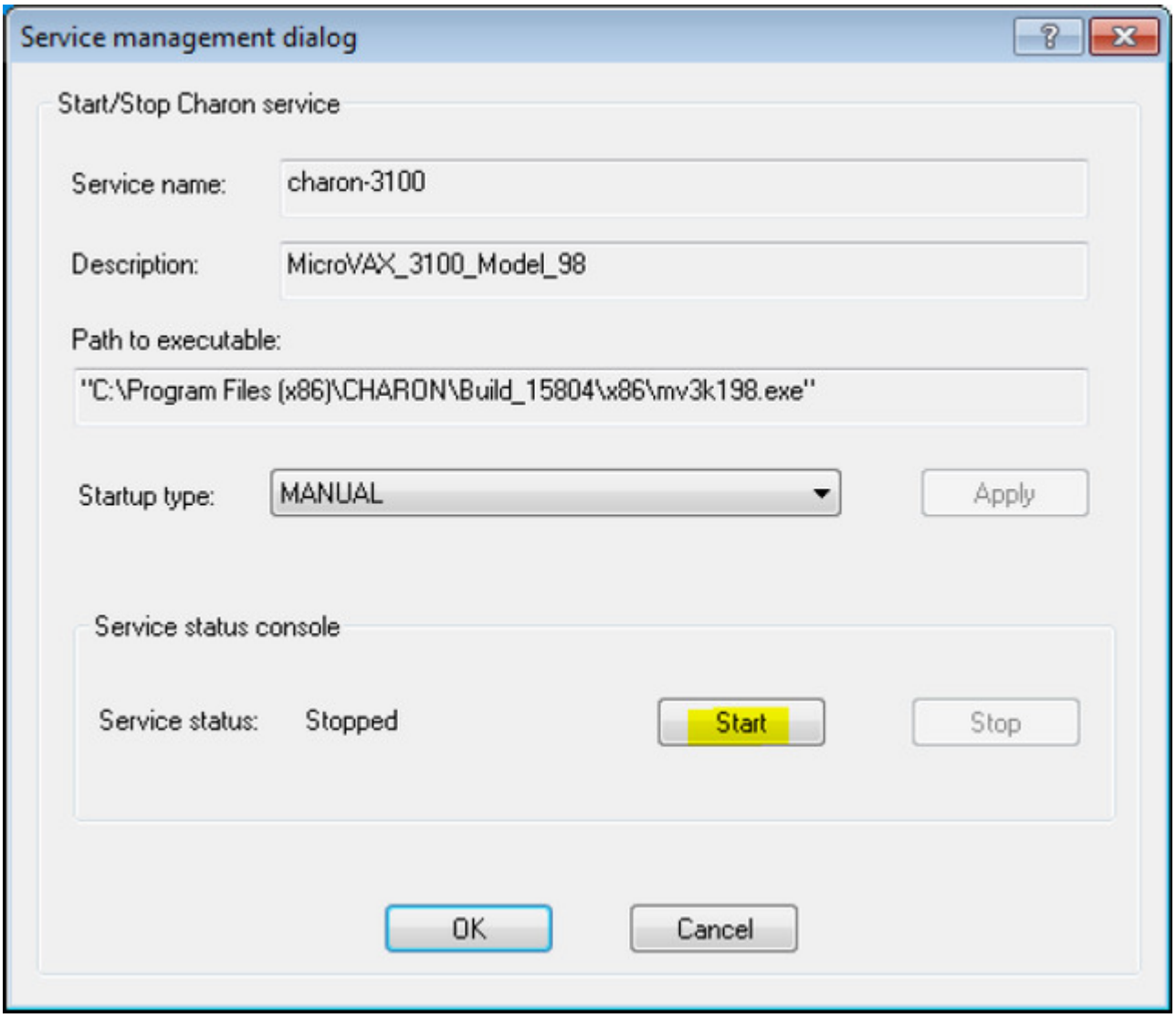
© Stromasys, 2017

Starting CHARON service

In the main window of the utility, select the target CHARON service ("charon-3100" in our example) and press the "Service" button (labeled with yellow on the screenshot below):



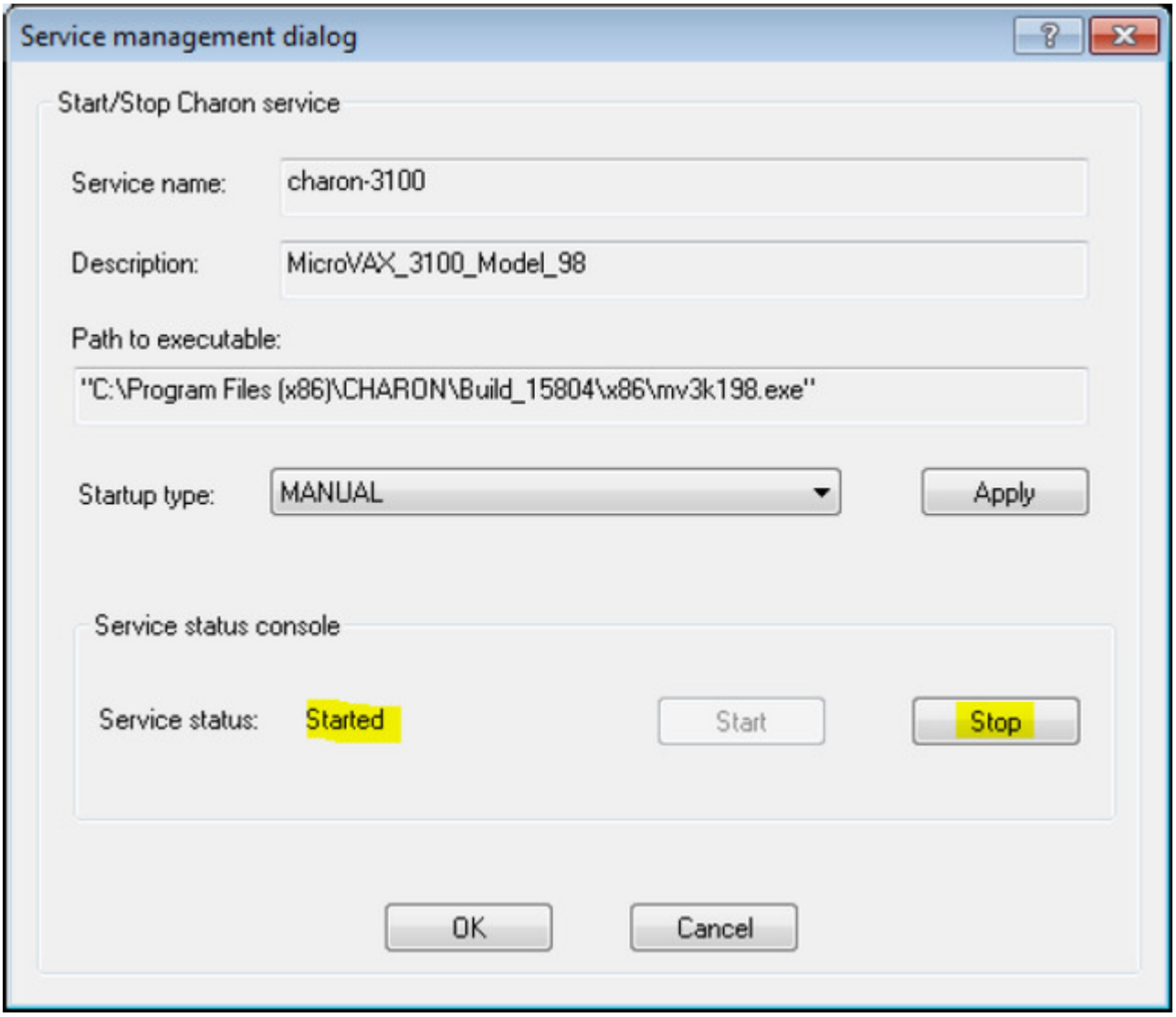
The "Service Management dialog" will appear:



The "Service Management dialog" shows the current status of the selected CHARON service and permits changes to its start up type and to start and stop it. The example screenshot above shows that the service "charon-3100" startup type is "MANUAL" (can be started / stopped only manually) and in the "Stopped" state - these are the default values.

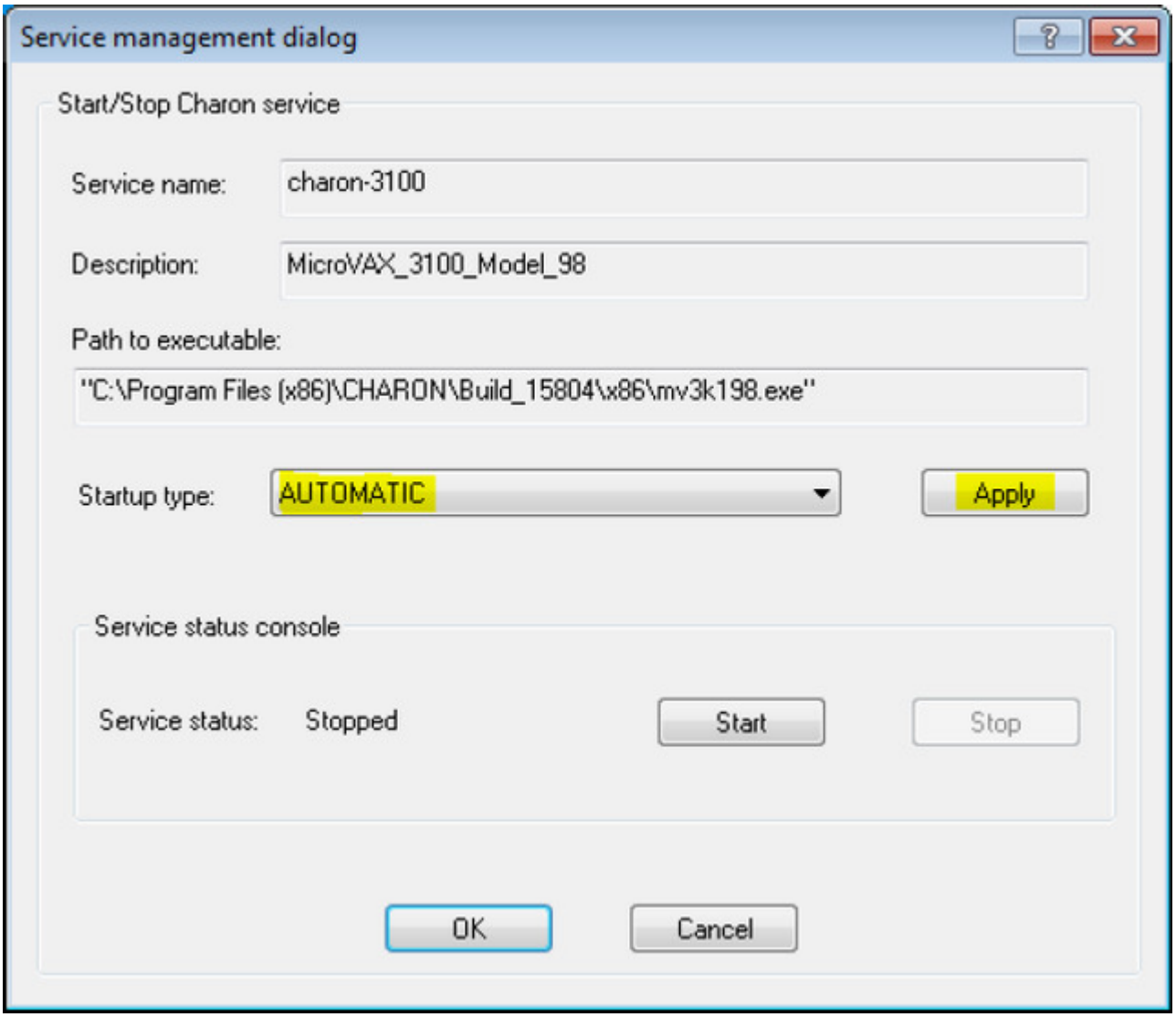
© Stromasys, 2017

Press the "Start" button to start the service:



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

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



Note that a certain delay may occur before the Sentinel Run-time process finds the network license on CHARON-VAX/PDP11 host system startup. If the CHARON-VAX/PDP11 service is starting automatically on the host system, it may report a "License not found" error and exit.

This problem can be avoided by specifying "license_key_lookup_retry" parameter in the configuration file:

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

where:

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

Example:

```
set session license_key_lookup_retry = 5
```

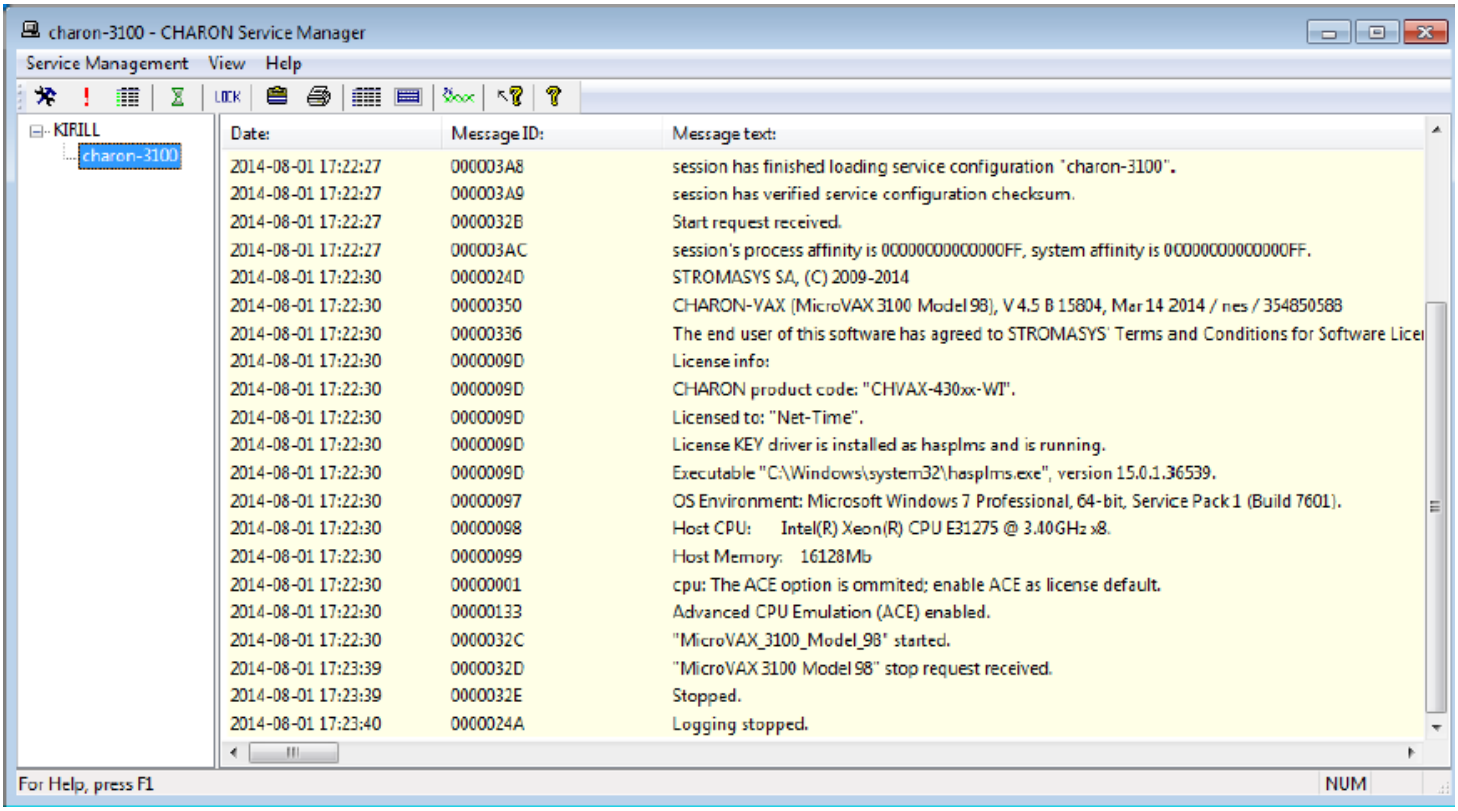
In this example, if the license key is not found during the initial scan, CHARON-VAX/PDP11 will scan 5 more times, waiting 60 seconds between them before it stops.

See General Settings section for more details.

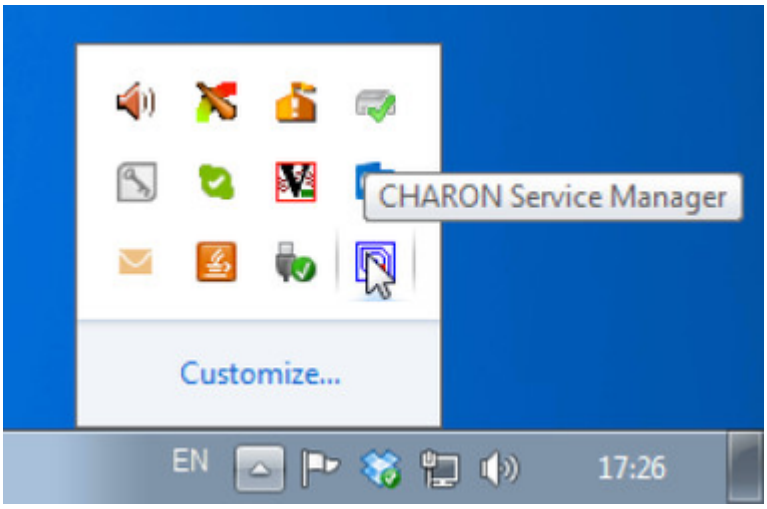
Back to Table of Contents

Managing CHARON service

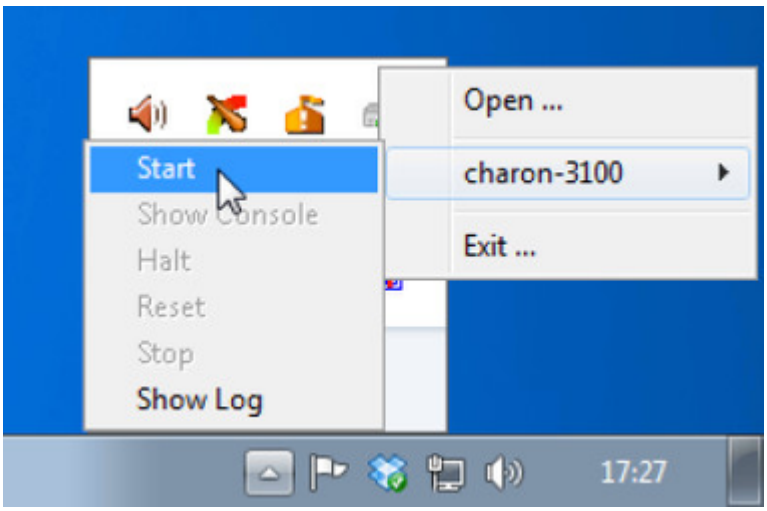
To see the execution log of a CHARON service, select the service ("charon-3100" in the example below) and the current log messages will be displayed in the right panel of the utility interface. You may need to scroll up to see startup/previous messages:



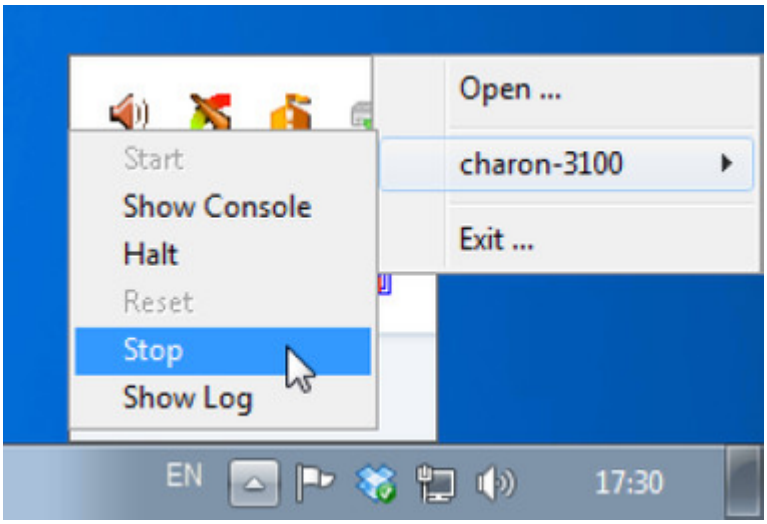
To control execution of CHARON services, open the "CHARON Service Manager" utility from the "Start" menu or use the system tray shortcut to manage the services w/o opening the main window of the utility:



Press the right button of the mouse to see a list of the installed CHARON services ("charon-3100" in our example):



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



Explanation:

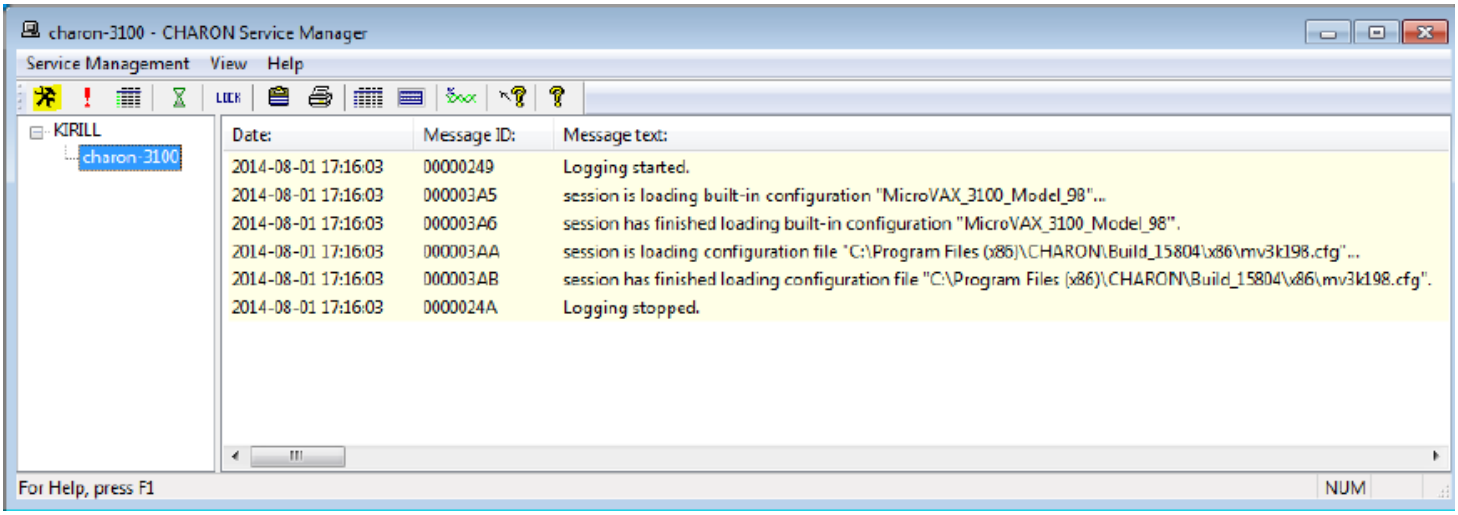
Menu Item	Function
Start	Start CHARON service
Show Console	Show CHARON service console if it has been closed or not active on the automatic service startup
Halt	Halt VAX/PDP11 emulator associated with the selected CHARON service
Reset	Reset the virtual VAX/PDP11 associated with the selected CHARON service
Stop	Stop CHARON service
Show Log	Show execution log of the selected CHARON service

Choose the desired action and click on it.

Back to Table of Contents

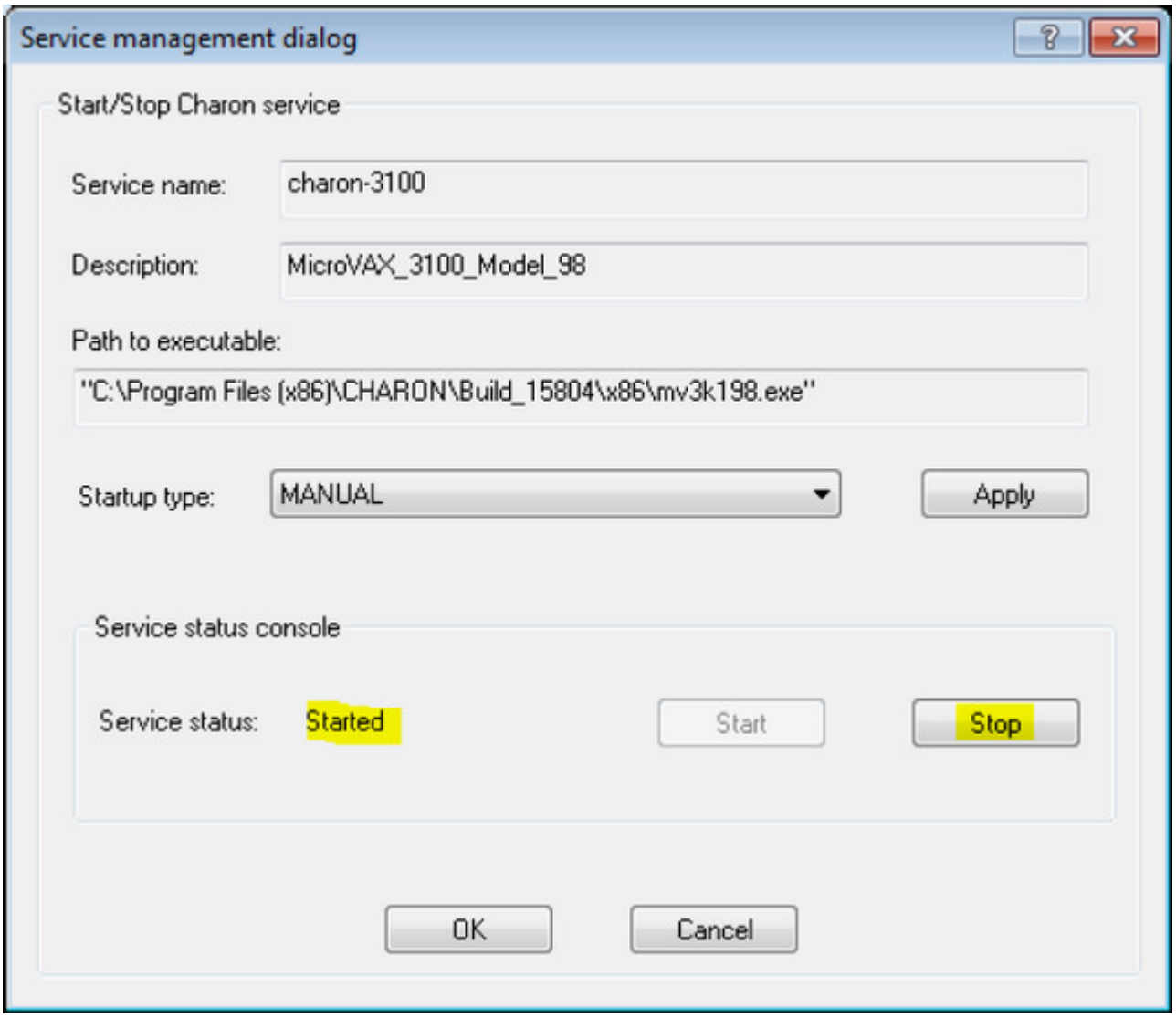
Stopping CHARON service

In the main window of the utility, select the target CHARON service ("charon-3100" in our example) and press the "Service" button (labeled with yellow on the screenshot below):



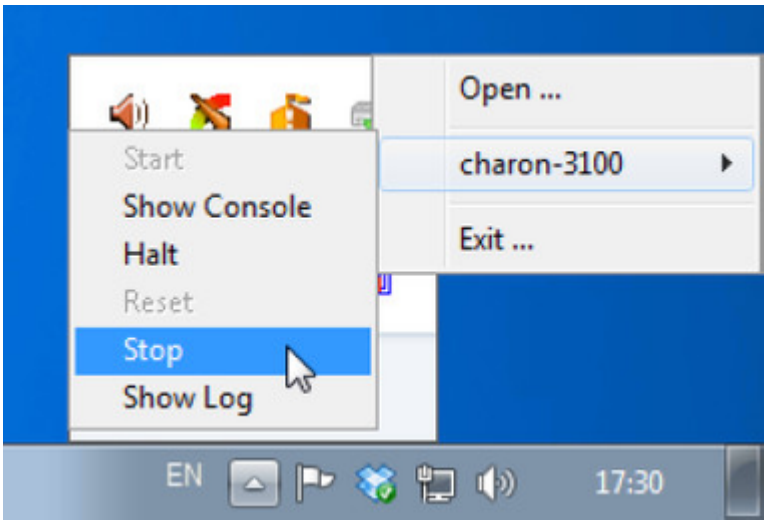
© Stromasys, 2017

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



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

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

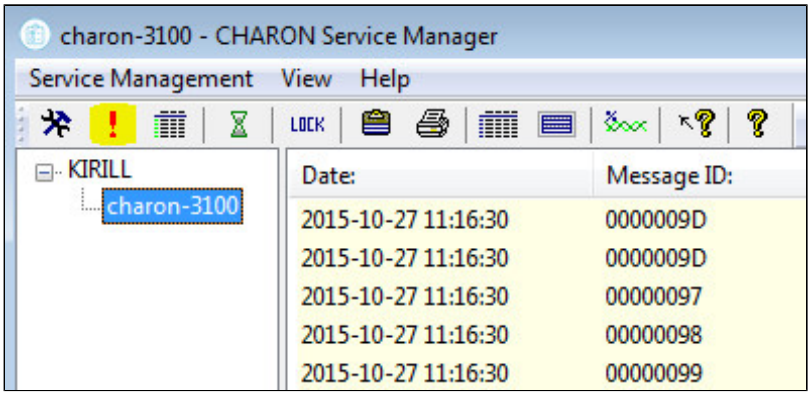


Please note: before stopping the CHARON-VAX/PDP11 service, shutdown the operating system running in CHARON-VAX/PDP11.

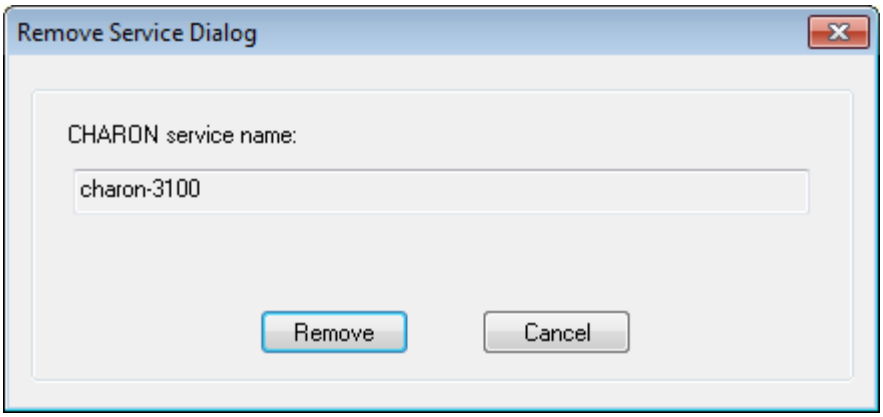
Back to Table of Contents

Removing CHARON service

In the main window of the utility select a CHARON service to remove ("charon-3100" in our example) and press the button with an exclamation sign on it:



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



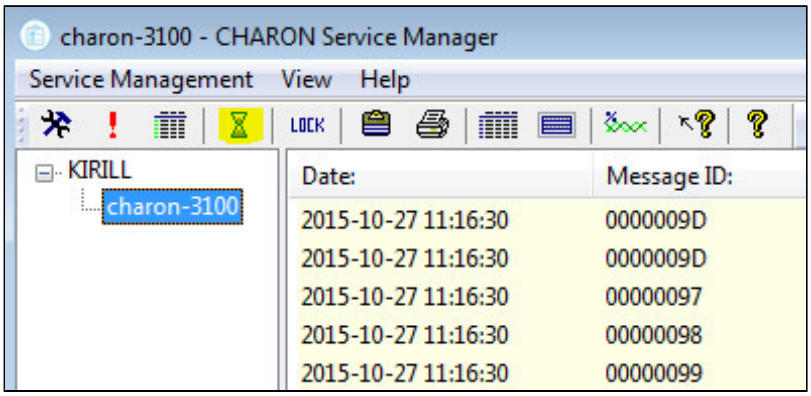
Back to Table of Contents

Scheduling CHARON service

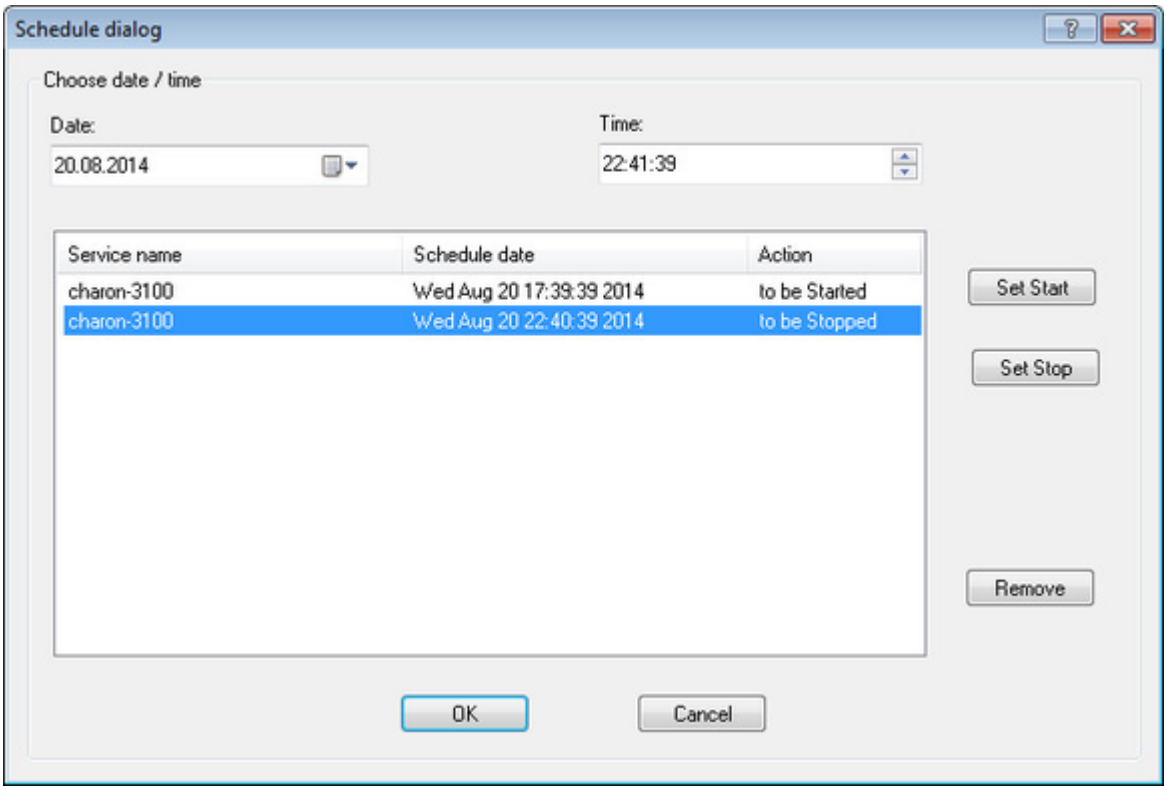
The "CHARON Service Manager" utility provides a function to set the start and stop date/time for each installed CHARON service.

© Stromasys, 2017

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



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



Press "Ok" to apply settings.

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

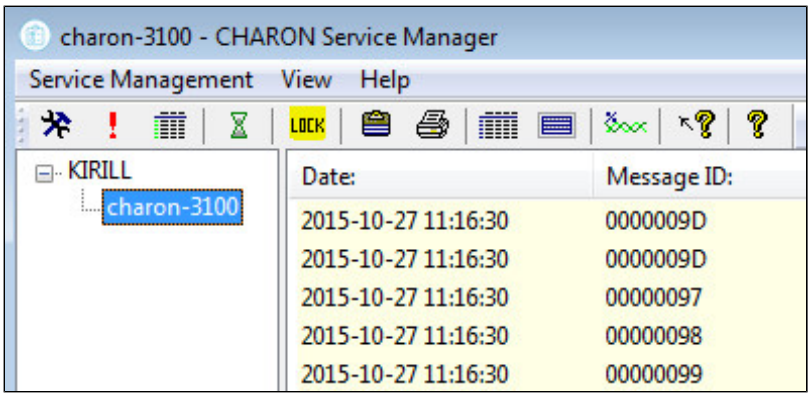
To remove the scheduled event select it and press the "Remove" button.

Back to Table of Contents

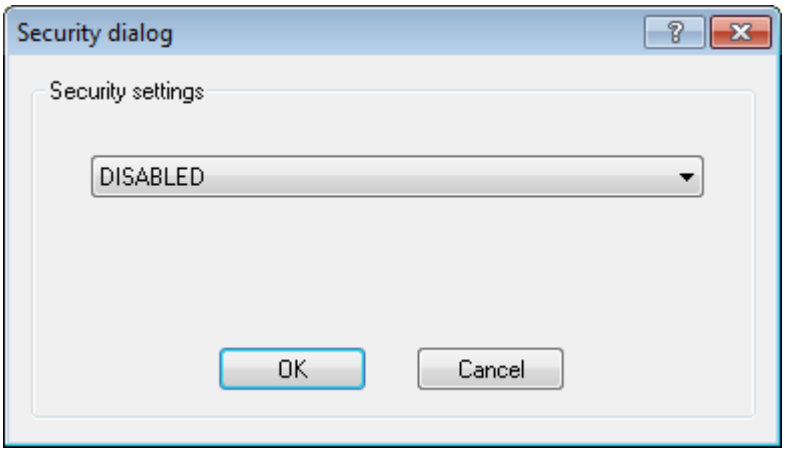
Security settings

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

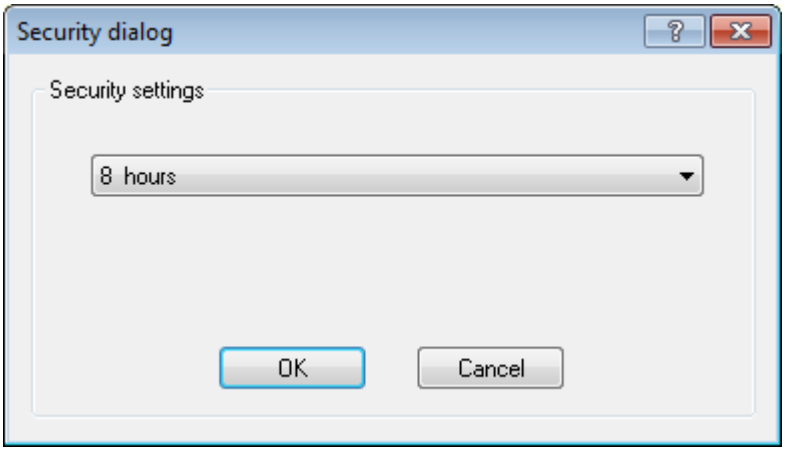
To enable this function press the "Lock" button:



The following popup will appear:



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

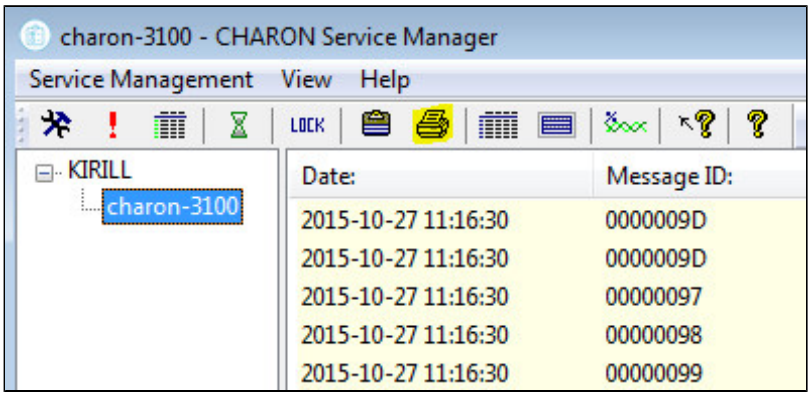


Note that the security settings are active only when the "CHARON Service Manager" is running. On exit from the utility the security settings reset.

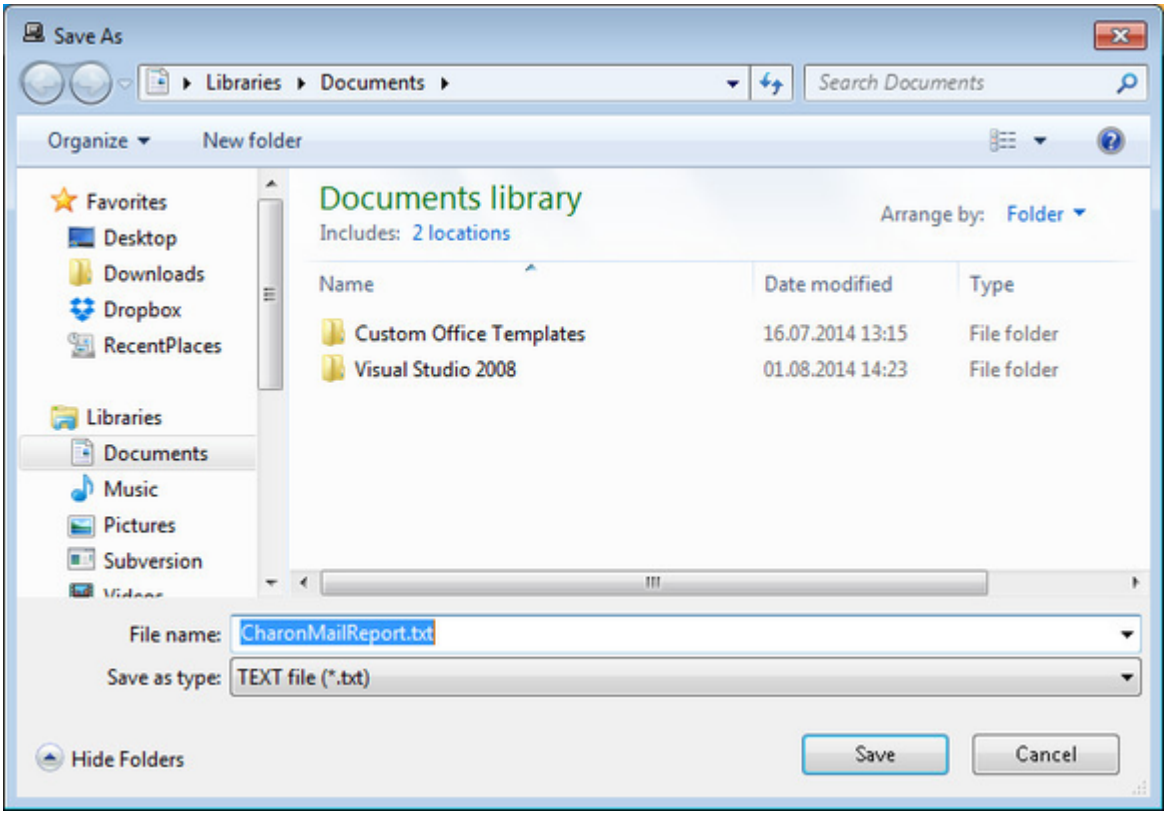
Back to Table of Contents

Sending problem report

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



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



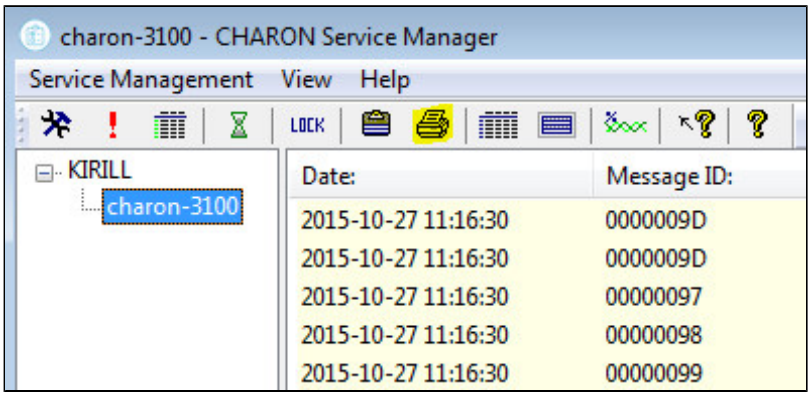
Send the file to the STROMASYS customer support along with a detailed description of the problem.

Back to Table of Contents

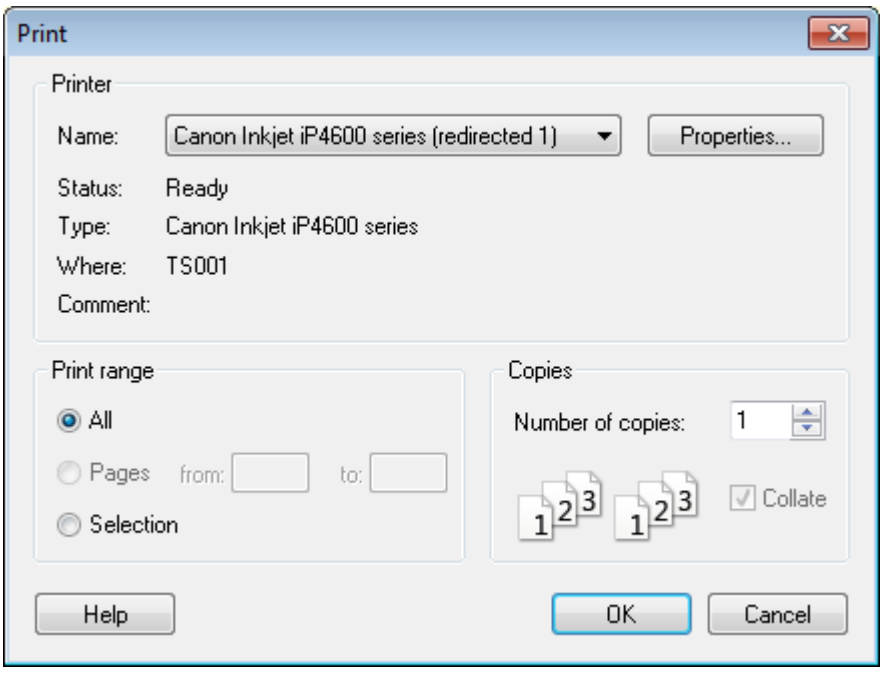
© Stromasys, 2017

Printing the CHARON service log file

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



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

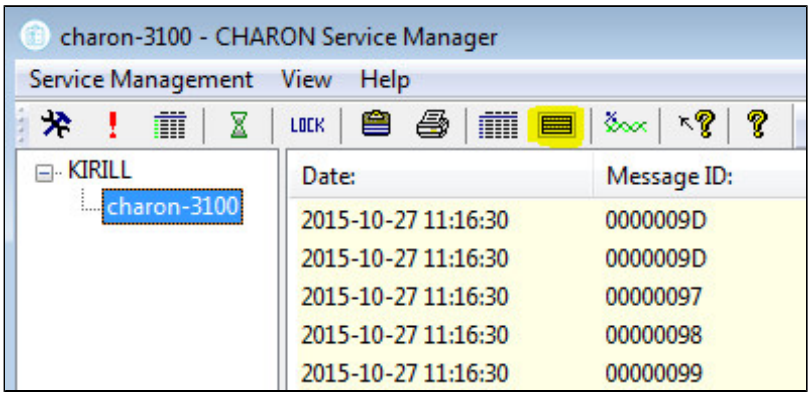


Back to Table of Contents

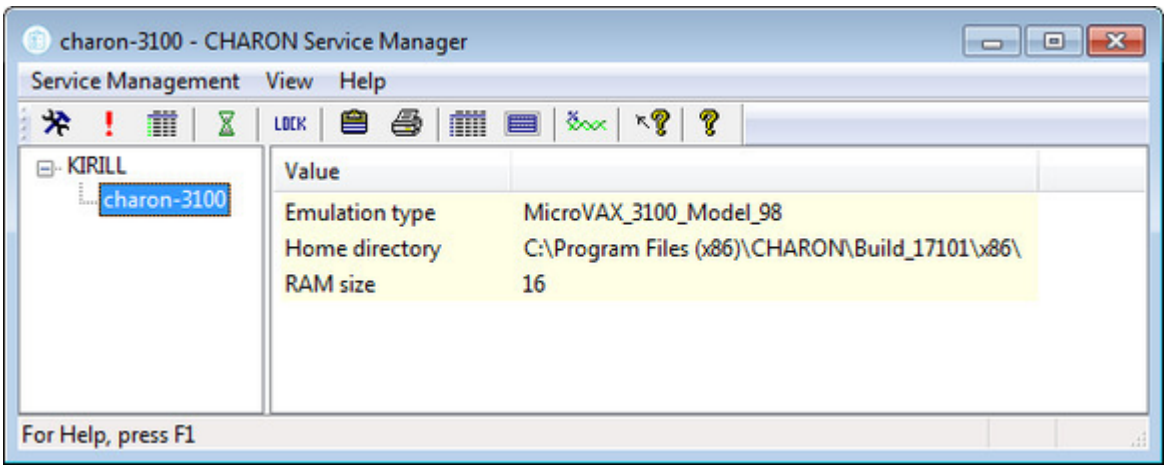
© Stromasys, 2017

Reviewing the CHARON service configuration

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



This utility will display the selected CHARON service configuration:



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

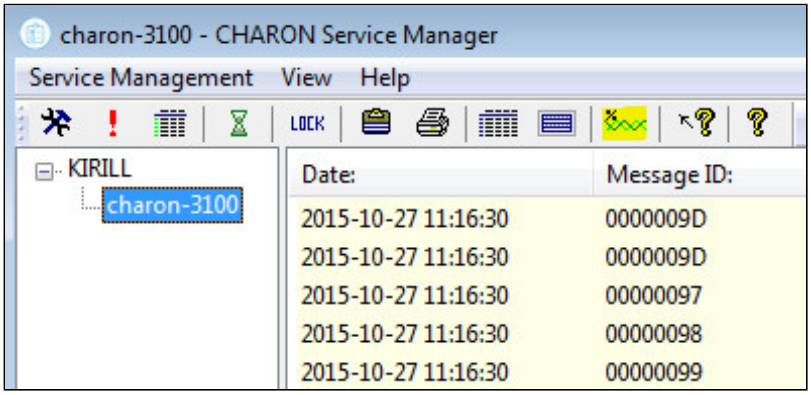
Back to Table of Contents

© Stromasys, 2017

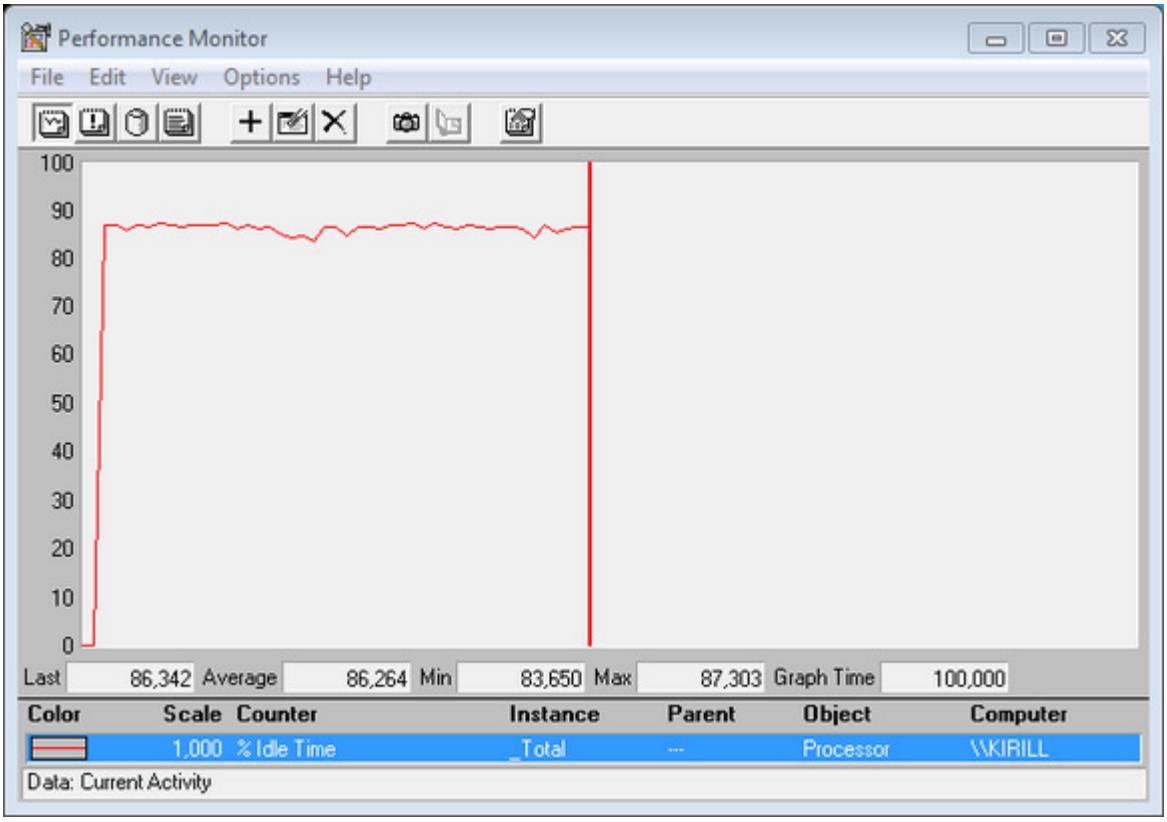
Tracing system resources

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

Click on the yellow highlighted button as shown below:



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



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

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

Back to Table of Contents

© Stromasys, 2017

Host Device Check

Table of Contents

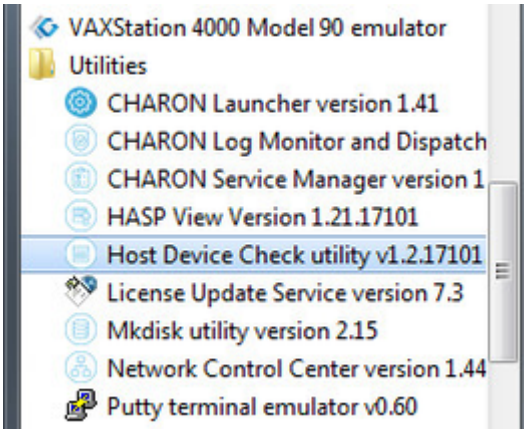
- General Description
- Reviewing available physical disks
- Reviewing all the available host resources
- Collecting the configuration strings

Back to Table of Contents

General Description

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

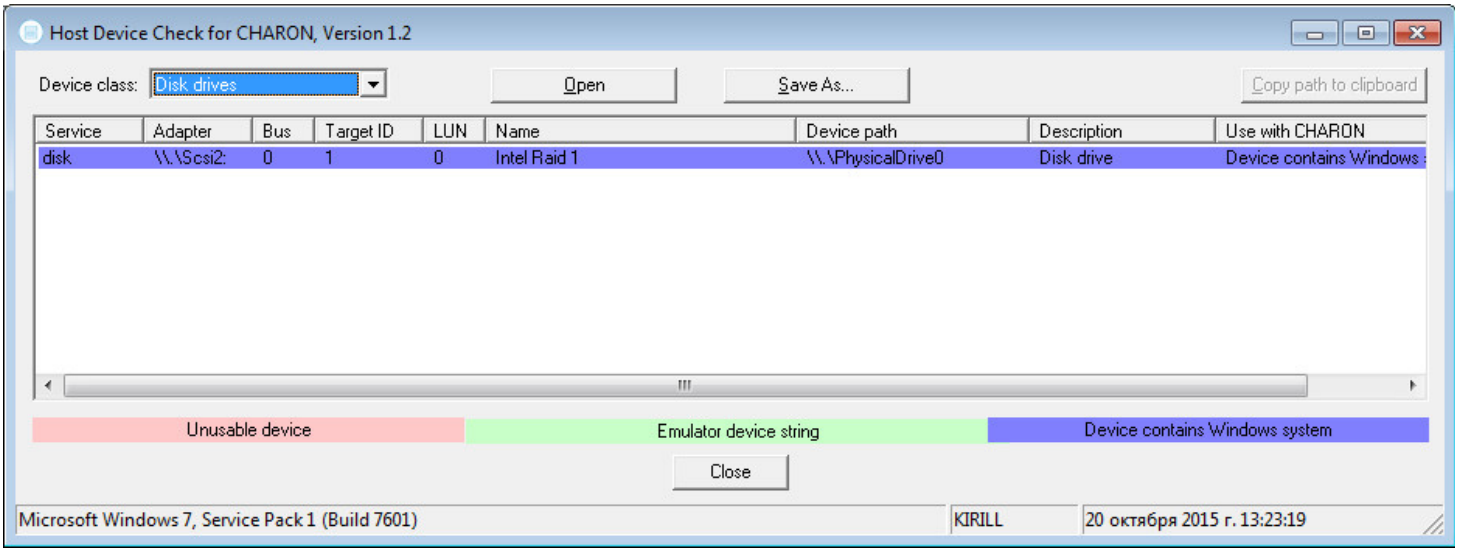
To start the utility click on the following icon under the "Start" menu:



Back to Table of Contents

Reviewing available physical disks

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



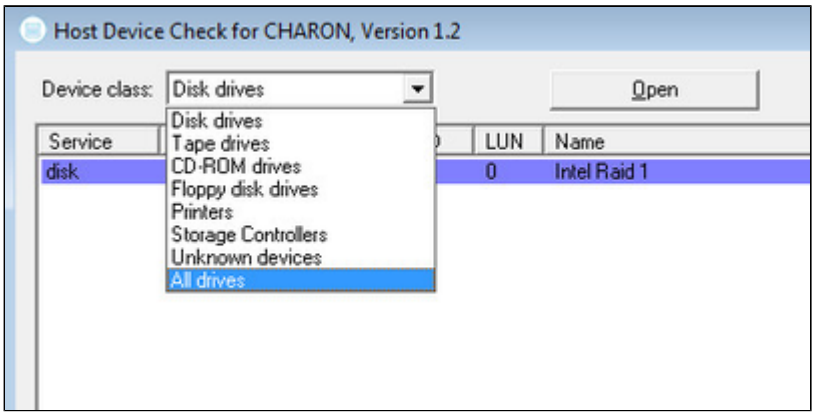
Please note that the "Unusable device" (marked with red) and "Device contains Windows system" must not be mapped to CHARON!

Back to Table of Contents

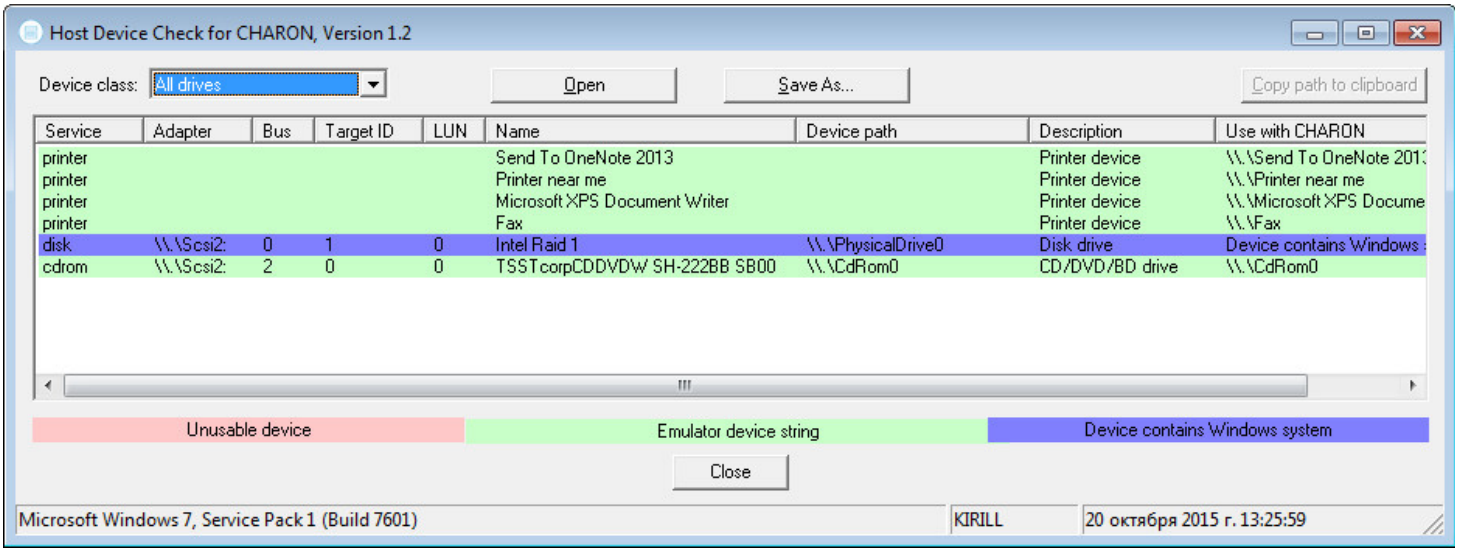
© Stromasys, 2017

Reviewing all the available host resources

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



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



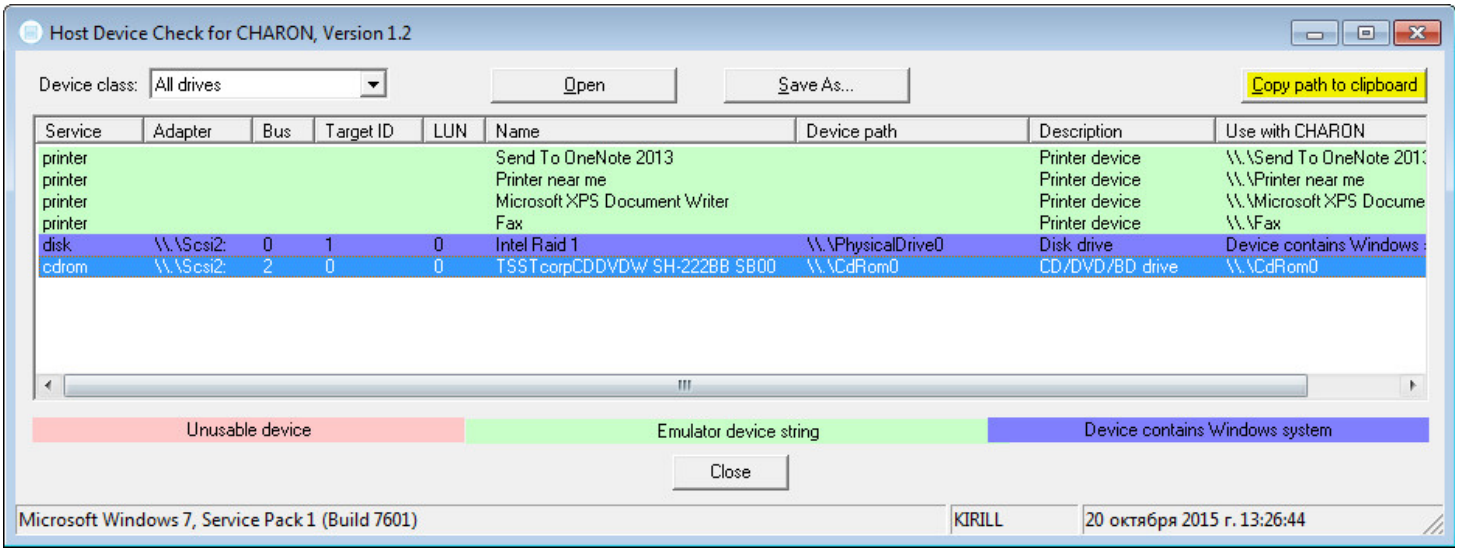
It is also possible to choose all the other categories to narrow the possible mapping options list.

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

Back to Table of Contents

Collecting the configuration strings

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



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

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

Back to Table of Contents

© Stromasys, 2017

Network Control Center

Table of Contents

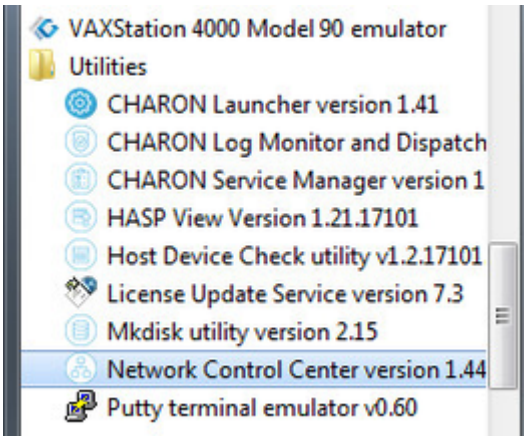
- General Description
- Installation of CHARON network drivers
- Deinstallation of the CHARON network driver
- Configuring the host network interfaces for CHARON
- Release of the host network interfaces
- Troubleshooting the CHARON network interfaces configuration
- Disable "offload" parameters
    - For Windows 2008 R2, Windows 2012 R2, Windows 7, 8 and 8.1
    - For older versions of Windows
- Monitoring the CHARON network activity

Back to Table of Contents

General Description

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

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



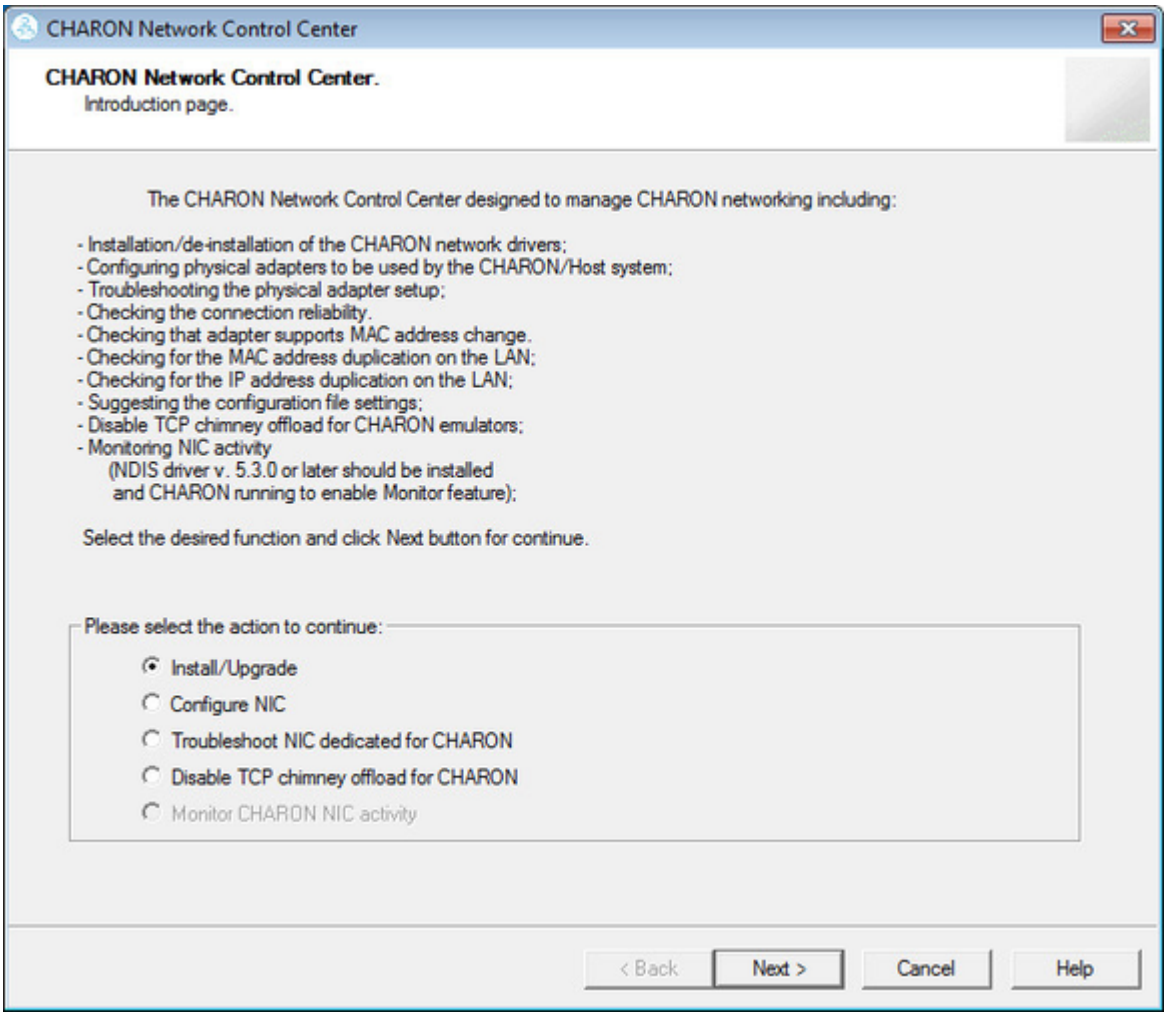
Back to Table of Contents

© Stromasys, 2017

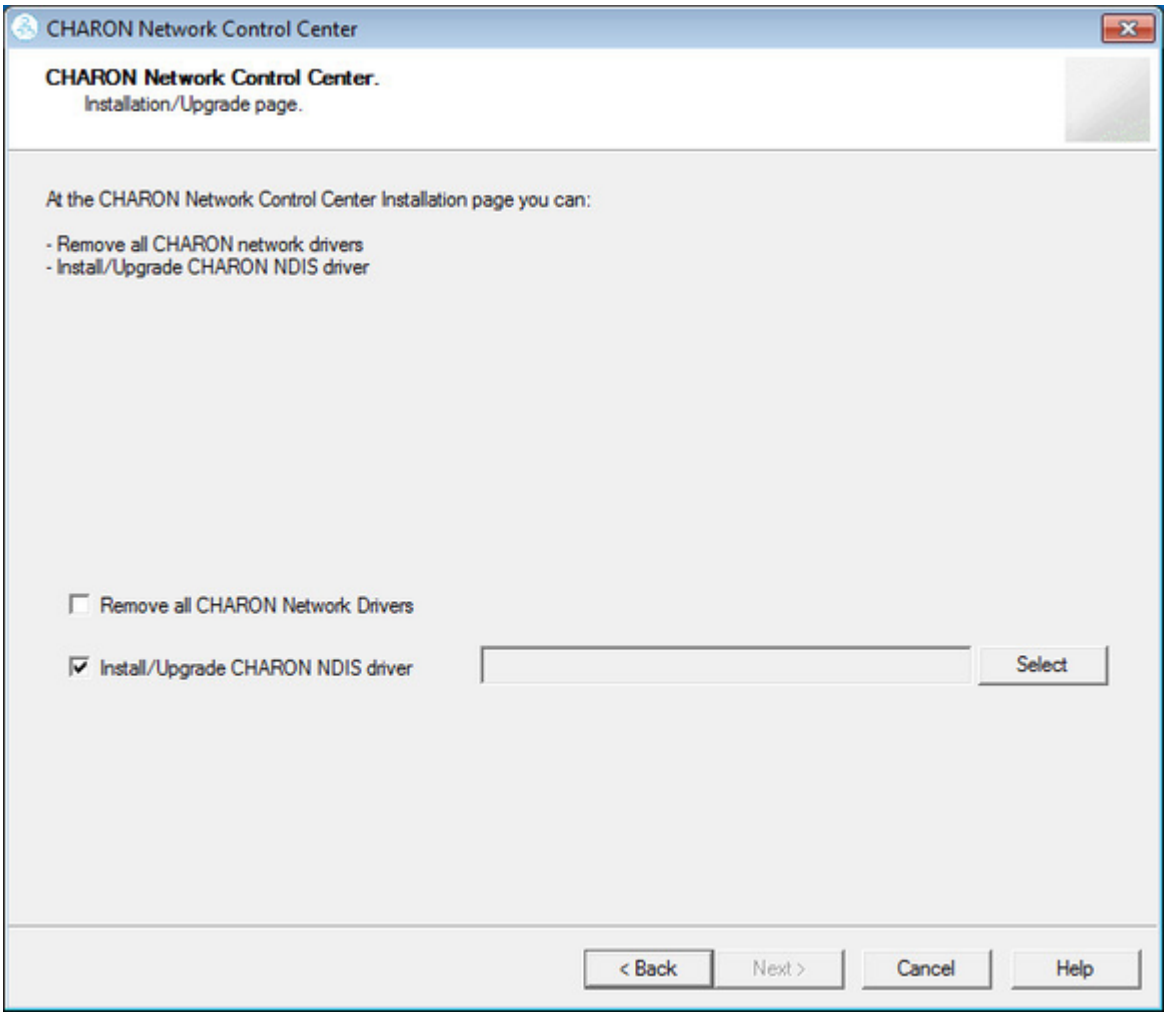
Installation of CHARON network drivers

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

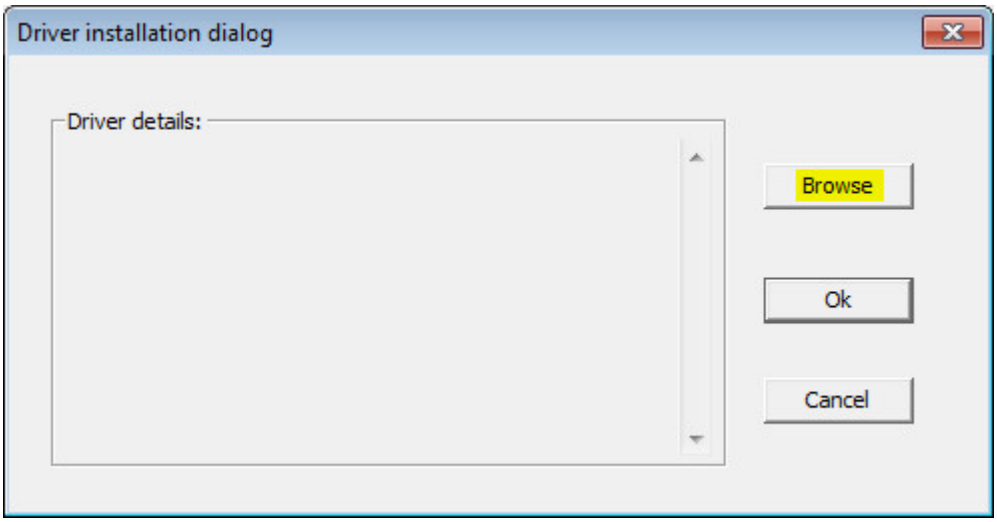
When it is needed to install a modified driver, use the following procedure.Select"Install/Upgrade" and press "Next":



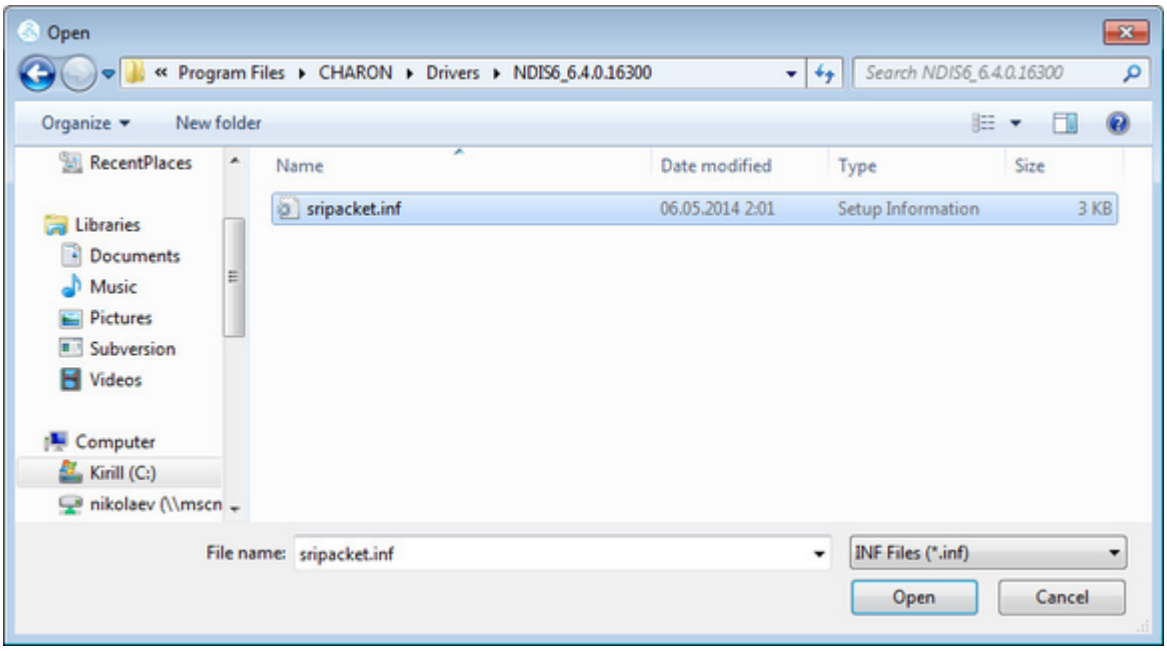
Select "Install/Upgrade CHARON NDIS driver" checkbox and press "Select":



Press "Browse":

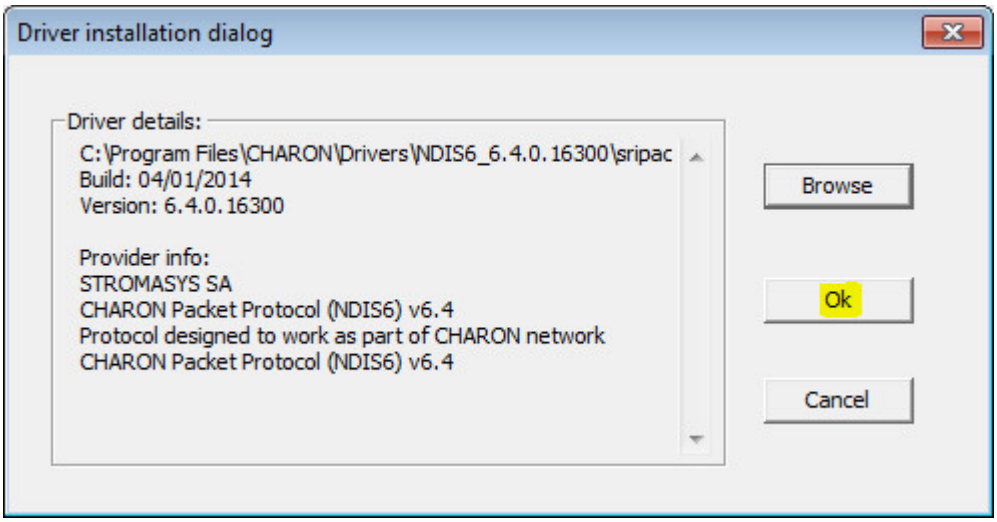


Browse for the target "sripacket.inf" file (by default it is located in the "C:\Program Files\CHARON\Drivers\NDIS6_X.X.X.XXXXX" directory), select it and press "Open":



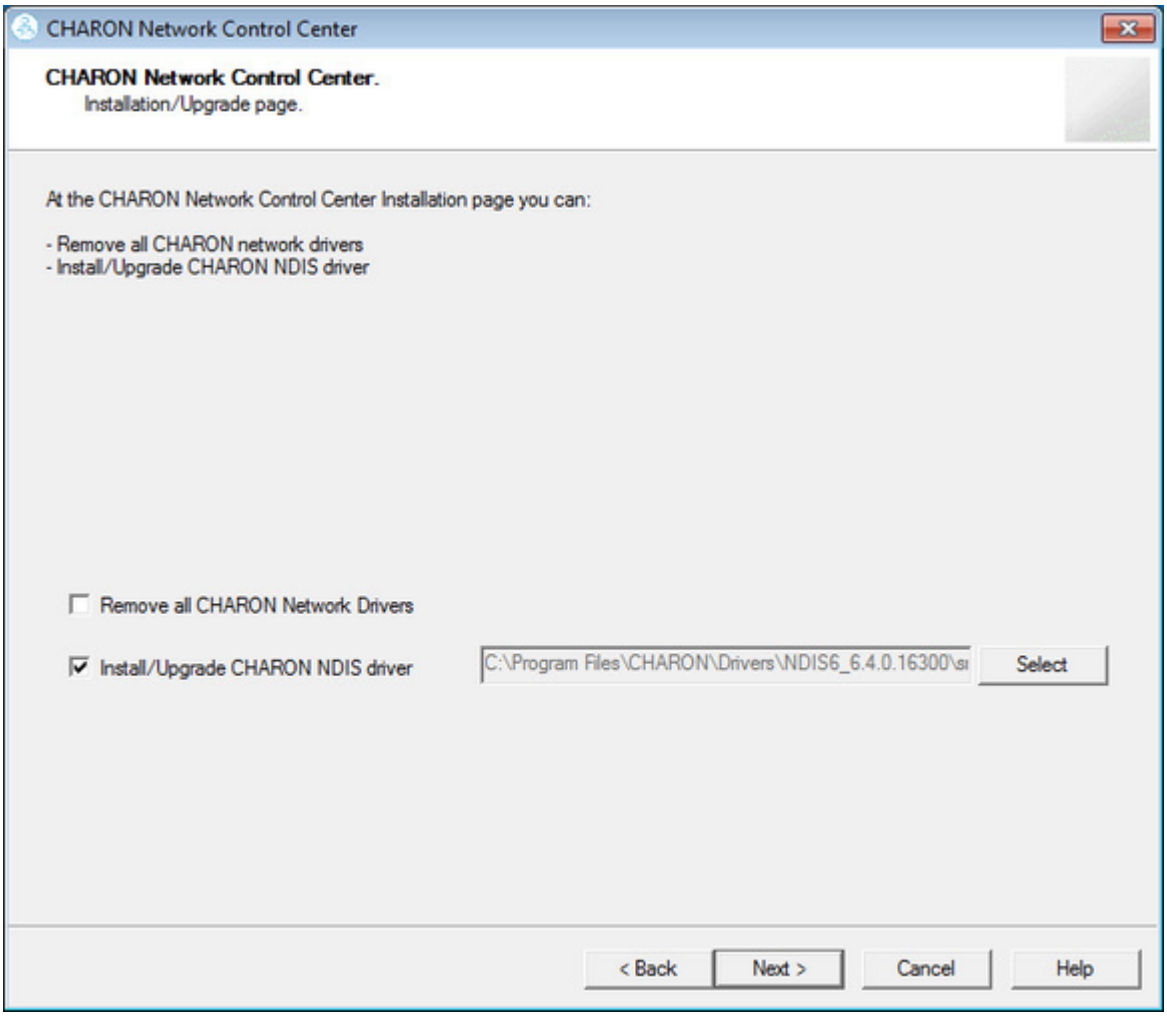
If the CHARON network driver has been acquired directly from STROMASYS, put it in a temporary directory and choose the "sripacket.inf" file from this directory in the dialog above.

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

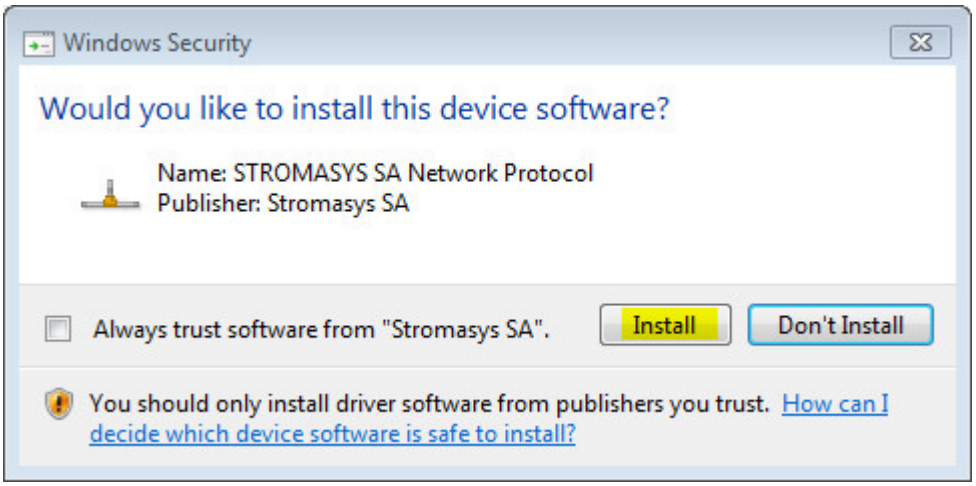


© Stromasys, 2017

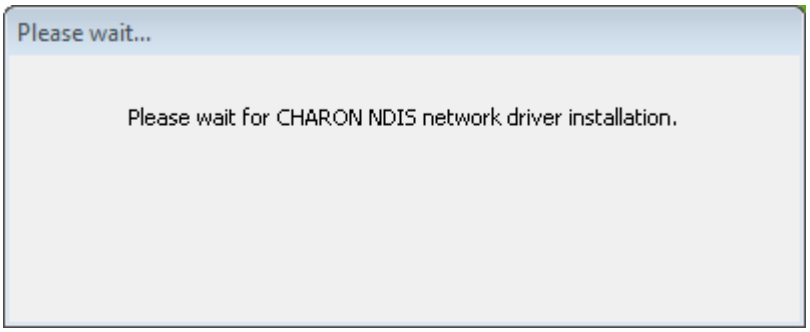
The Network Control Center will display the previous dialog with the path to the CHARON network driver. Press "Next" in the popup below to continue:



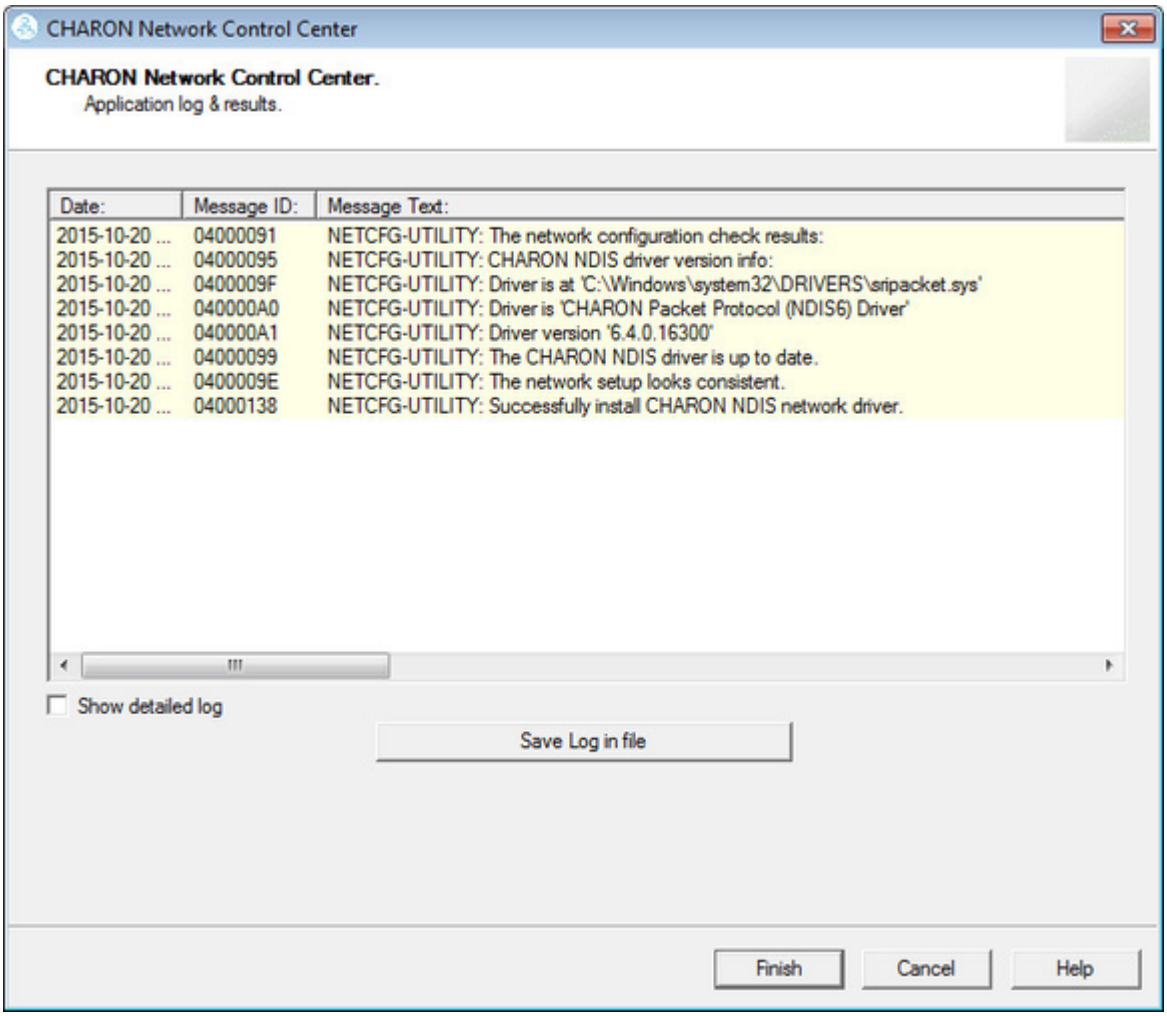
The utility will begin the CHARON network driver installation. If Windows Security asks you to confirm the driver installation, press "Install":



The driver installation will resume:



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



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

Select "Show detailed log" to display more detail.

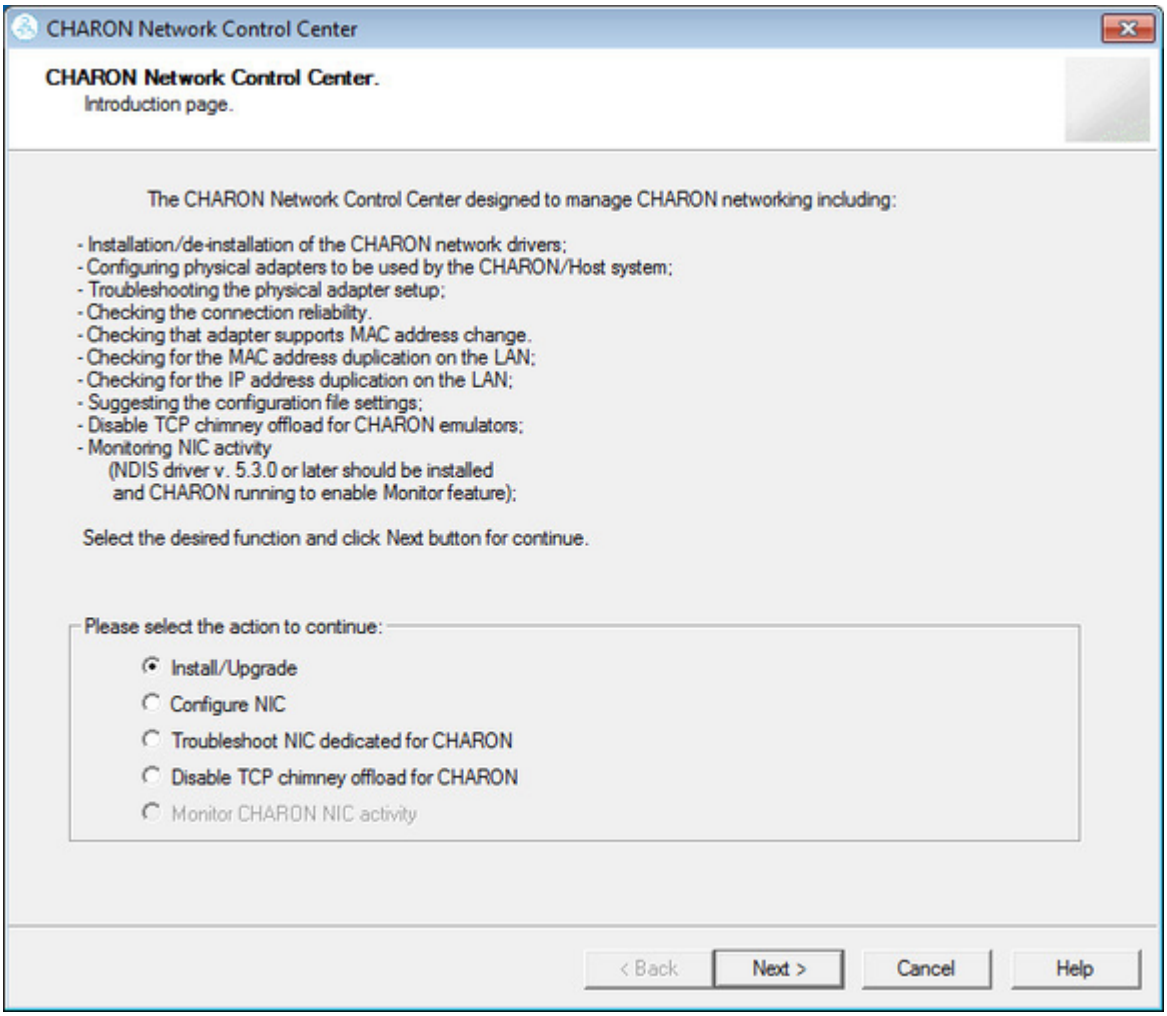
Press "Finish" to exit.

Back to Table of Contents

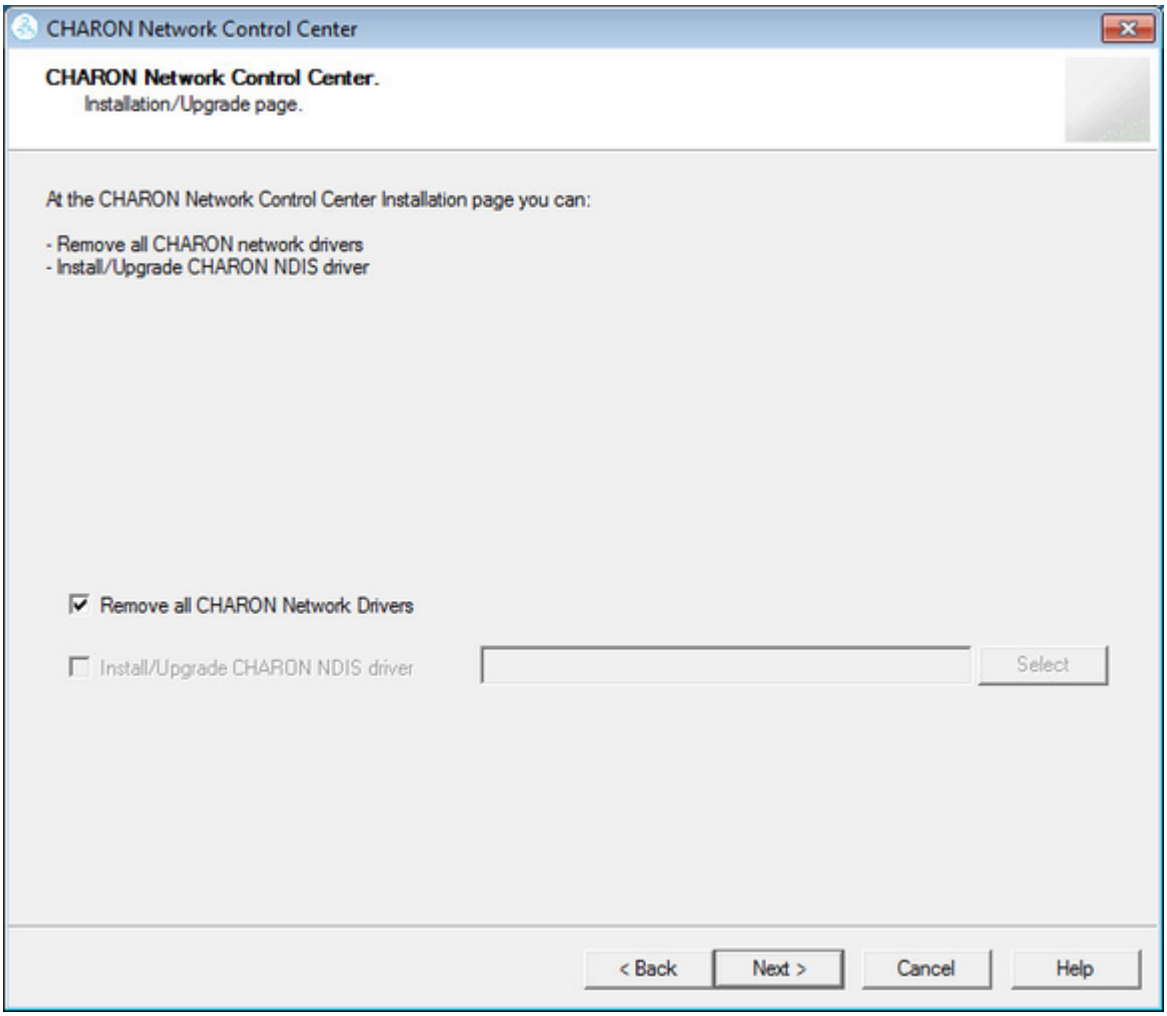
© Stromasys, 2017

Deinstallation of the CHARON network driver

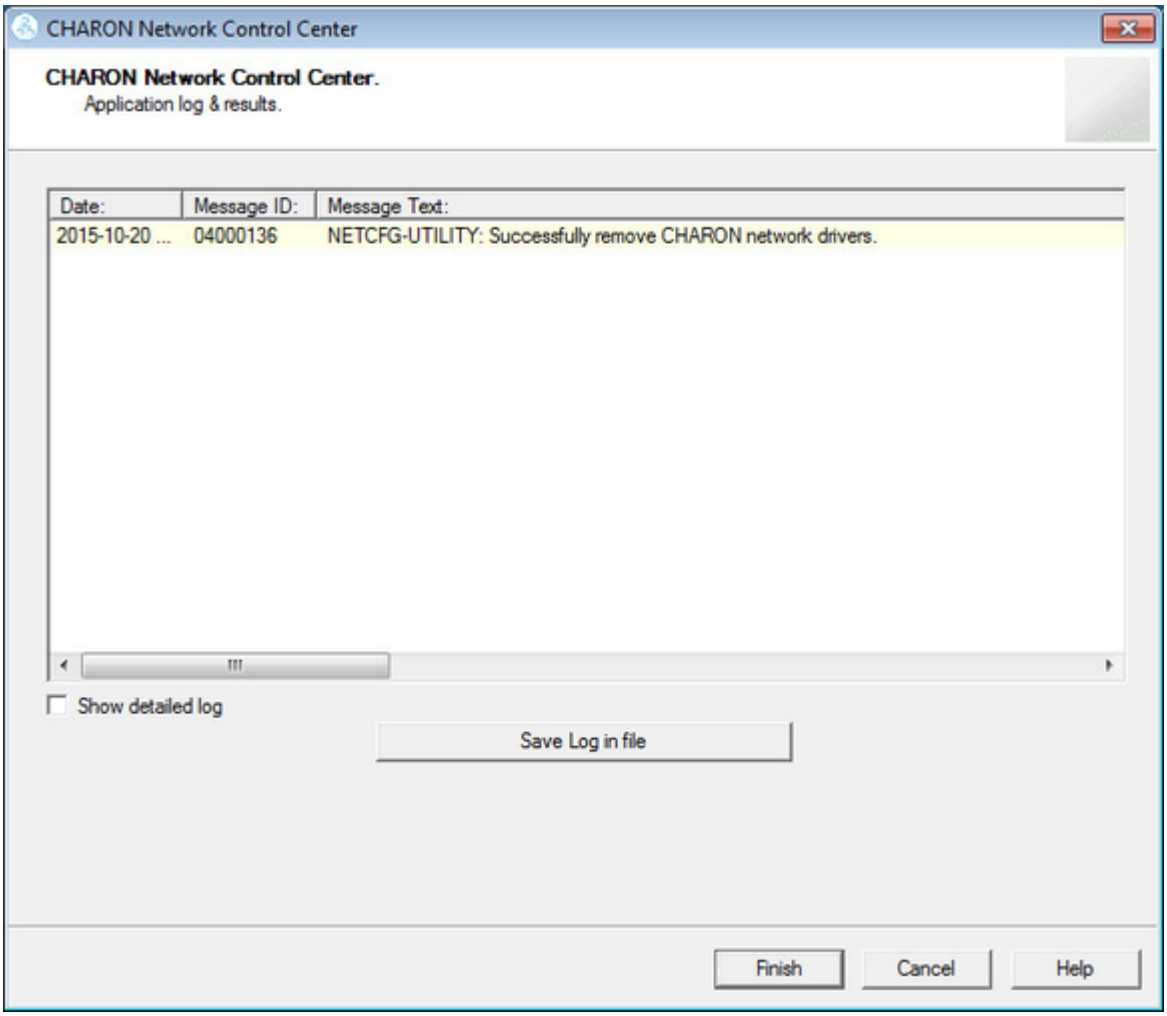
Start the utility and select "Install/Upgrade"; press "Next":



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

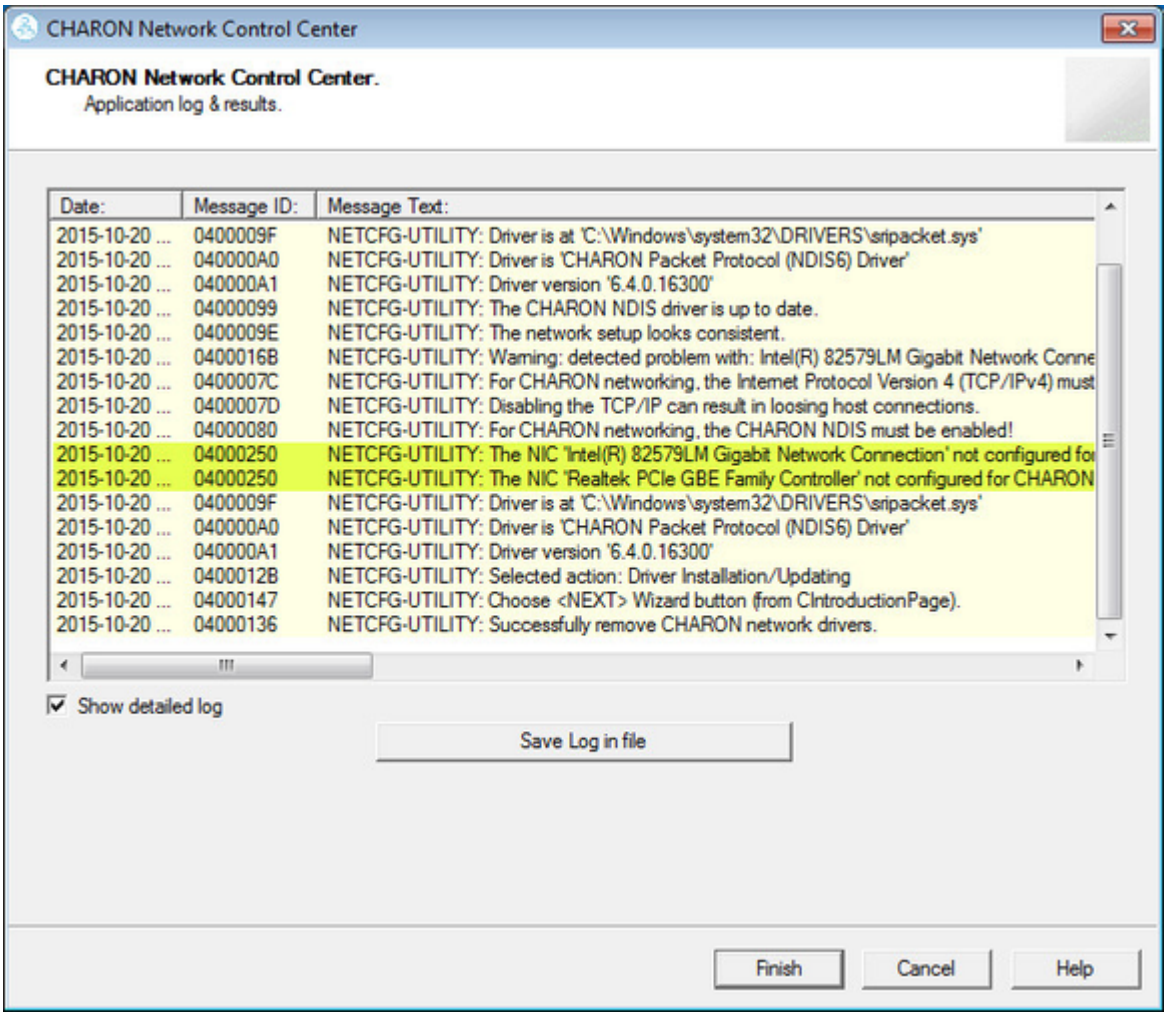


The utility will report that the CHARON drivers have been removed:



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

Select "Show detailed log" to display more detail.



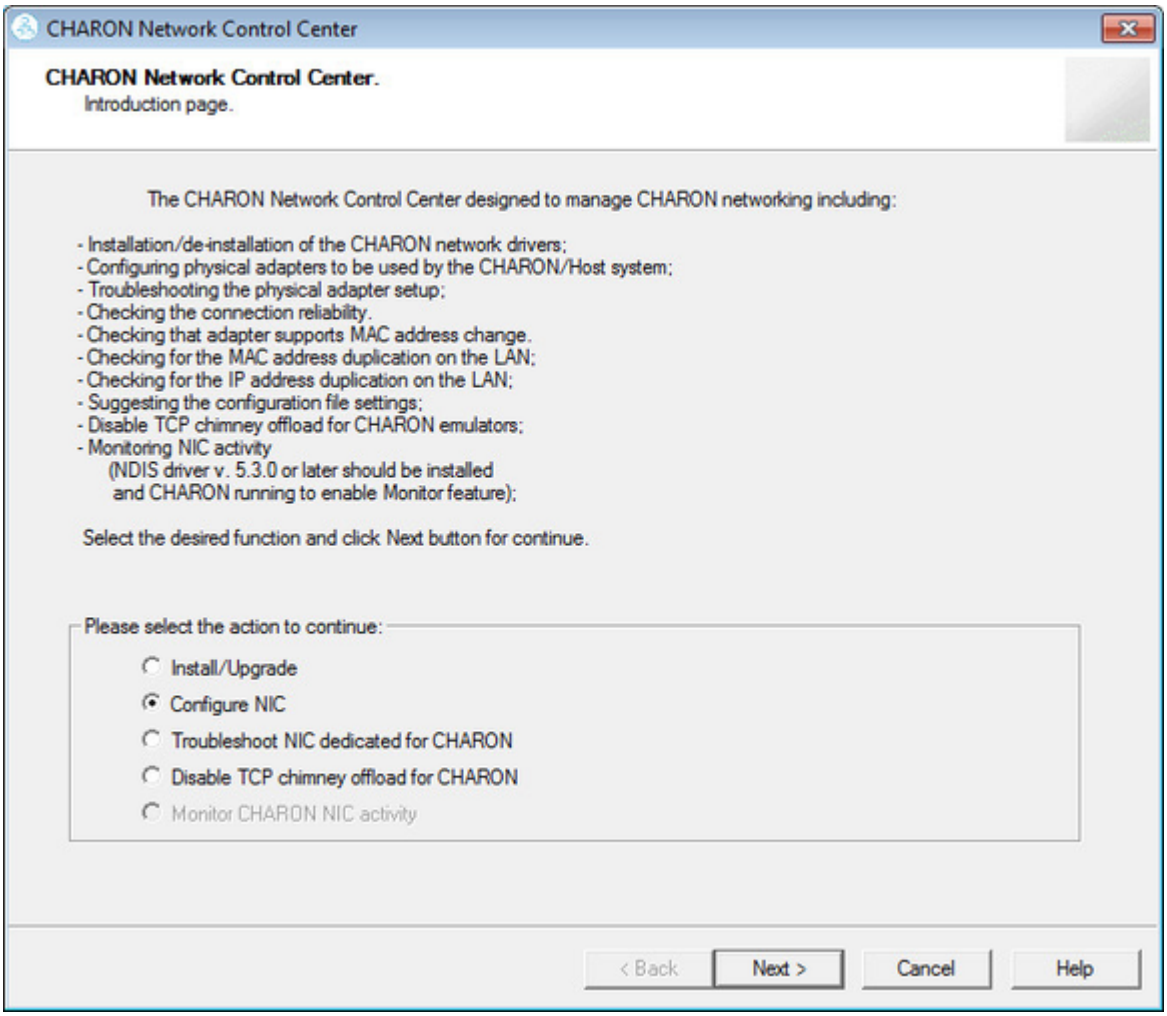
Press "Finish" to exit.

Back to Table of Contents

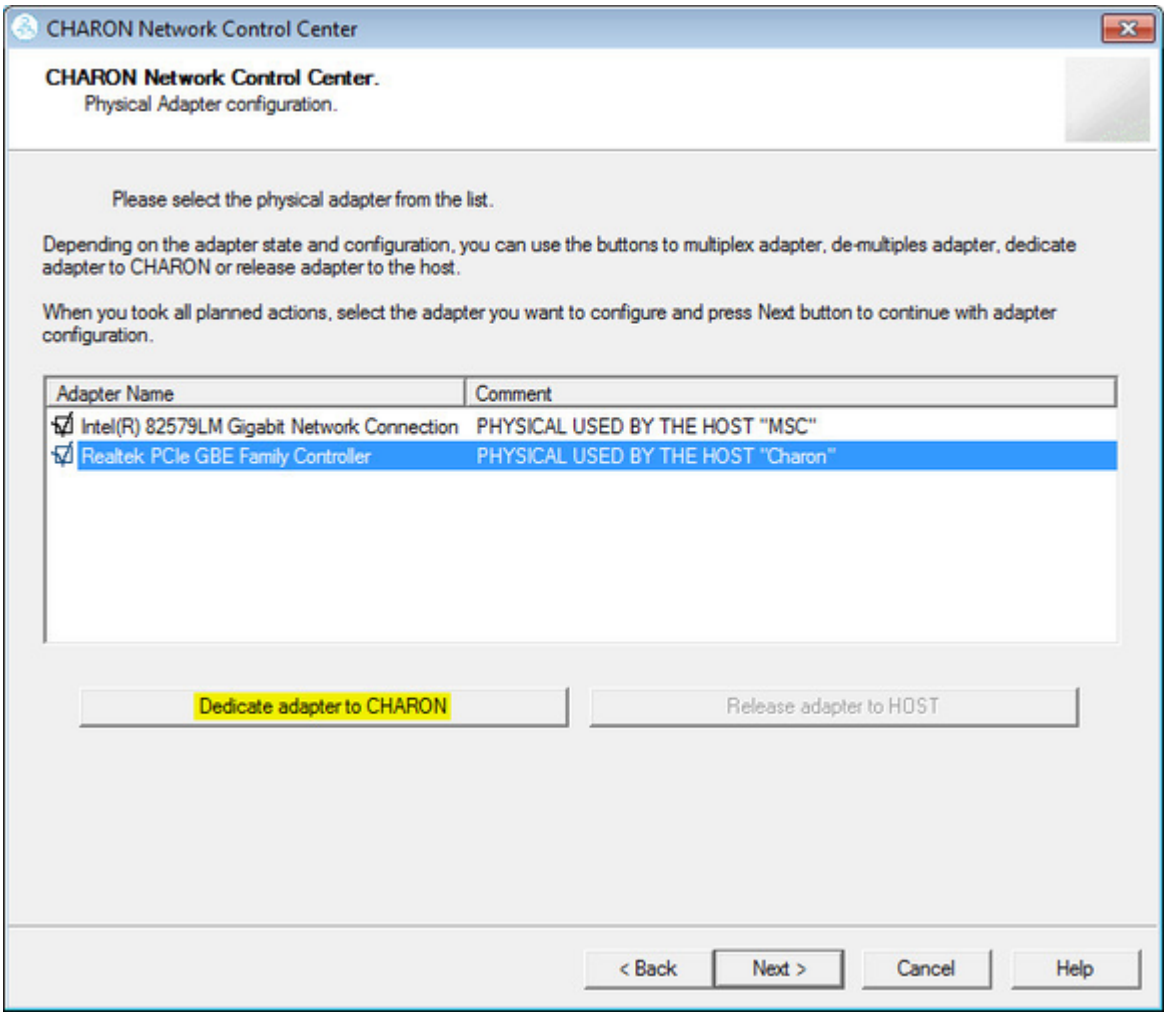
© Stromasys, 2017

Configuring the host network interfaces for CHARON

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



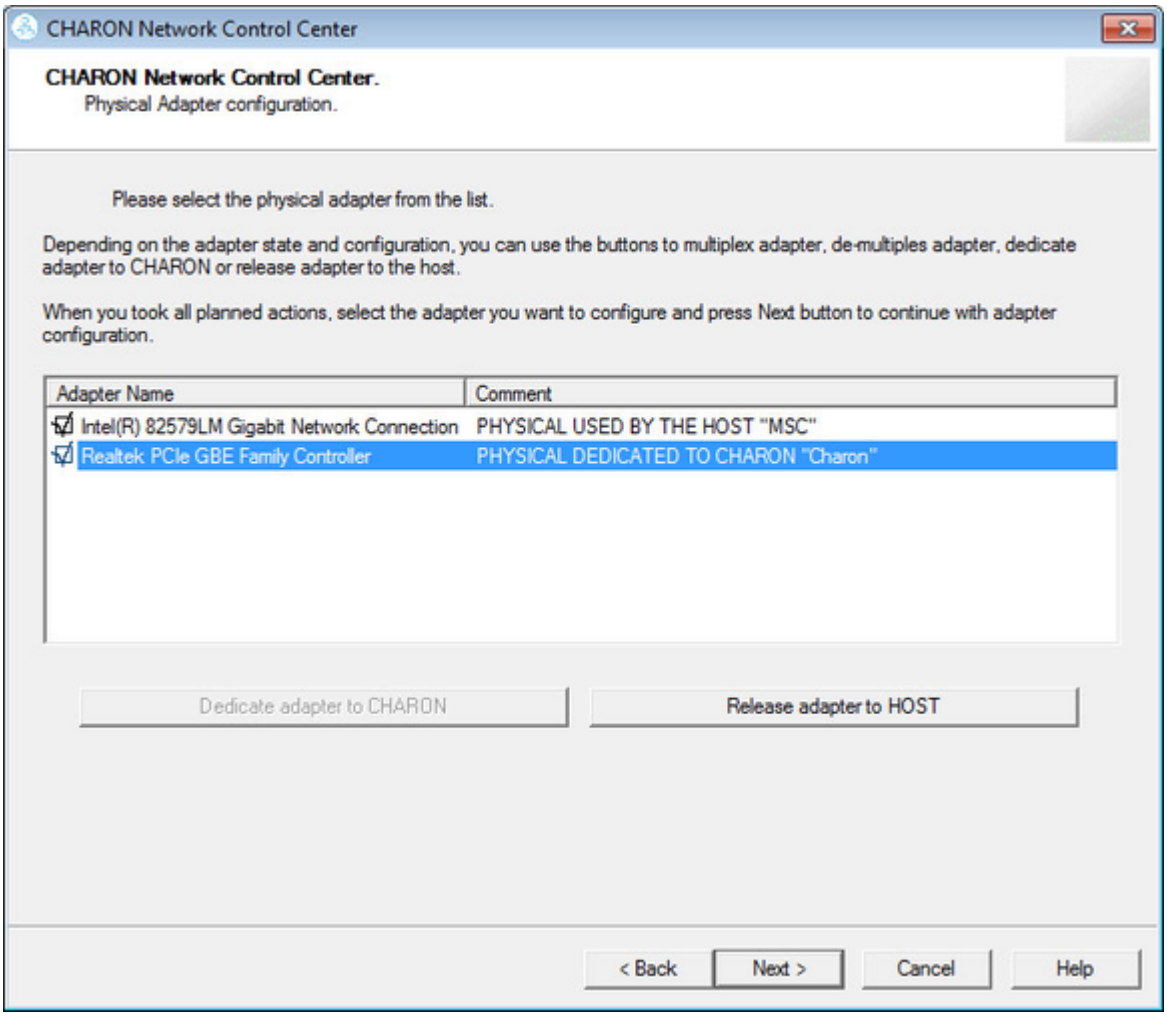
Select the host interface to be dedicated to CHARON (in the example below it is "Charon"), press "Dedicate adapter to CHARON":



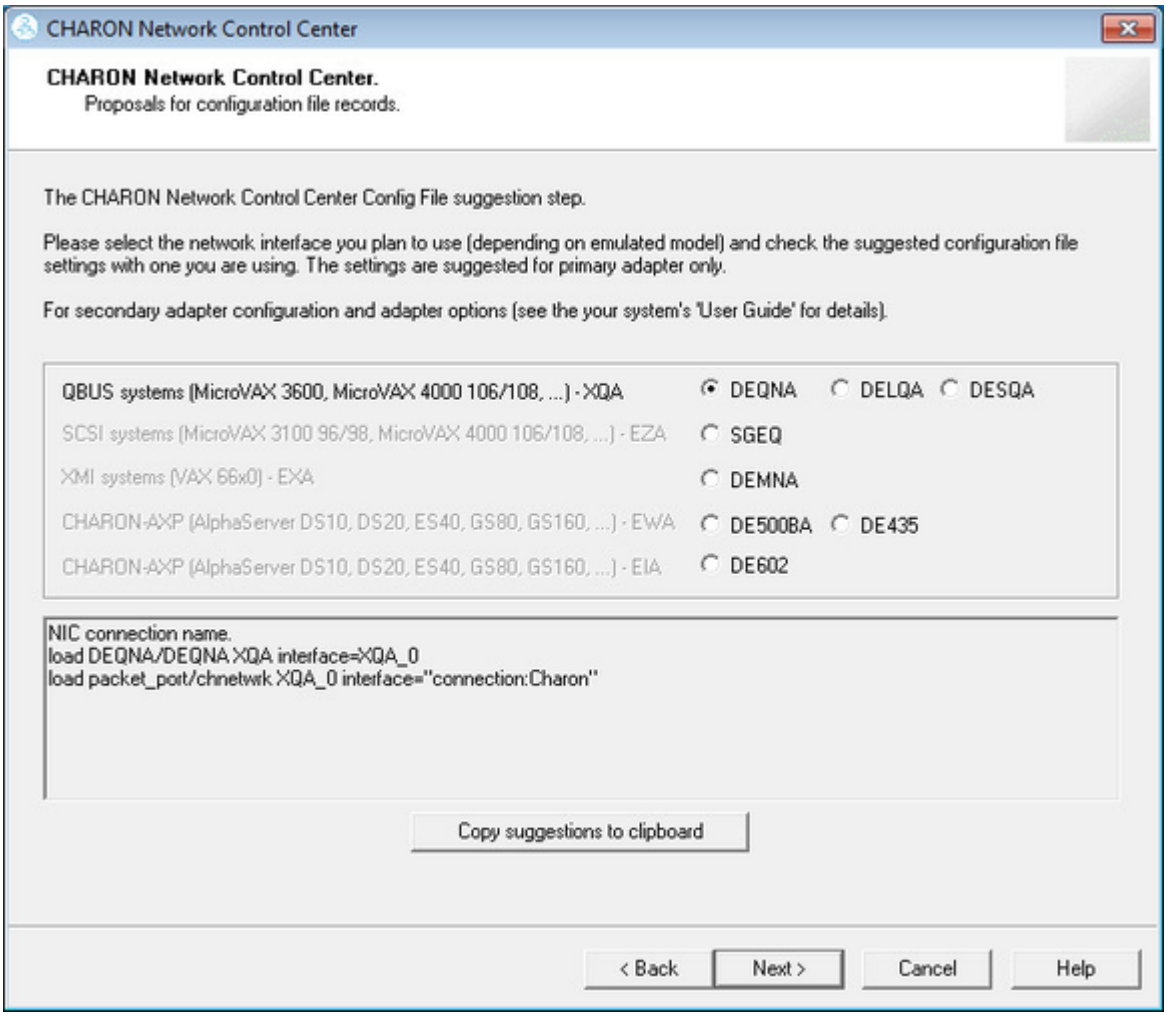
Install and configure VLAN adapters, according to the vendor's User's Guide, if required. Select the VLAN adapter in the dialog box in the above example. A VLAN adapter is not configured differently, the same procedure should be followed.

© Stromasys, 2017

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



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

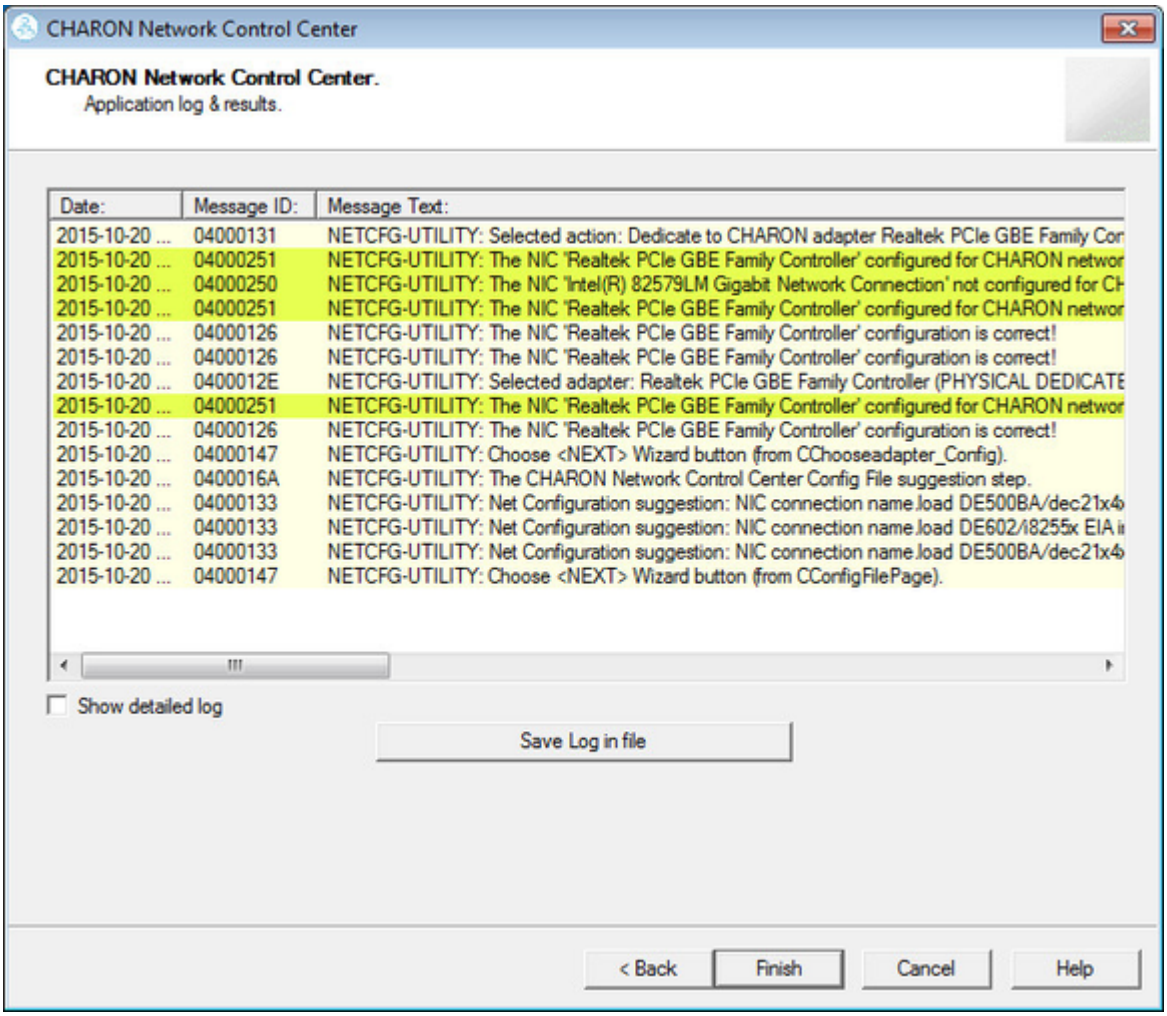


It is possible to paste the content of the clipboard to the CHARON configuration file as shown in the example below:

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

© Stromasys, 2017

Press "Next" to see the log file:



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

Select "Show detailed log" to display more detail.

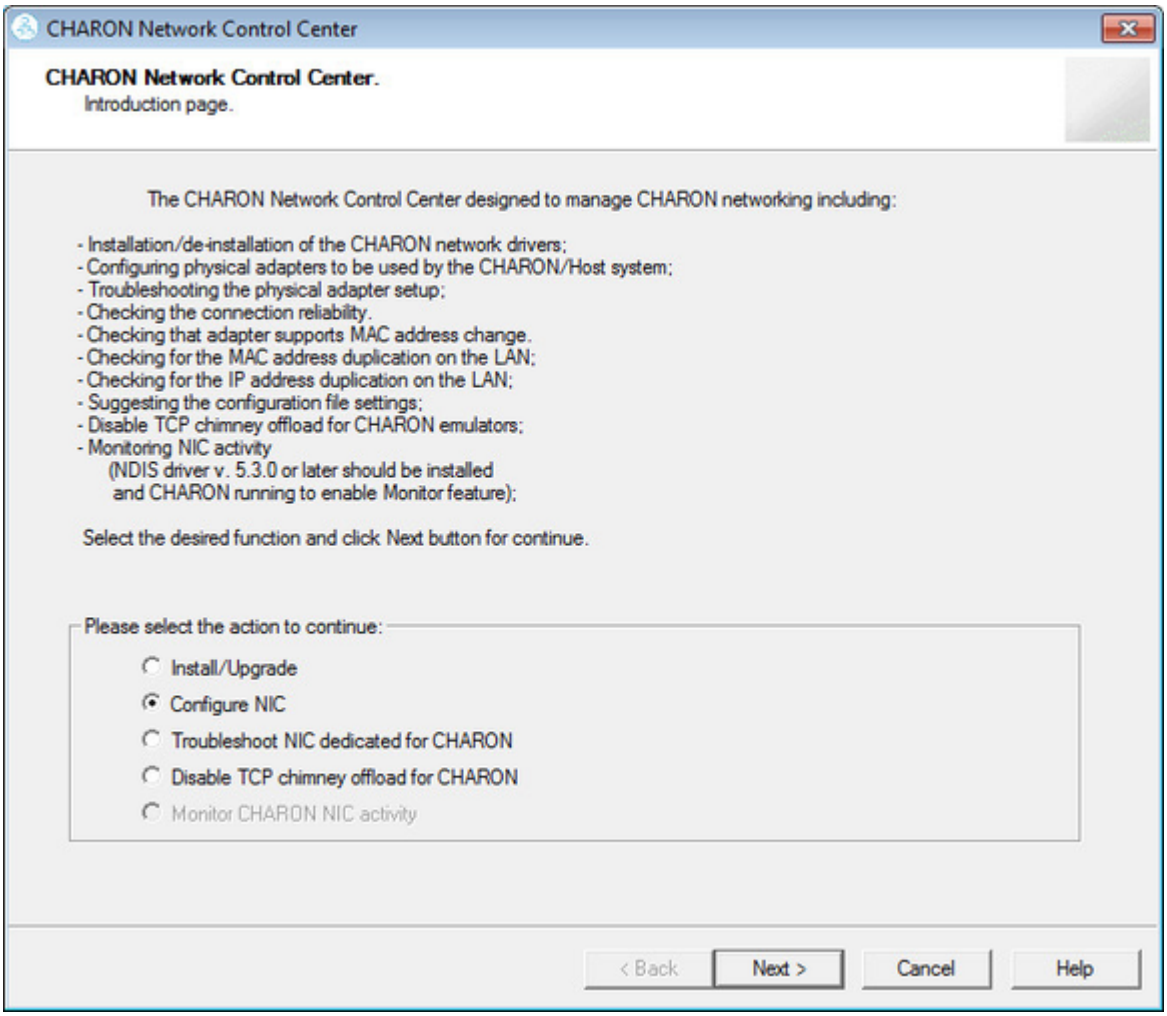
Press "Finish" to exit.

Back to Table of Contents

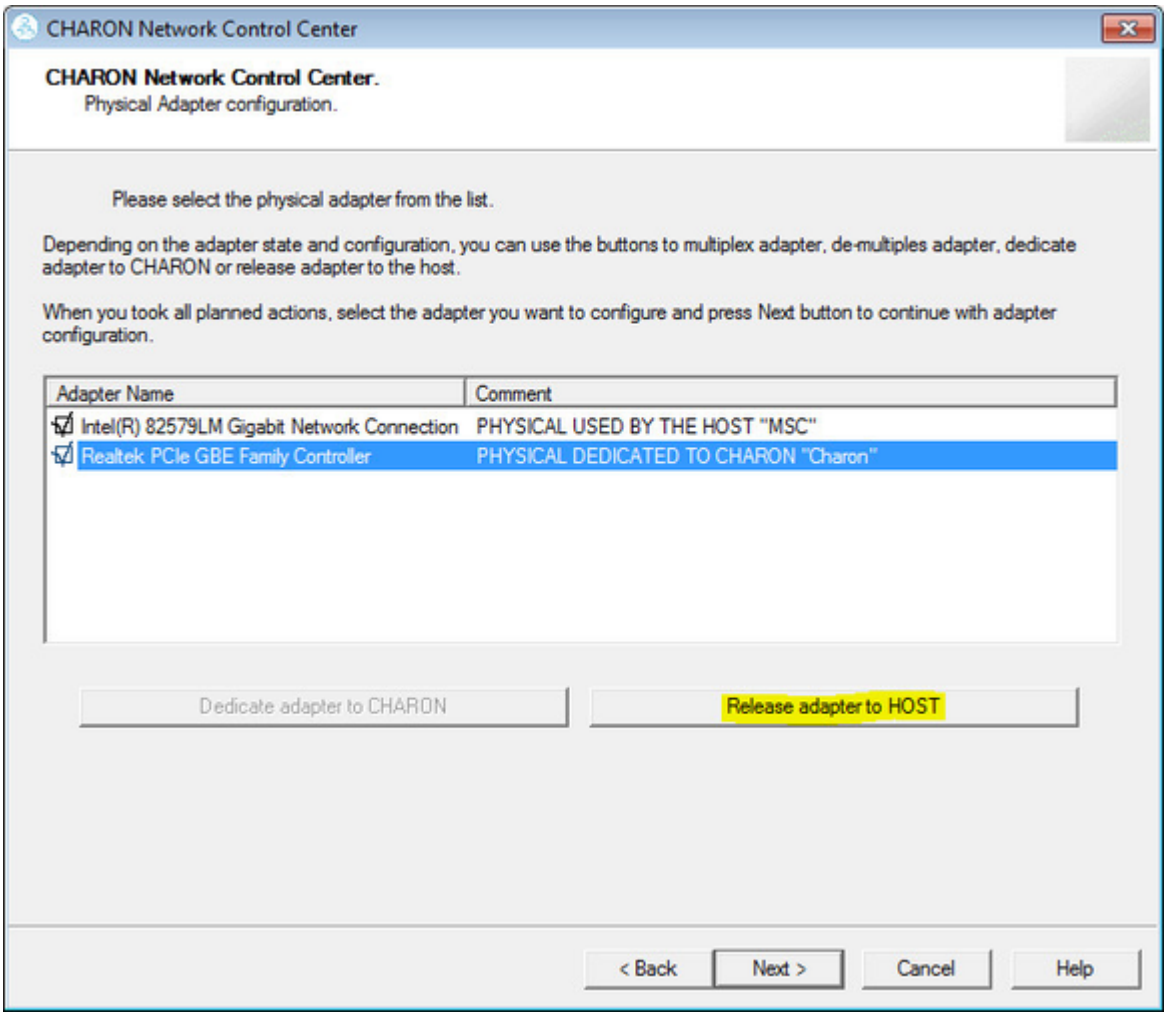
© Stromasys, 2017

Release of the host network interfaces

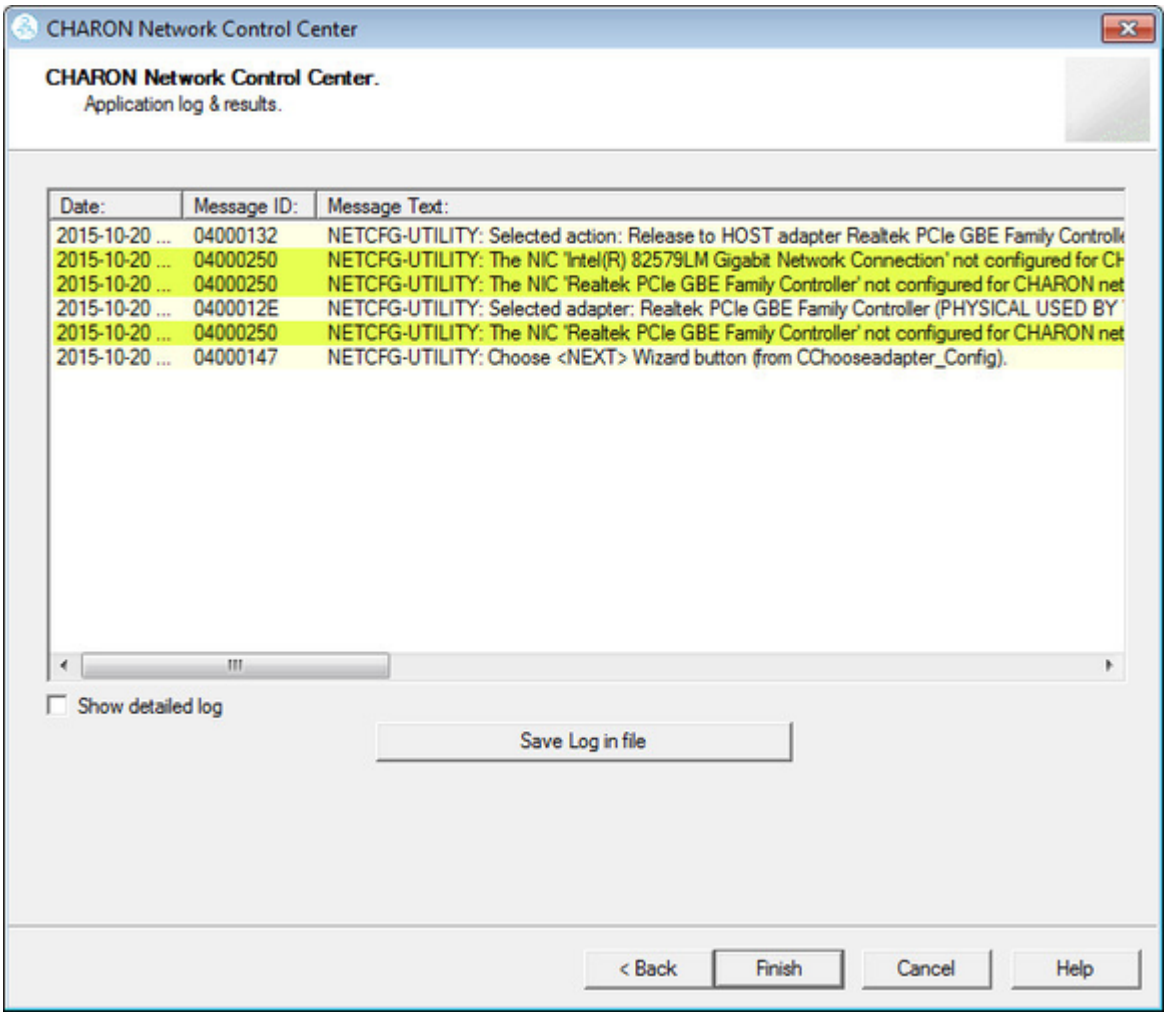
Start the utility and select "Configure NIC"; press "Next":



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



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



Select "Show detailed log" to display more detail.

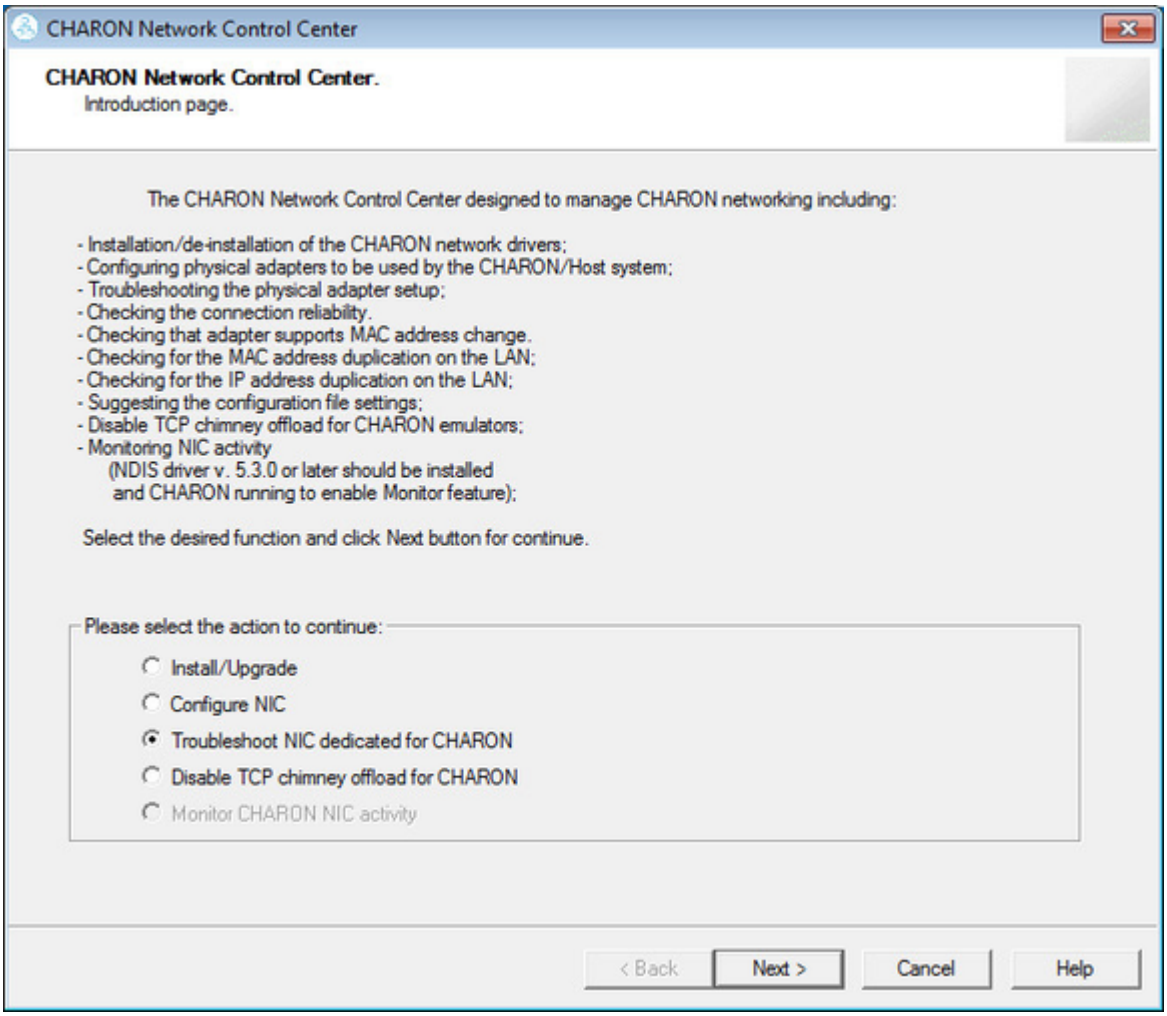
Press "Finish" to exit.

Back to Table of Contents

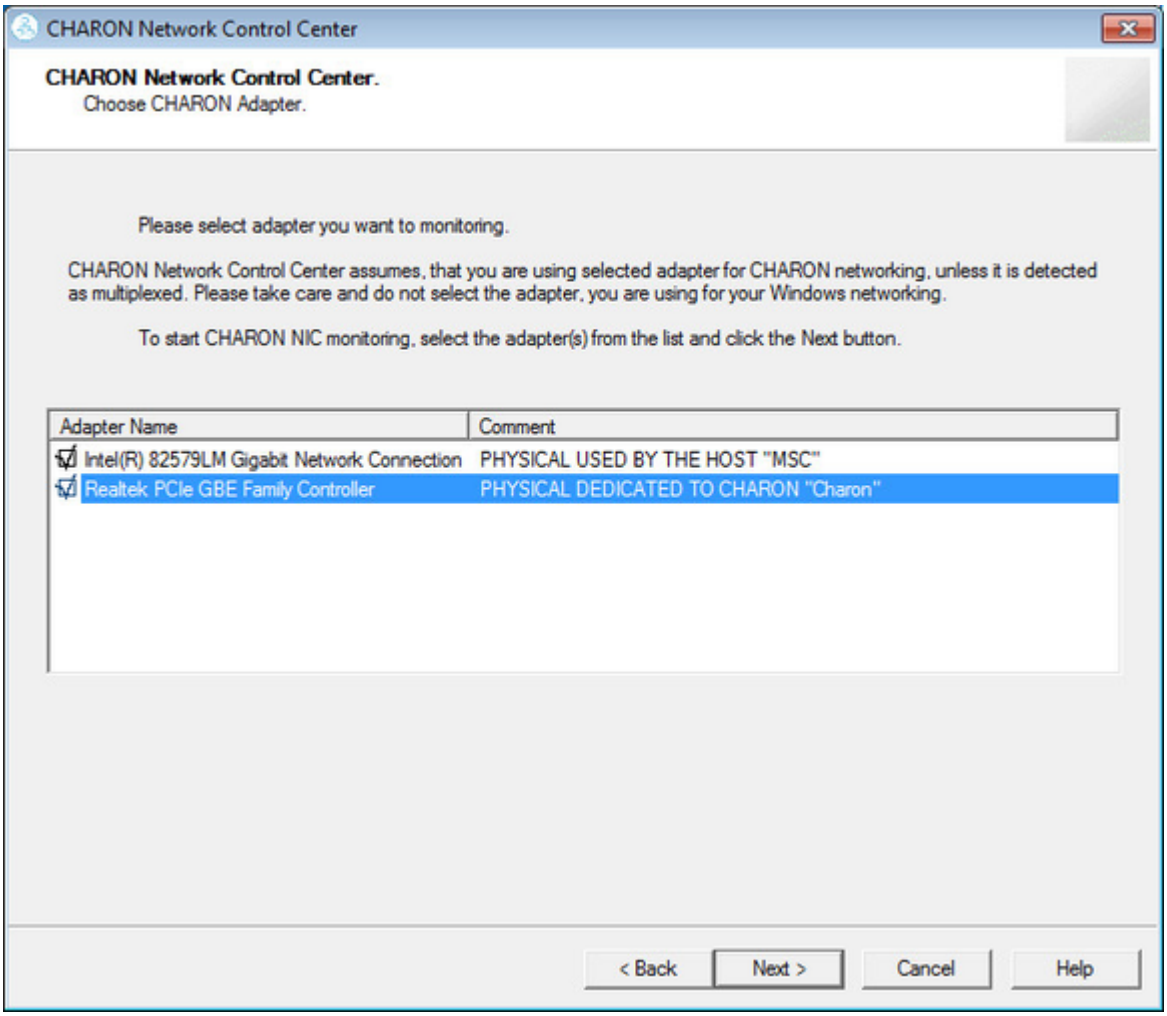
© Stromasys, 2017

Troubleshooting the CHARON network interfaces configuration

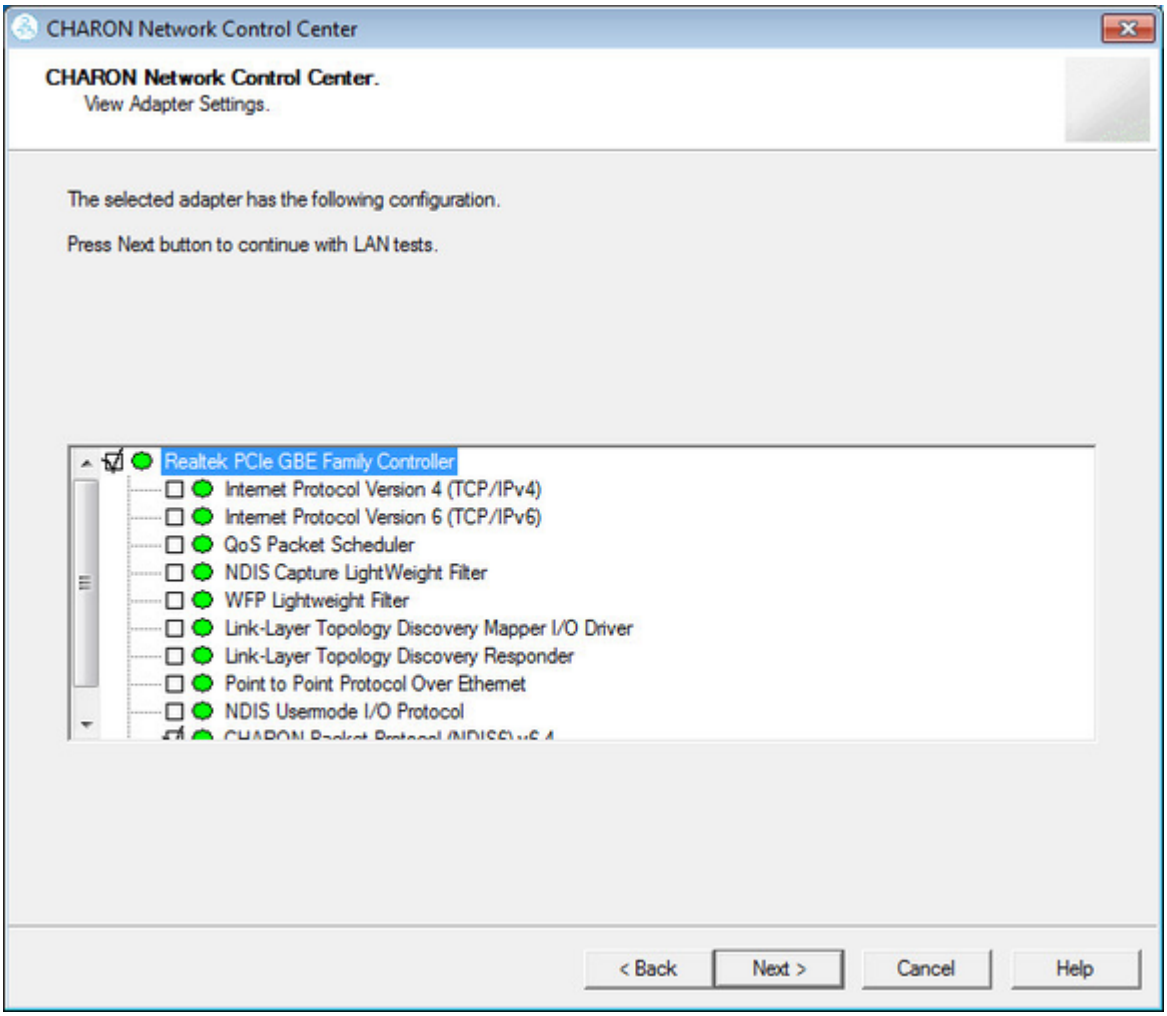
Start the utility and select "Troubleshoot NIC dedicated for CHARON"; press "Next":



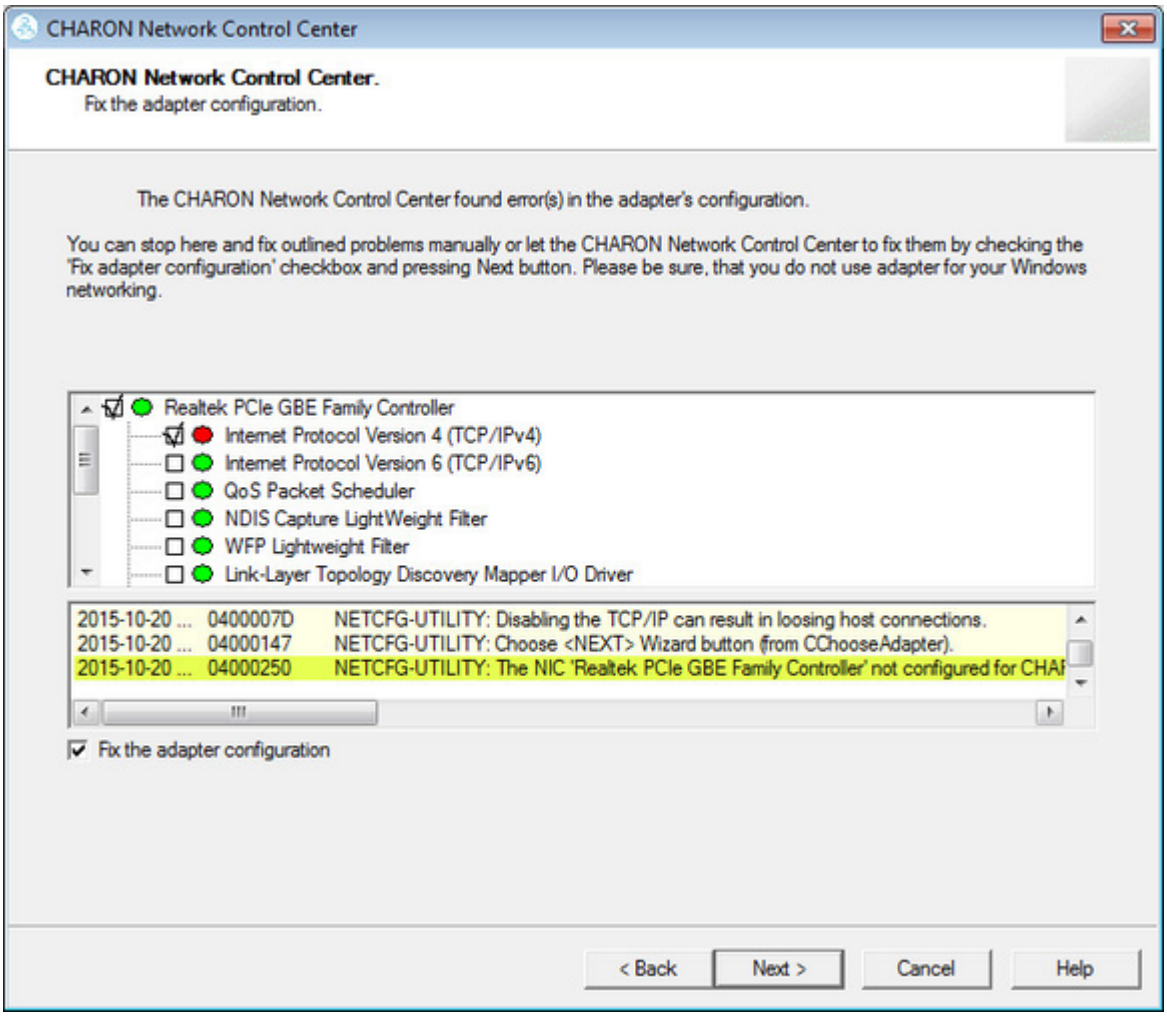
Select the target interface and press "Next":



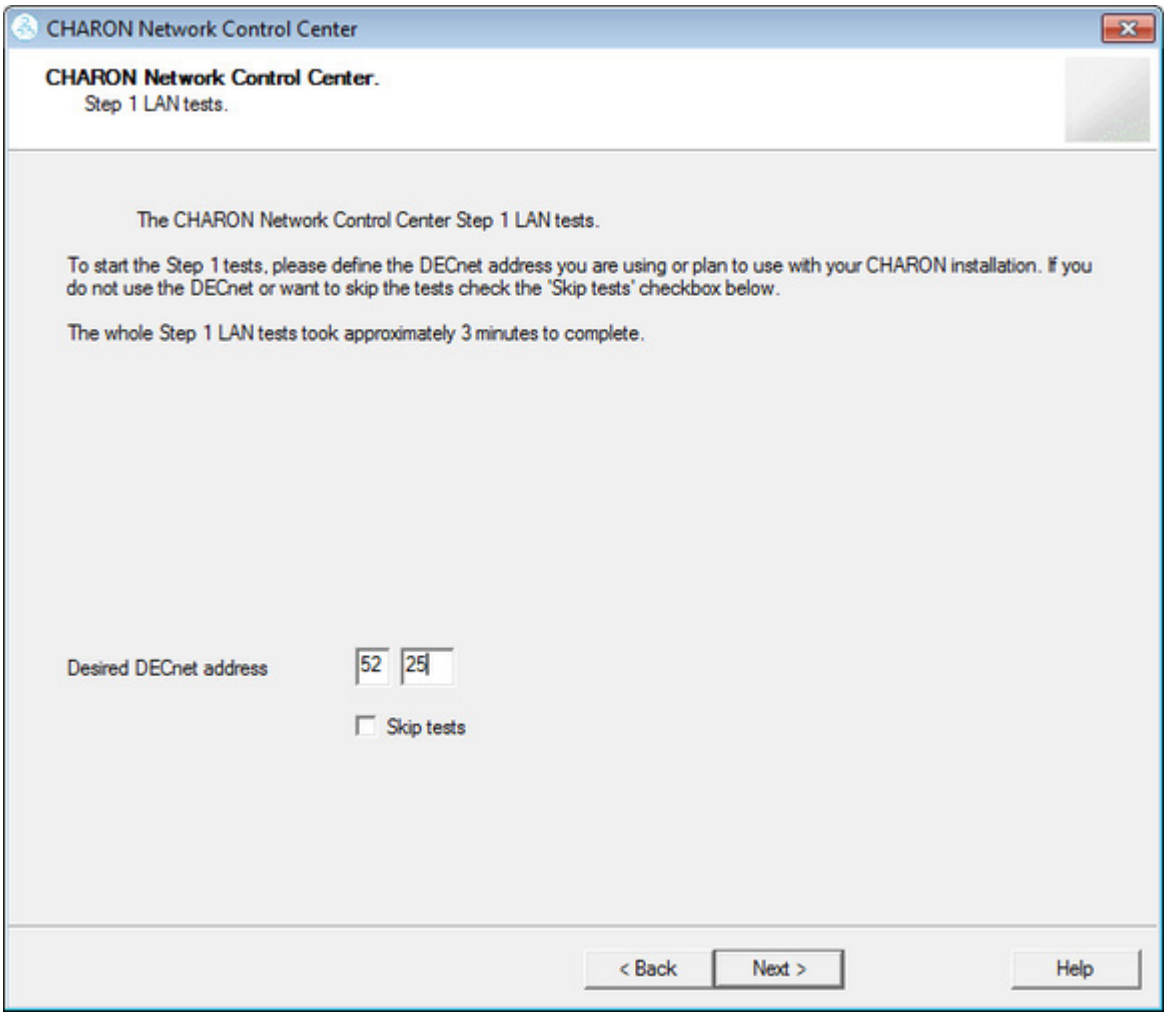
Review the status of the interface:



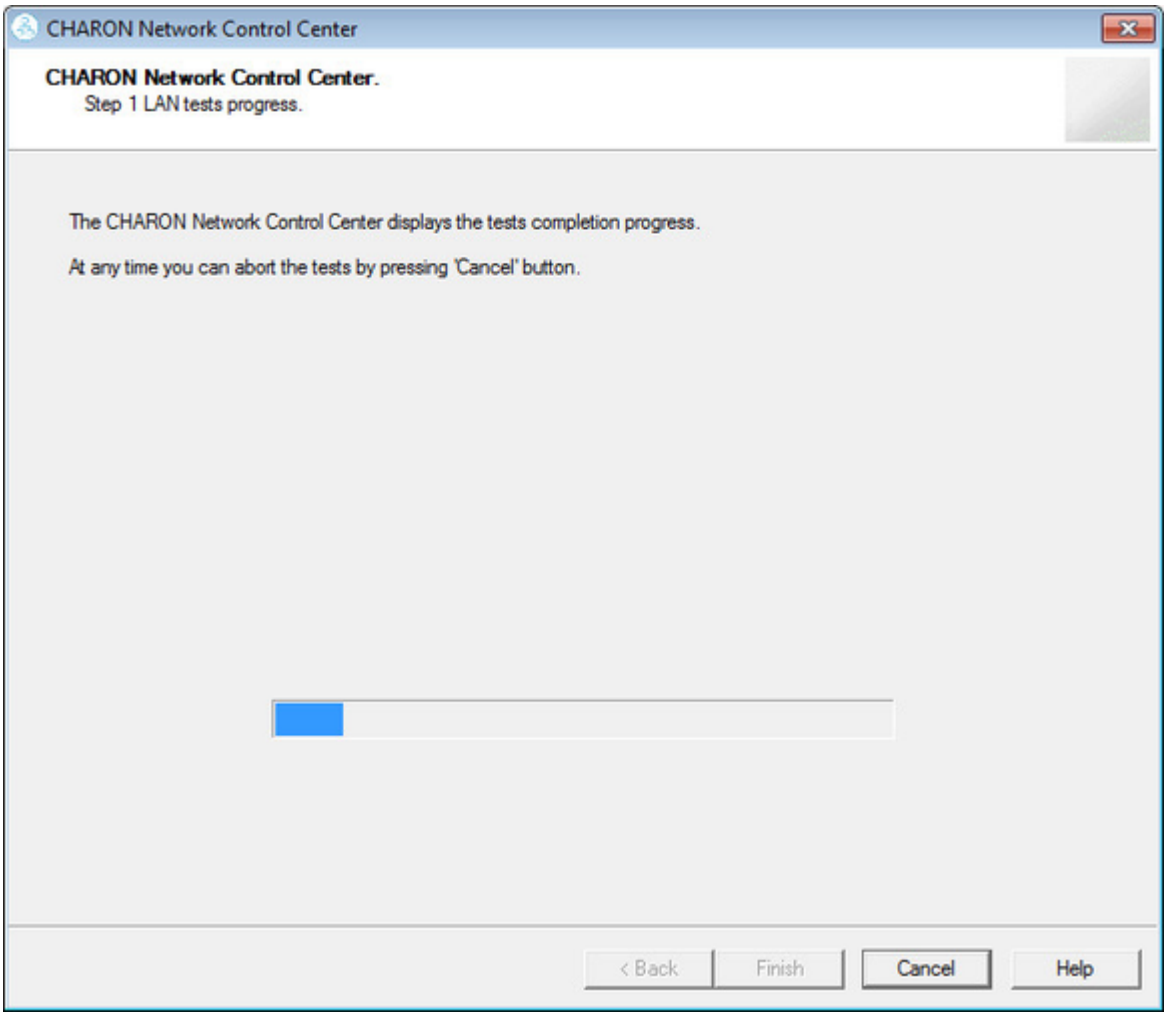
In the example below, there is an issue with the host TCP/IP being enabled on the CHARON interface. Review the problem description, press "Fix the adapter configuration" and press "Next".



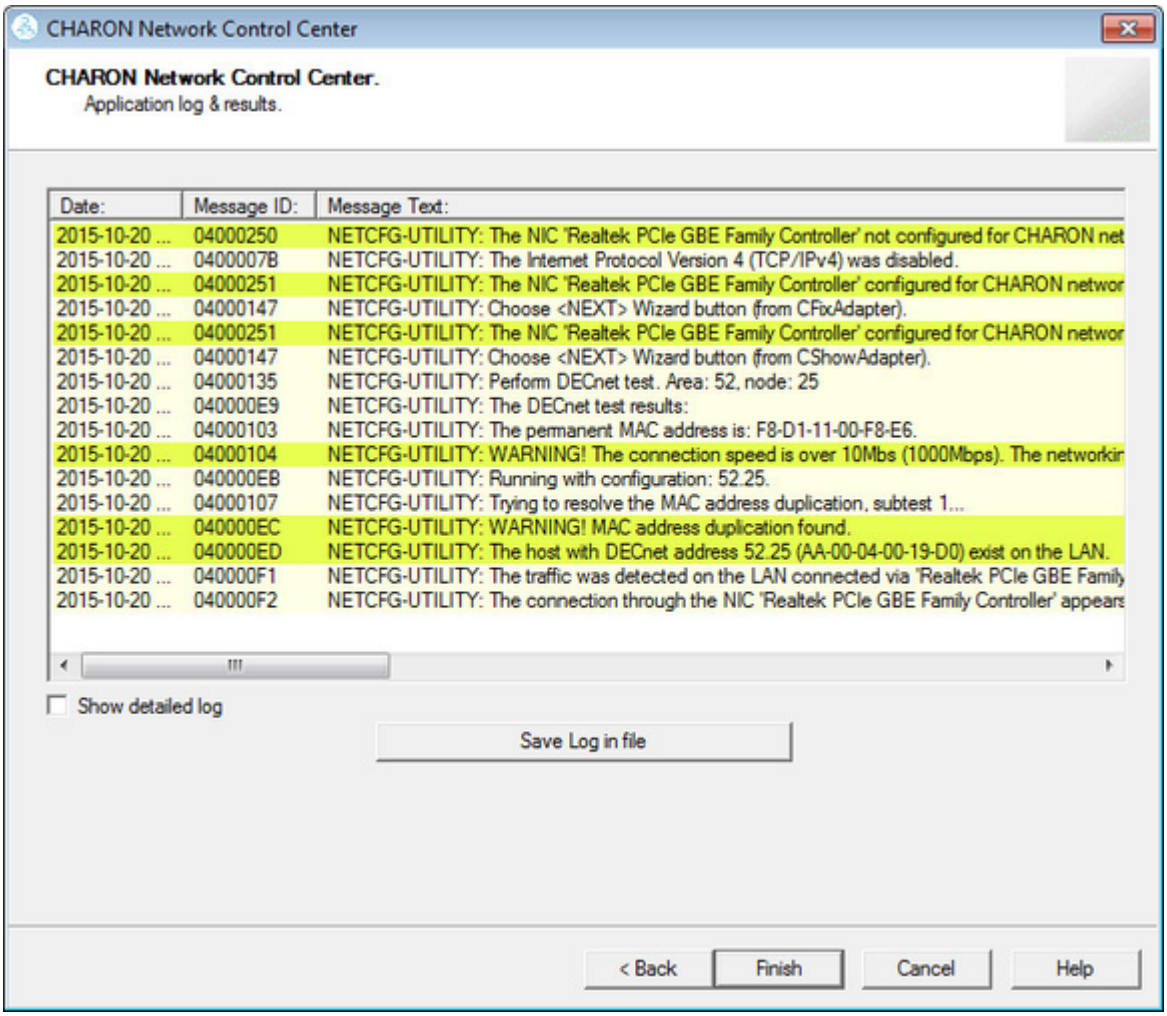
The utility will fix any issues and report a good status for the interface. Press "Next", the following dialog will appear:



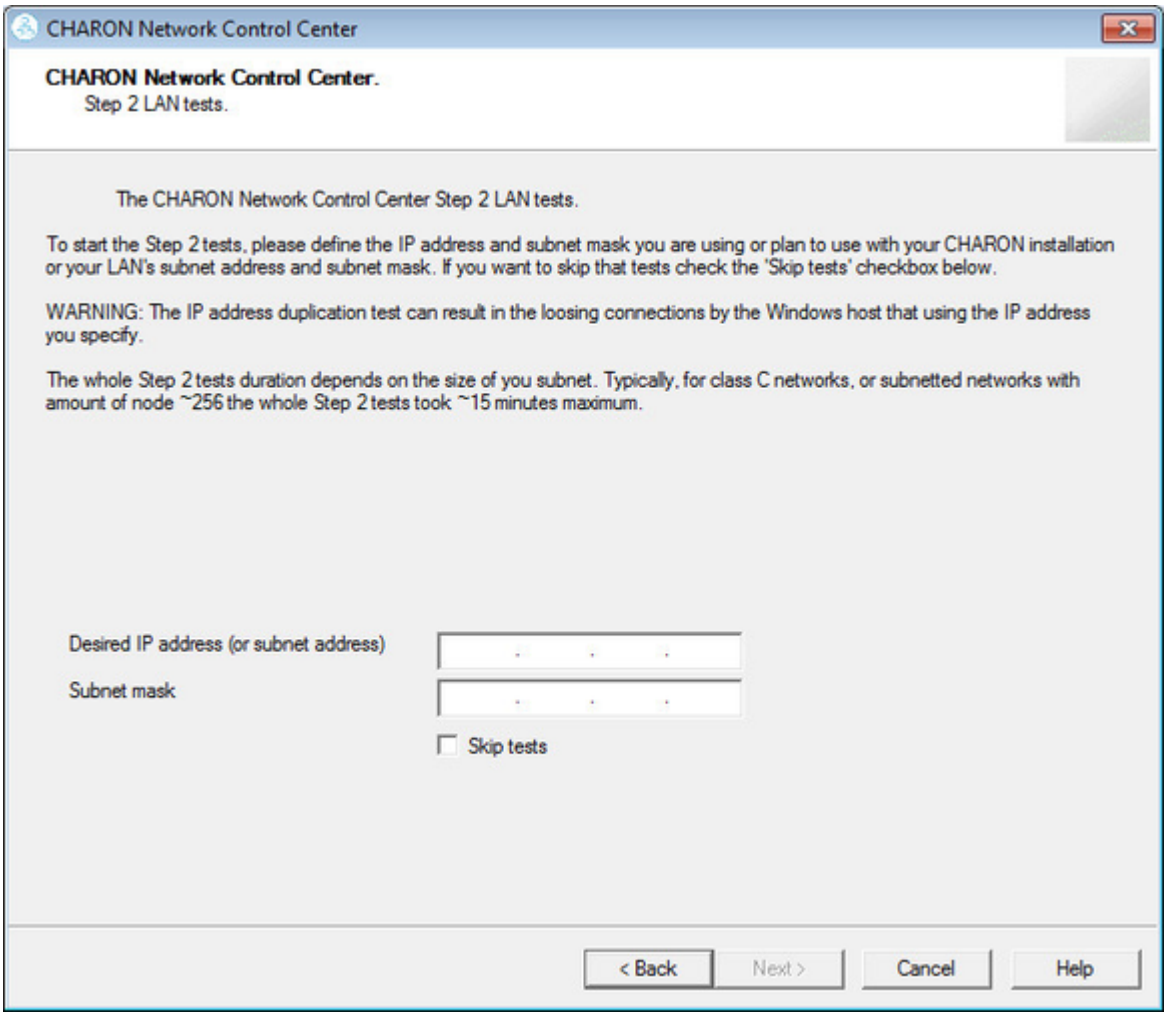
The Network Control Center offers to check whether the DECnet address, to be used by CHARON, is unique on the network in this step. Enter the desired DECnet address, for the CHARON guest, and press "Next":



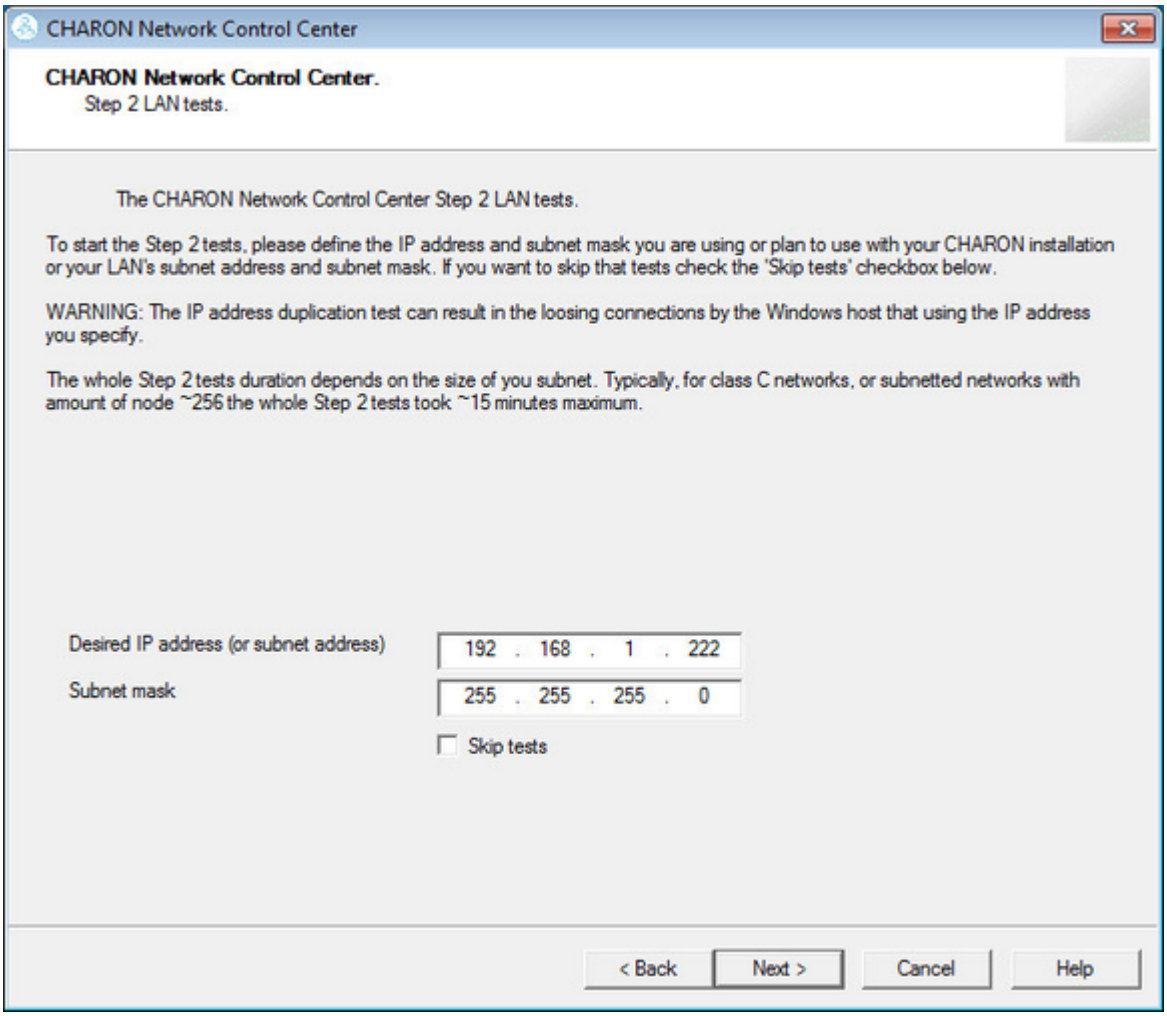
Once the checking is completed, if there are any issues, the utility will abort the test and display a log containing information about the issues found:



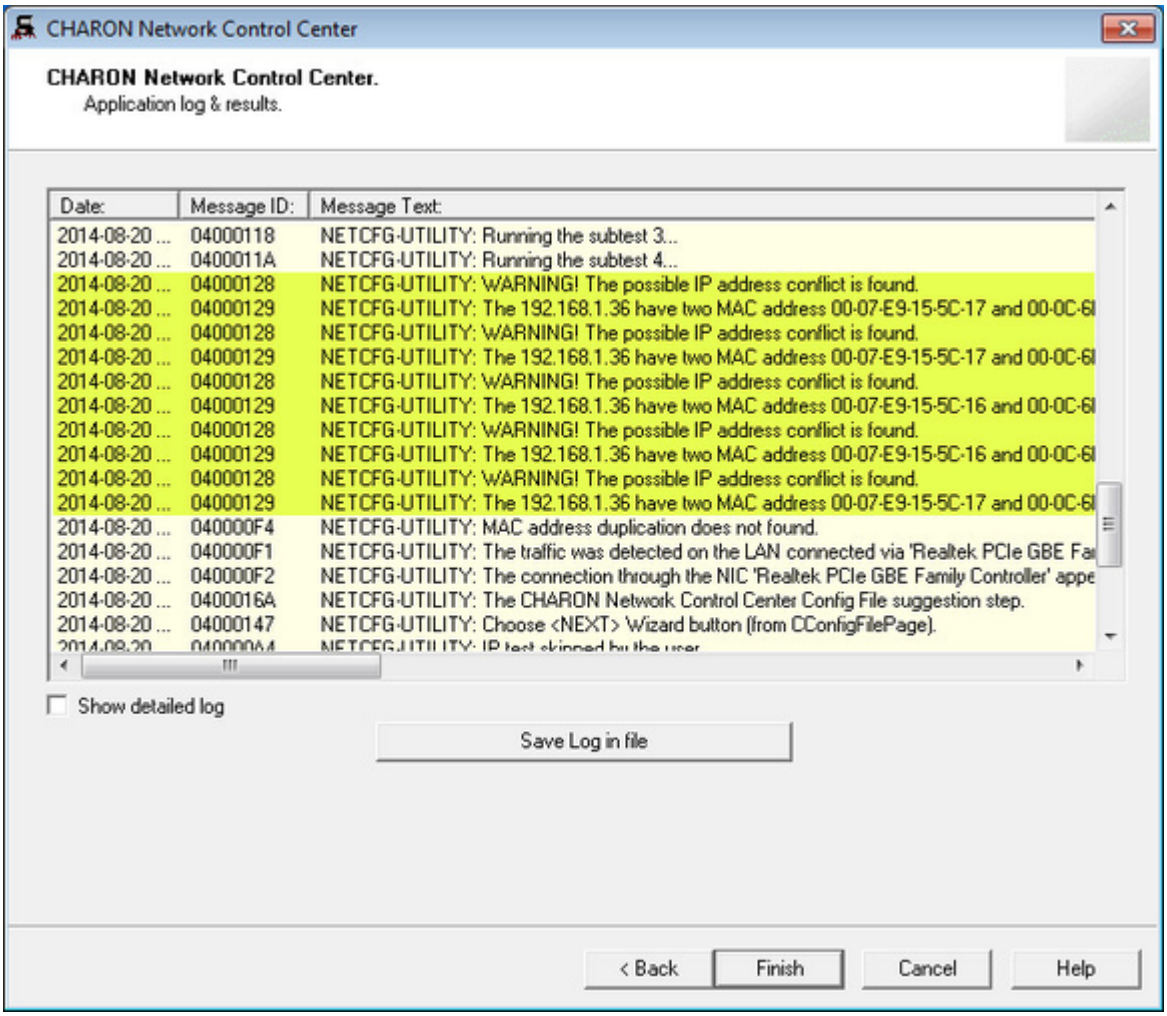
If no issues are found, the utility will offer to check the CHARON TCP/IP networking the same way:



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



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



Select "Show detailed log" for more detail.

Press "Finish" to exit.

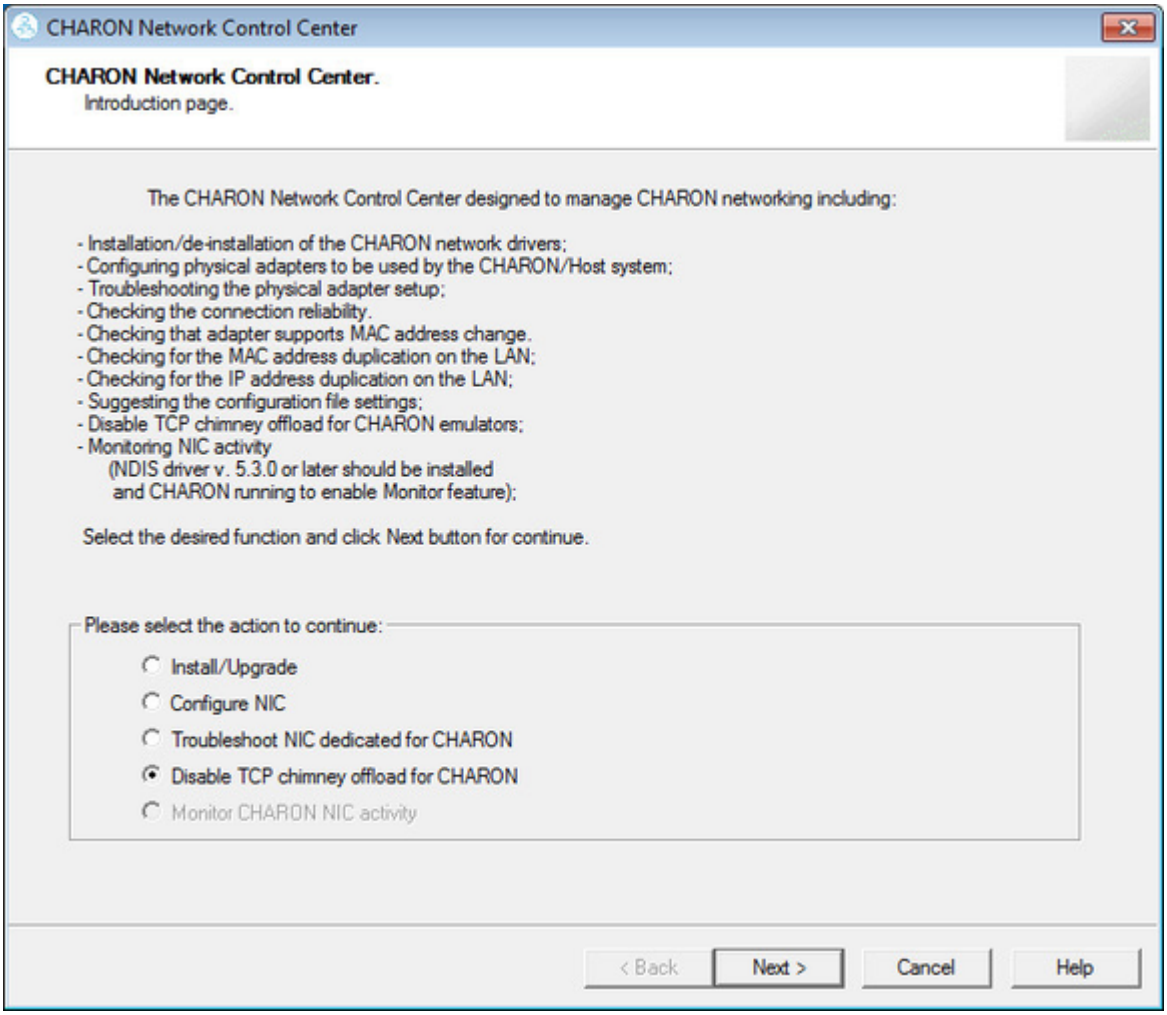
Back to Table of Contents

© Stromasys, 2017

Disable "offload" parameters

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

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



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

Press "Finish" to exit.

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

More information on details of disabling TCP chimney offload is available

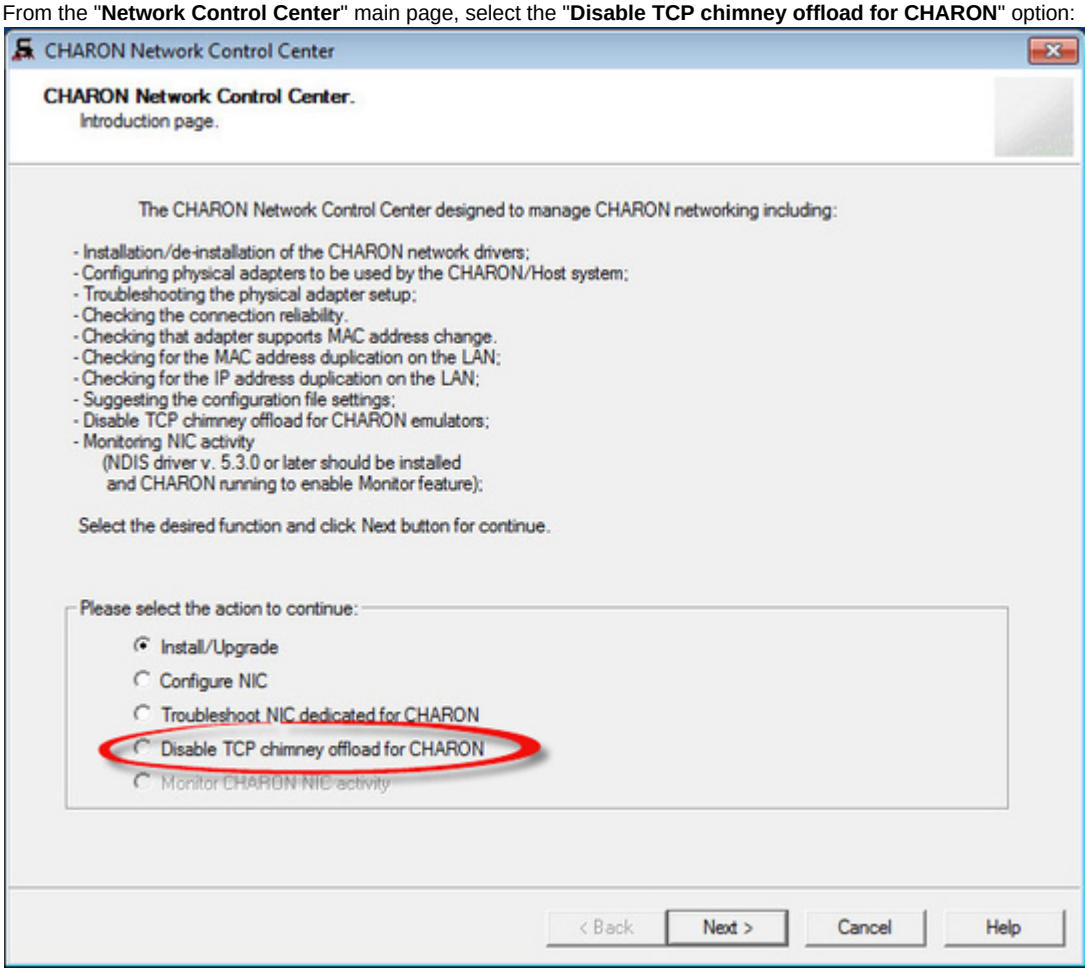
For Windows 2008 R2, Windows 2012 R2, Windows 7, 8 and 8.1

TCP Chimney offload must be disabled.

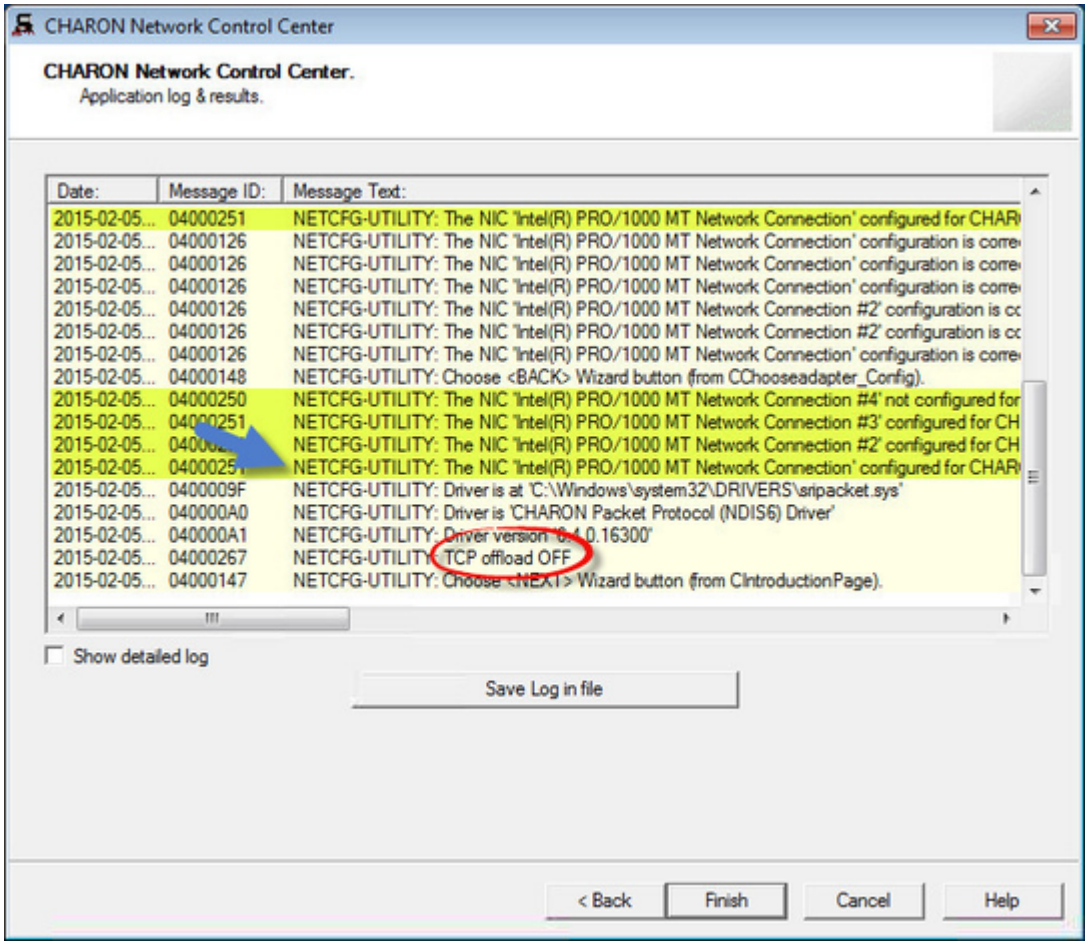
TCP Chimney Offload is a networking technology that helps transfer the workload from the CPU to a network adapter during network data transfer.

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

- Starting with CHARON-VAX V4.6 and CHARON-AXP V4.6, the Network Control Center Utility allows you to disable TCP Chimney offload:
  - 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. In the CHARON versions prior to 4.8 open the **Start** menu, select "**All Programs**" then "**CHARON**", select your Charon product version, "**Utilities**" then "**Network Controler Center version *x.xx***".
  - From the "**Network Control Center**" main page, select the "**Disable TCP chimney offload for CHARON**" option:



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



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

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

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

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

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

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

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

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

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

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

For older versions of Windows

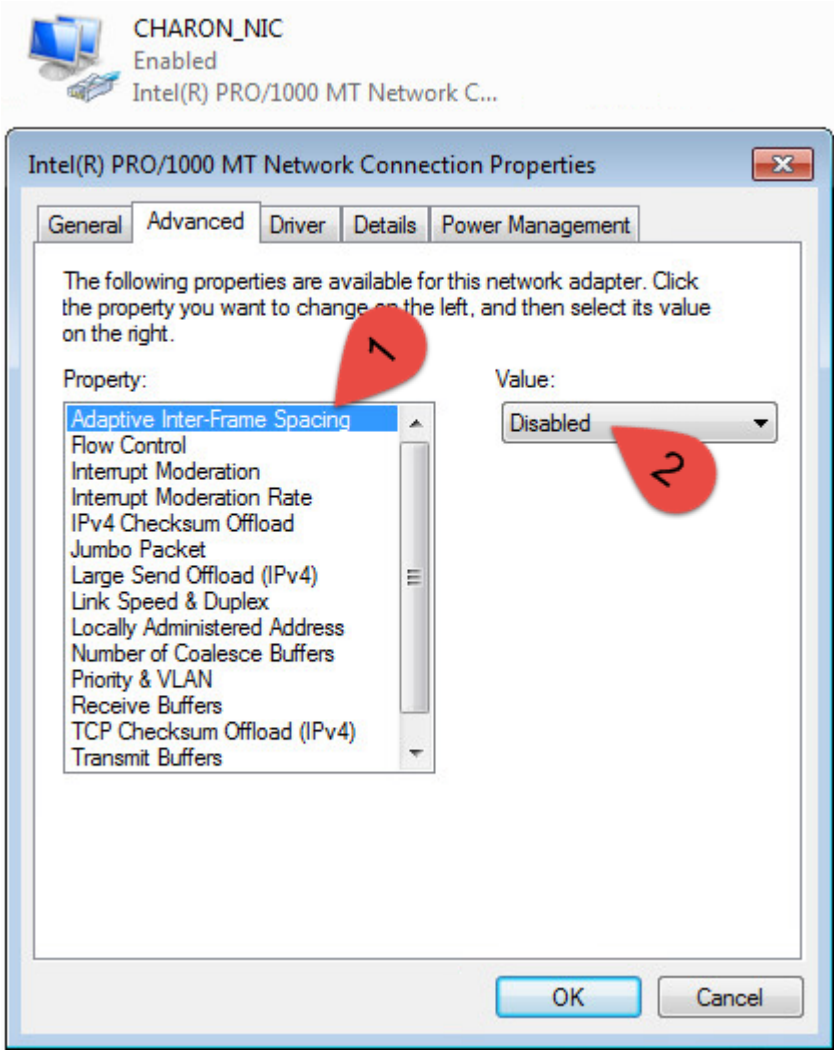
Intelligent packet processing should be switched off for the network adapters dedicated to CHARON-AXP (naming of the parameters depends on the network adapter driver):

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

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

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

Example:



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

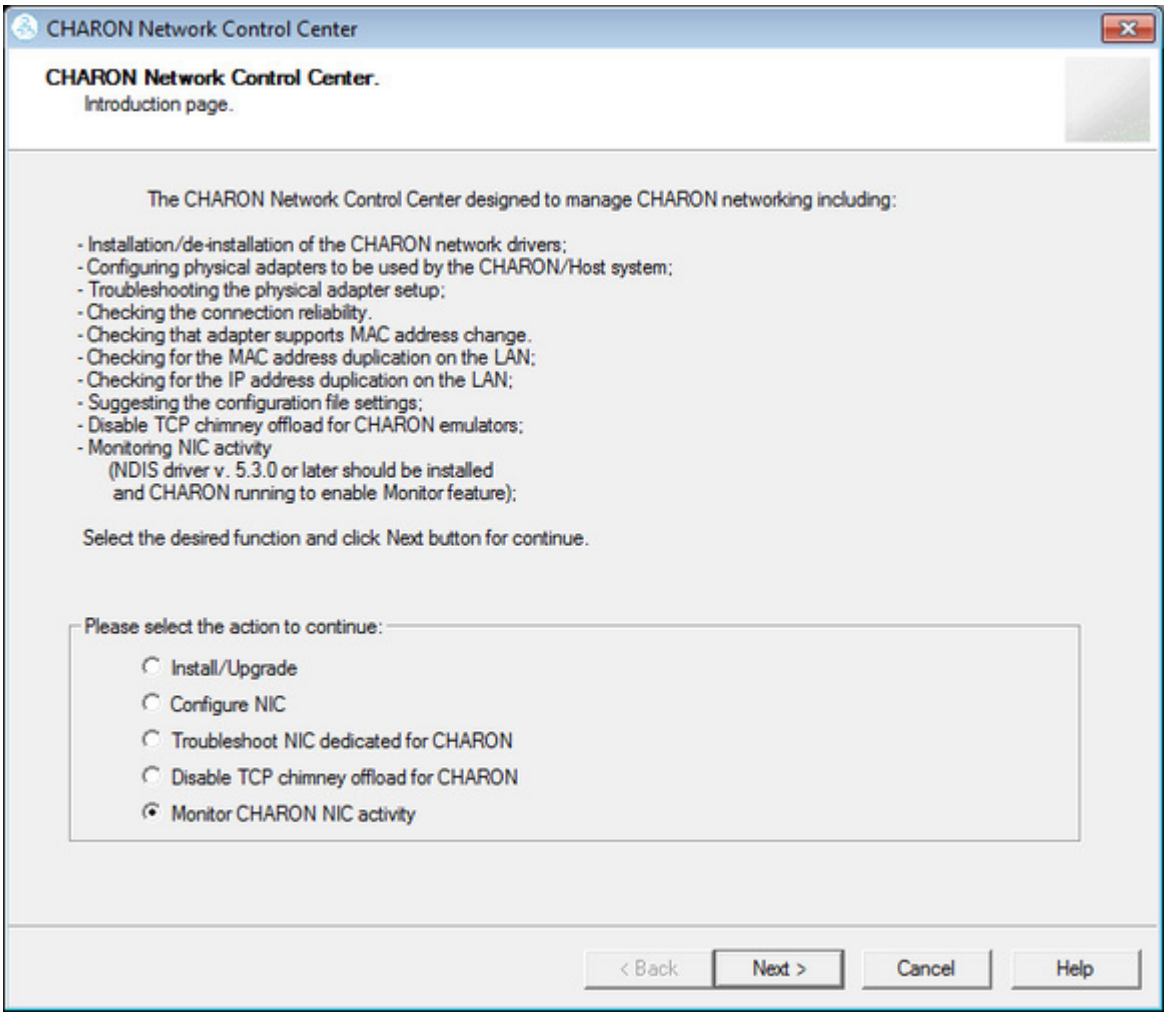
> **Warning**
> Running guests will have to be powered off/on for the new settings to be taken into account

Back to Table of Contents

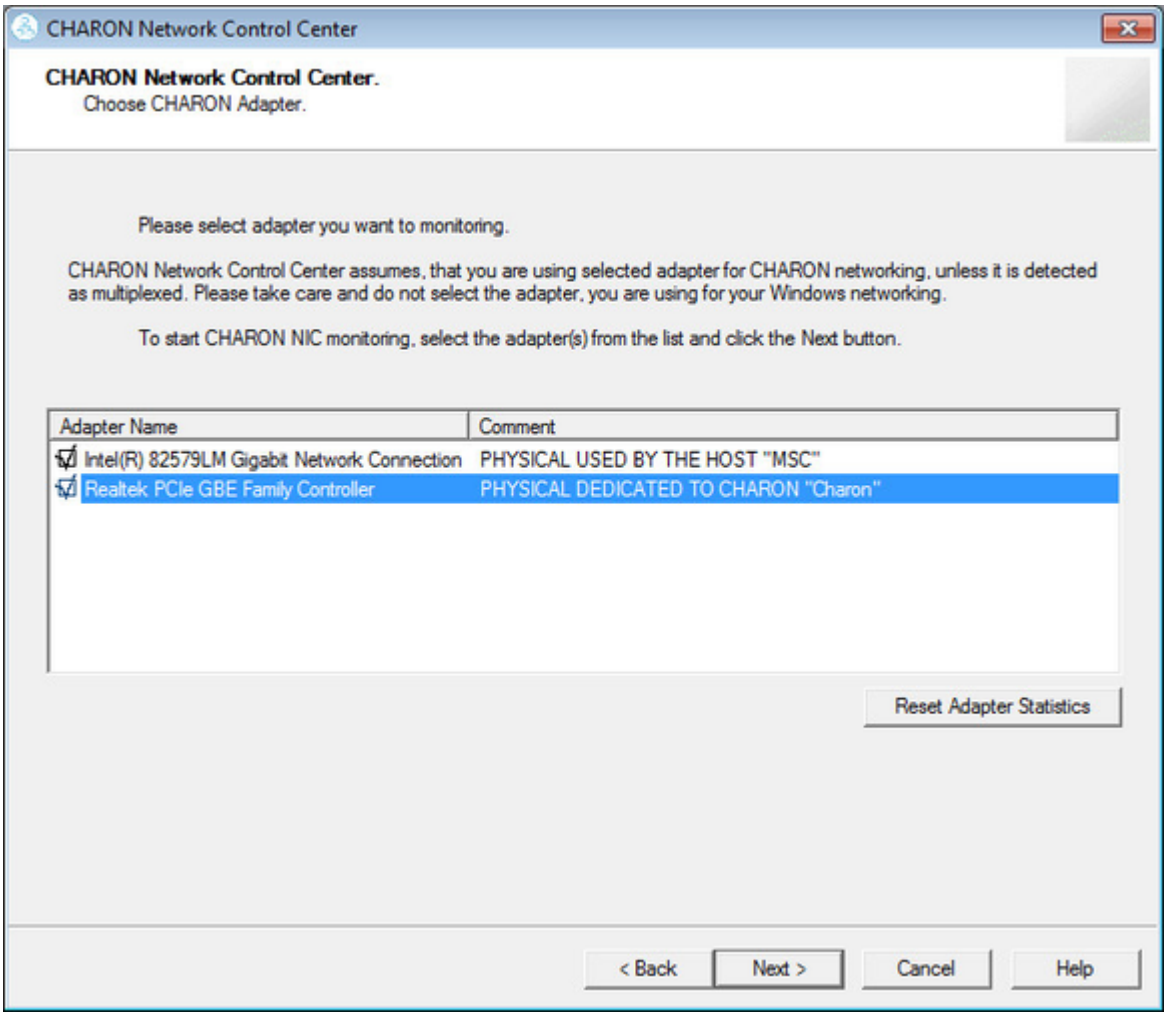
Monitoring the CHARON network activity

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

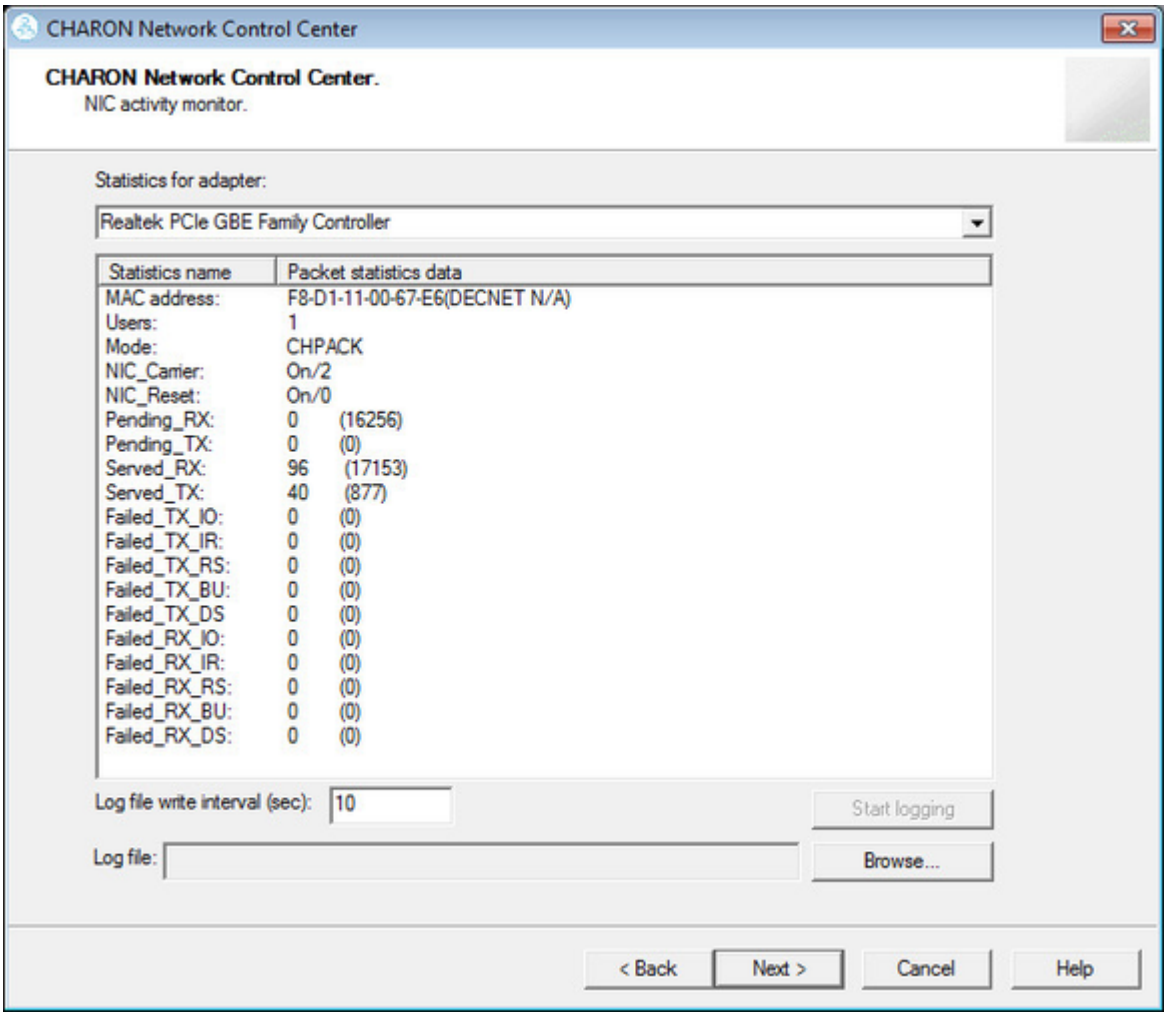
Start the utility and select "Monitor CHARON NIC activity"; press "Next":



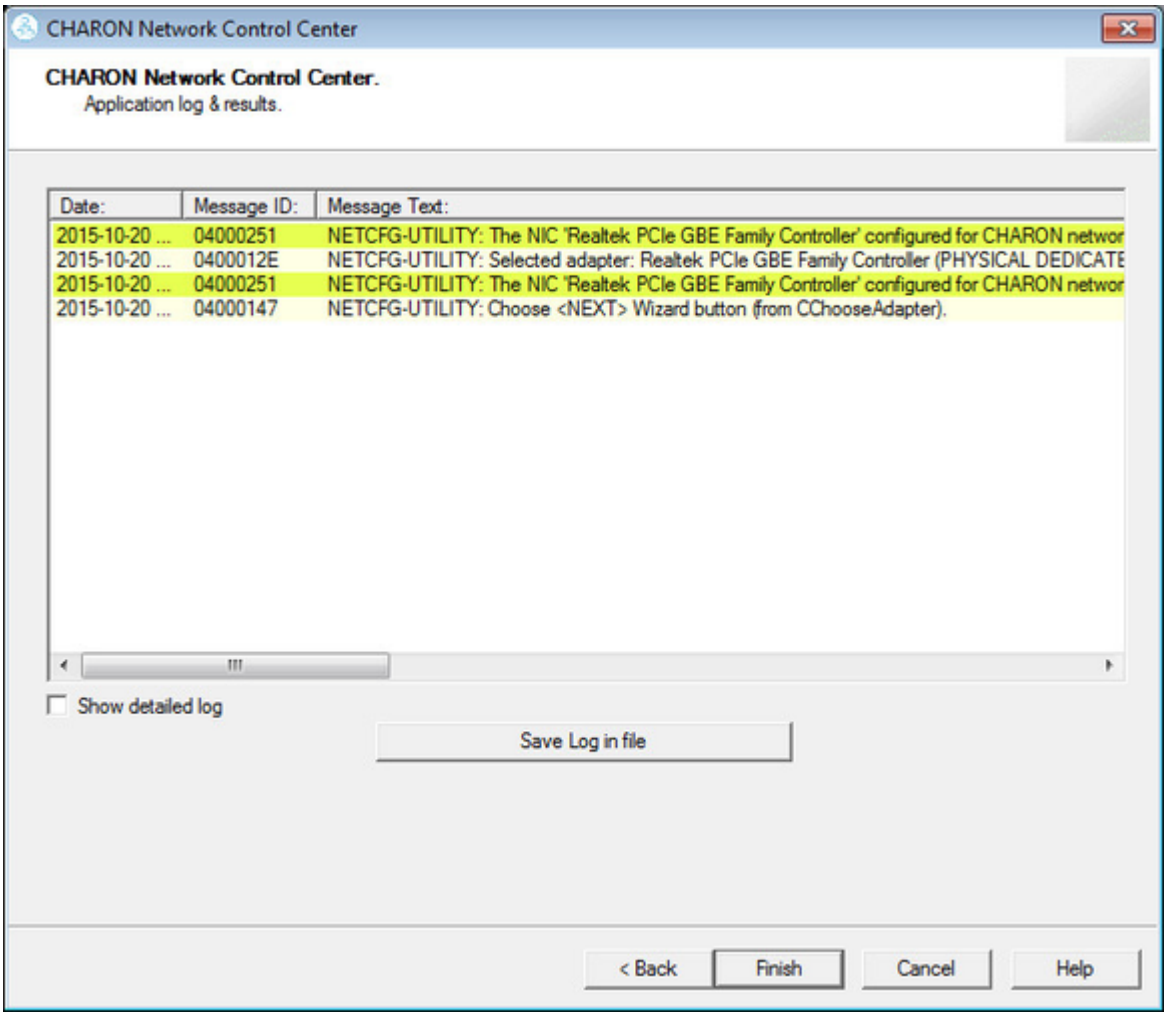
Select the network interface to monitor (it must be dedicated to CHARON); press "Next":



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



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



Select "Show detailed log" for more detail.

Press "Finish" to exit.

Back to Table of Contents

© Stromasys, 2017

MkDisk

Table of Contents

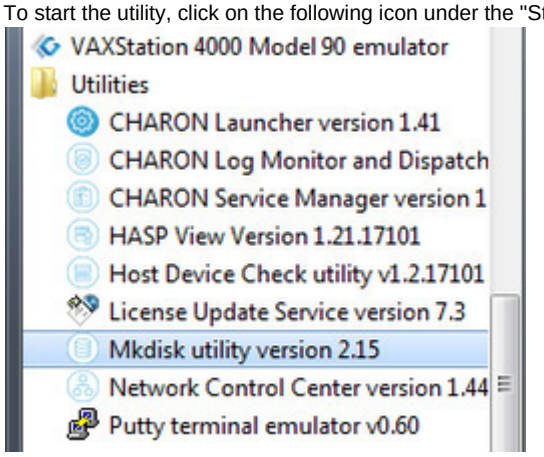
- General Description
- Creating empty disk container
- Creating disk metadata
- Creating custom disk image
- Getting information about available disk types

Back to Table of Contents

General Description

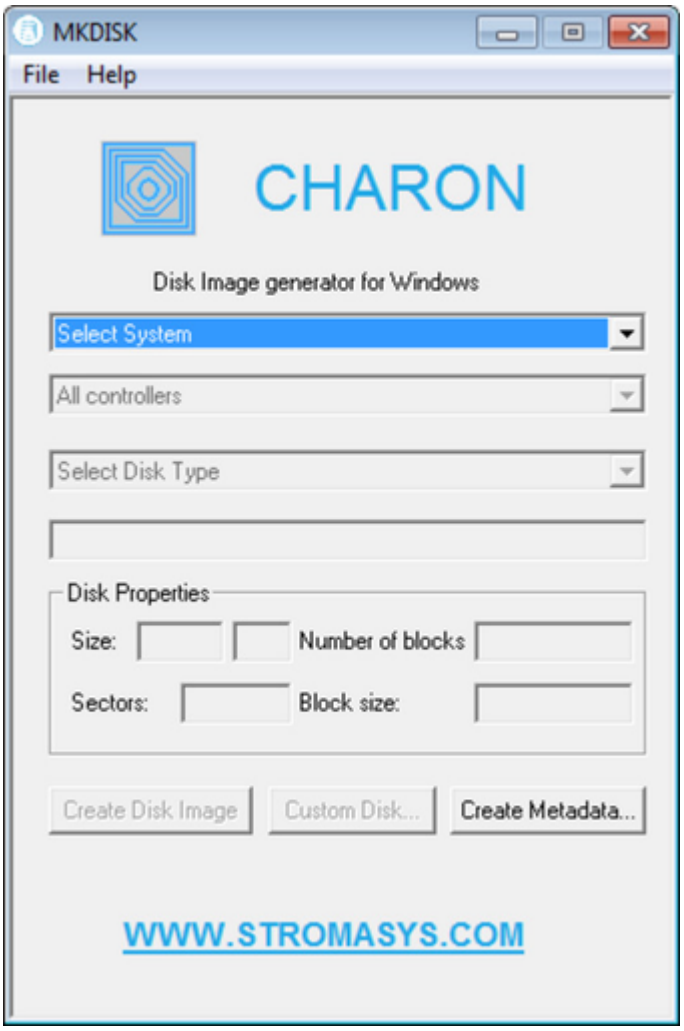
The "MkDisk" utility is used to create VAX/PDP11 standard or custom empty disk containers (disk images).

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



© Stromasys, 2017

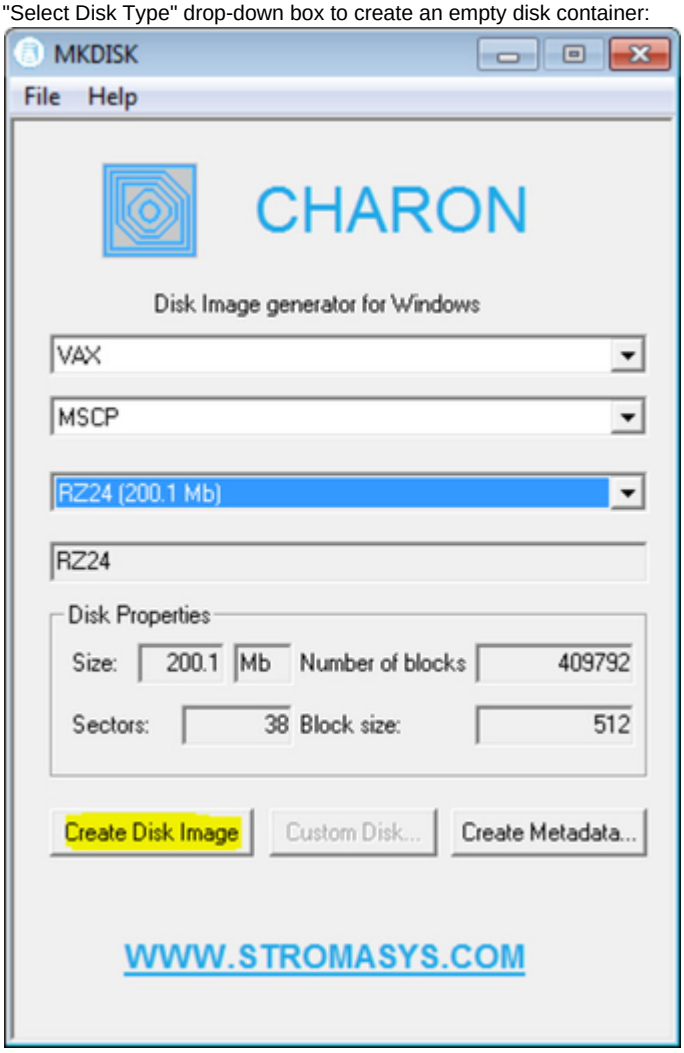
The main dialog of the utility will appear:



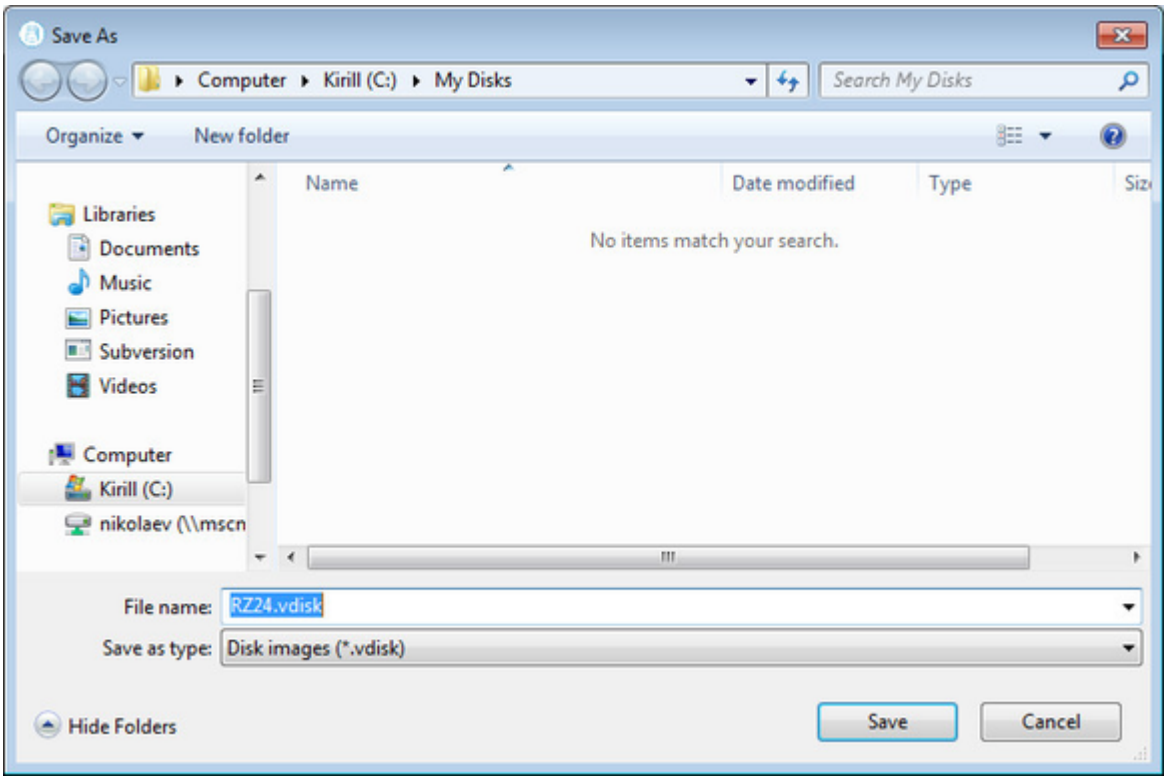
Back to Table of Contents

Creating empty disk container

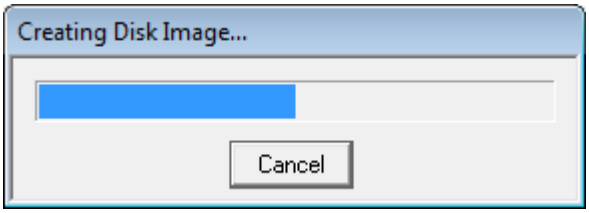
Select "VAX" or "PDP-11" in the "Select System" drop-down box, "MSCP" in the "All Controllers" drop-down box and choose the desired disk type in the "Select Disk Type" drop-down box to create an empty disk container:



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



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



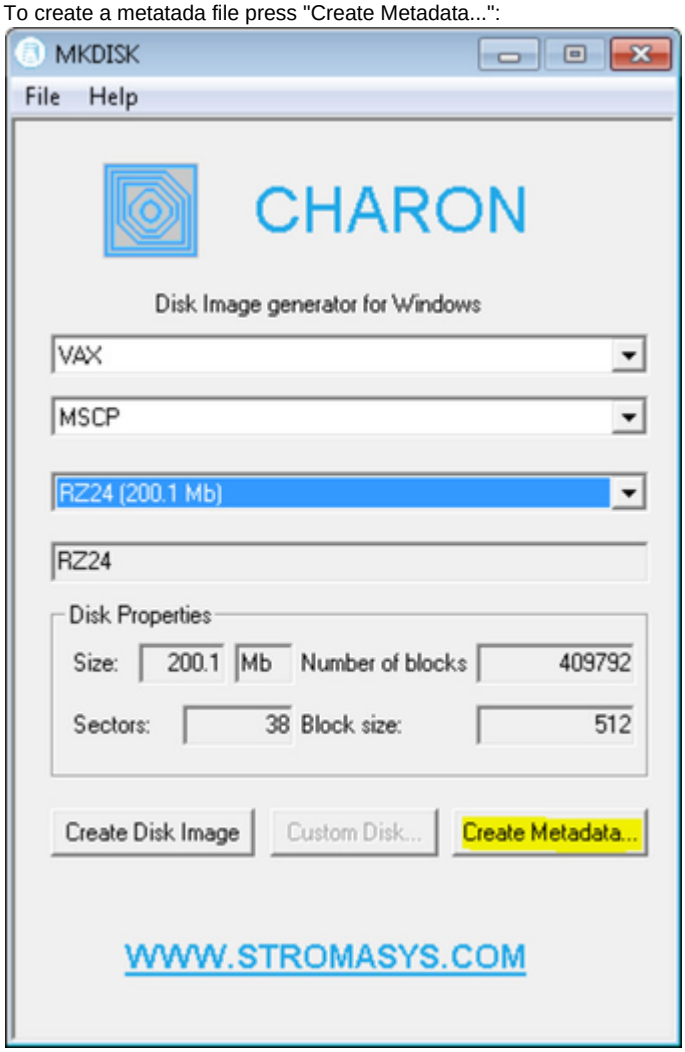
The target disk image is created - along with its metadata file (see below) once the process completes.

Back to Table of Contents

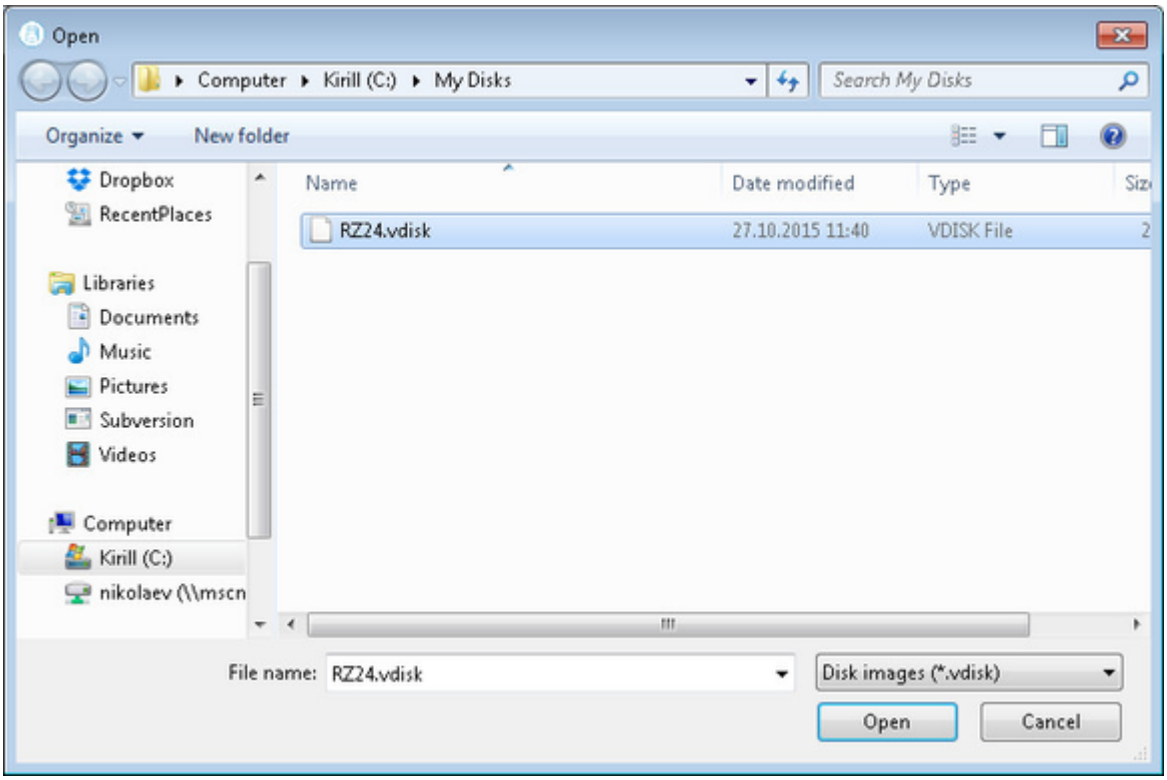
Creating disk metadata

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

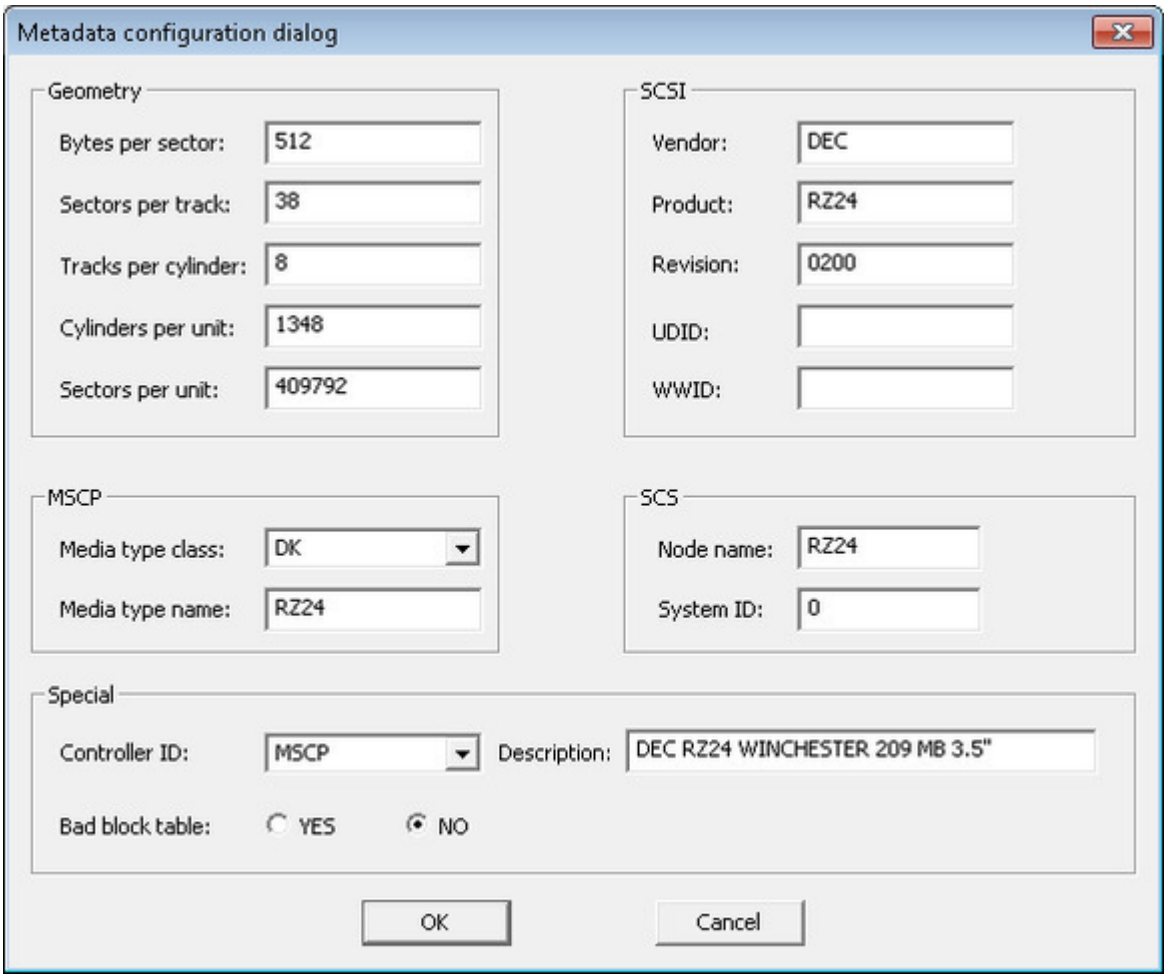
To create a metatada file press "Create Metadata...":



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



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



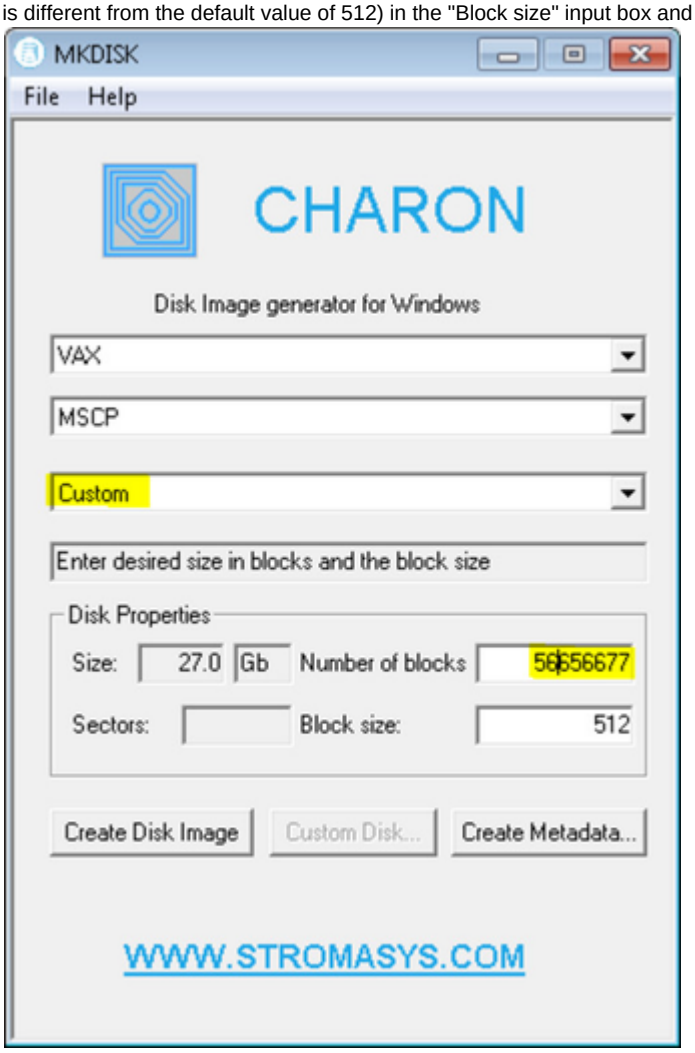
Correct the desired parameters (if needed) and press "Ok". The "MkDisk" utility will create a metadata file, having the same name as the target disk container, with extension ".avdisk"

Back to Table of Contents

© Stromasys, 2017

Creating custom disk image

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



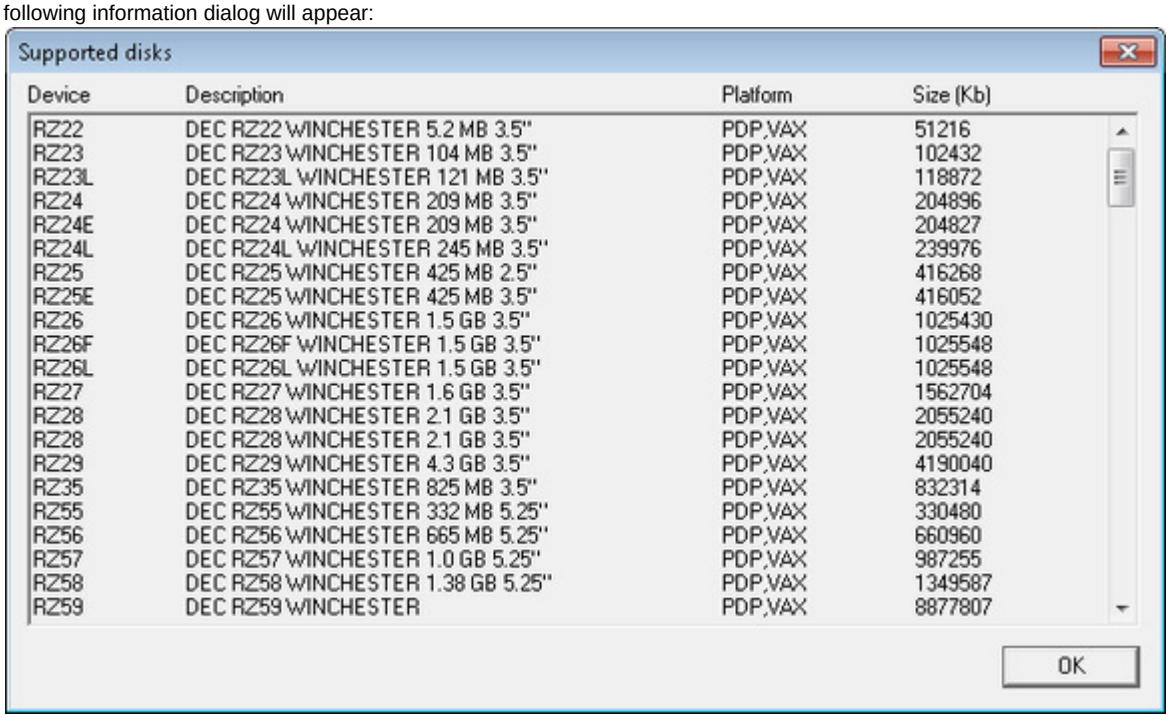
Specify the name of the target disk container (and the directory it will reside in) and press "Save". The utility will create the disk image.

Back to Table of Contents

© Stromasys, 2017

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:



Back to Table of Contents

© Stromasys, 2017

MkDskCmd

Table of Contents

- General Description
- Creating disk images
- Transferring disk images

Back to Table of Contents

General Description

The "MkDskCmd" utility:

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

To start the utility open "cmd.exe" in the Start menu and switch to the CHARON x86 utilities directory (typically "`C:\Program Files (x86)\CHARON\Utilities_X.X.XXXXX\x86`" or "`C:\Program Files\CHARON\Utilities_X.X.XXXXX\x86`").

Back to Table of Contents

Creating disk images

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

```
...> 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.

© Stromasys, 2017

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

```
...> mkdskcmd -help

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

Usage:
mkdskcmd [Options]

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

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

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

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

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

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

-t - please see the '-transform' options description
-transform <source_disk_name> <source_disk_params> - to transform the the disk image (change actual size)
<source_disk_name> - the file name of the disk image to be transformed
<source_disk_params> - the name of the disk from the list of available at the Disk table
The source disk size will changed accordingly the reach the specified parameters.
To specify the transform parameters manually, follow the option below:
-transform <source_disk_name> -blsize <number> -blcount <number>

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

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

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

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

Examples:
mkdskcmd -help
mkdskcmd -list
mkdskcmd -avtable \etc\mkdsk.vtable -output "\etc\rk07.vdisk" -disk rk07
mkdskcmd -output \etc\disks\custom.vdisk -blsize 512 -blcount 16384
```

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

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

Back to Table of Contents

Transferring disk images

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

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

Note: it is not possible to add more free space dynamically. CHARON-VAX/PDP11 must be stopped before performing this operation.

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

An example of the syntax follows:

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

where:

- <source disk file name> - the file name of the disk image to be transferred
- <source disk parameters> - the name of the disk from the list of available on "mkdskcmd --list" request or the disk geometry specification (see below).
- -shrink or -k - used in the case when the target disk is transferred to a smaller disk.

Example:

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

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

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

Example:

```
...> mkdskcmd -t "C:\My Disks\custom.vdisk" -z 512 -c 262134
```

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

Back to Table of Contents

© Stromasys, 2017

HASP View

Table of Contents

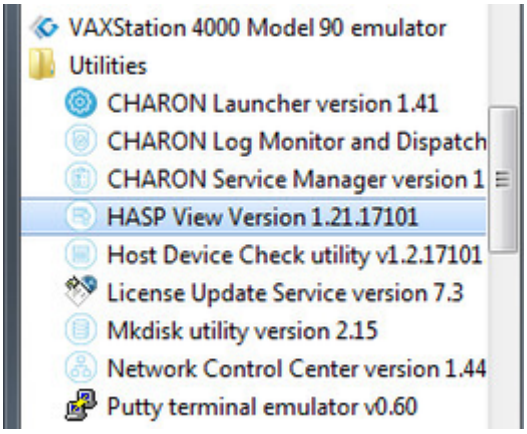
- General Description
- Getting CHARON licenses content

Back to Table of Contents

General Description

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

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

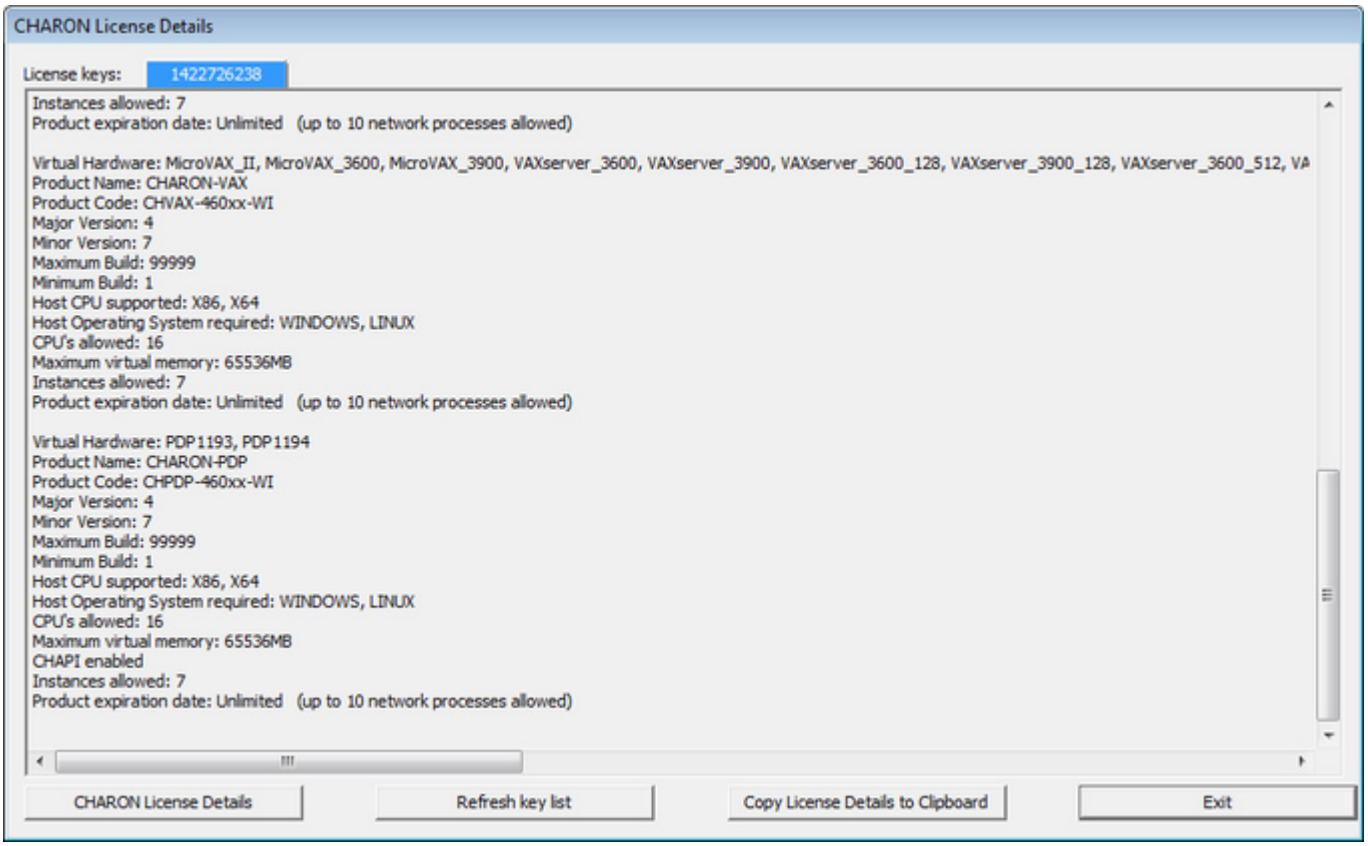


Back to Table of Contents

© Stromasys, 2017

Getting CHARON licenses content

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



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

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

Note: collecting the license content may take some time.Use the buttons below to refresh the license content and copy the license to clipboard.

Press "Exit" to exit from the utility.

Back to Table of Contents

© Stromasys, 2017

License Update Service

Table of Contents

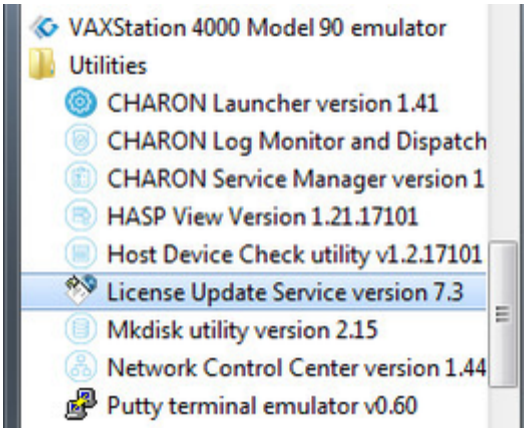
- General Description
- Collecting the host fingerprint and information on an existing license
- Installation and update of a CHARON license
- CHARON software license transfer
- Software License Removal

Back to Table of Contents

General Description

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

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



Back to Table of Contents

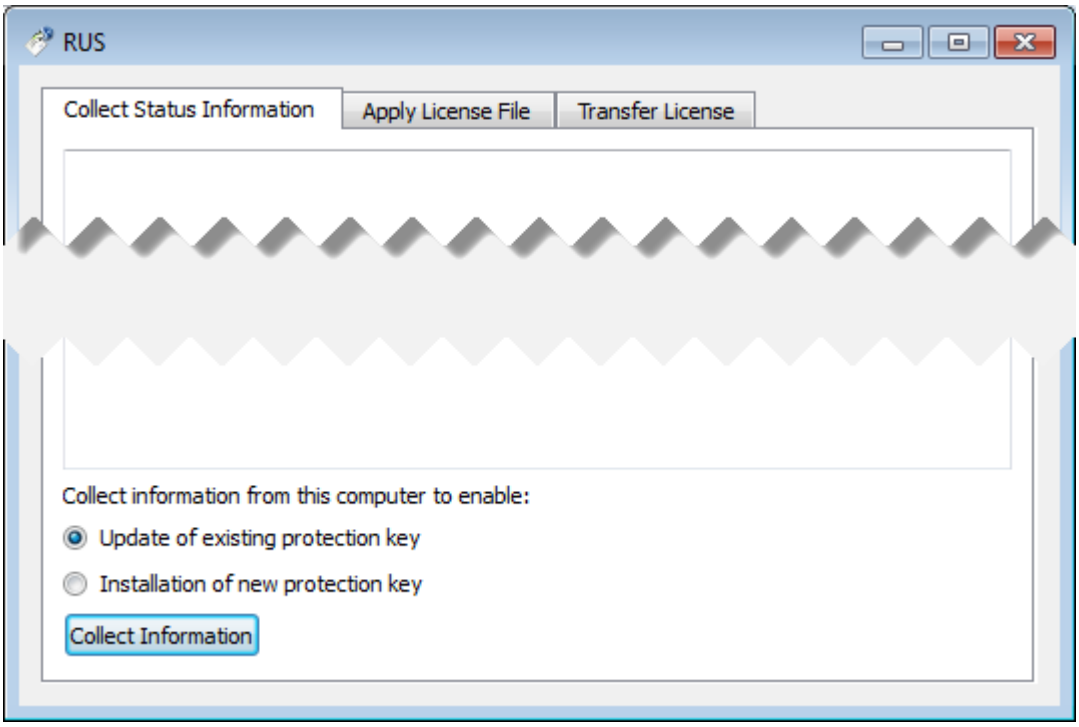
Collecting the host fingerprint and information on an existing license

Open the "Collect Status Information" tab.

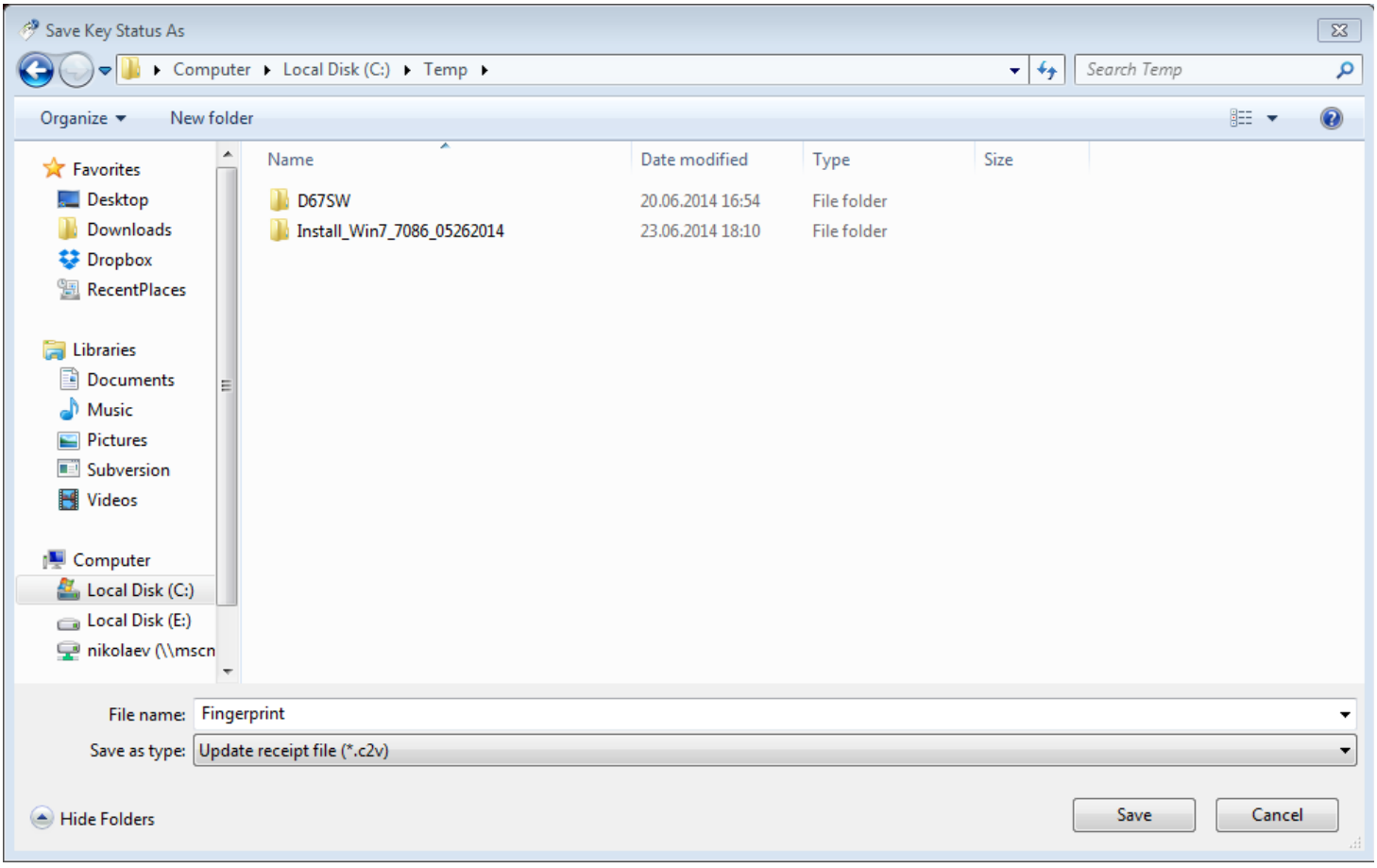
Select "Update of existing protection key" to acquire information on the current license (both the hardware dongle or the software license).

© Stromasys, 2017

Select "Installation of new protection key" to get information on the host system.



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



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

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

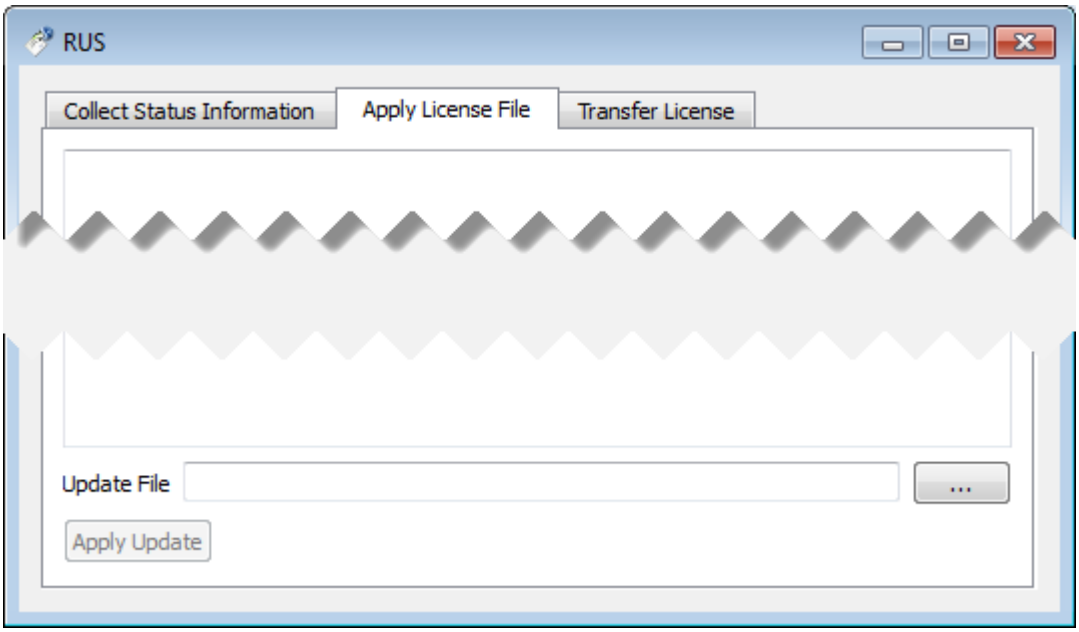
Back to Table of Contents

© Stromasys, 2017

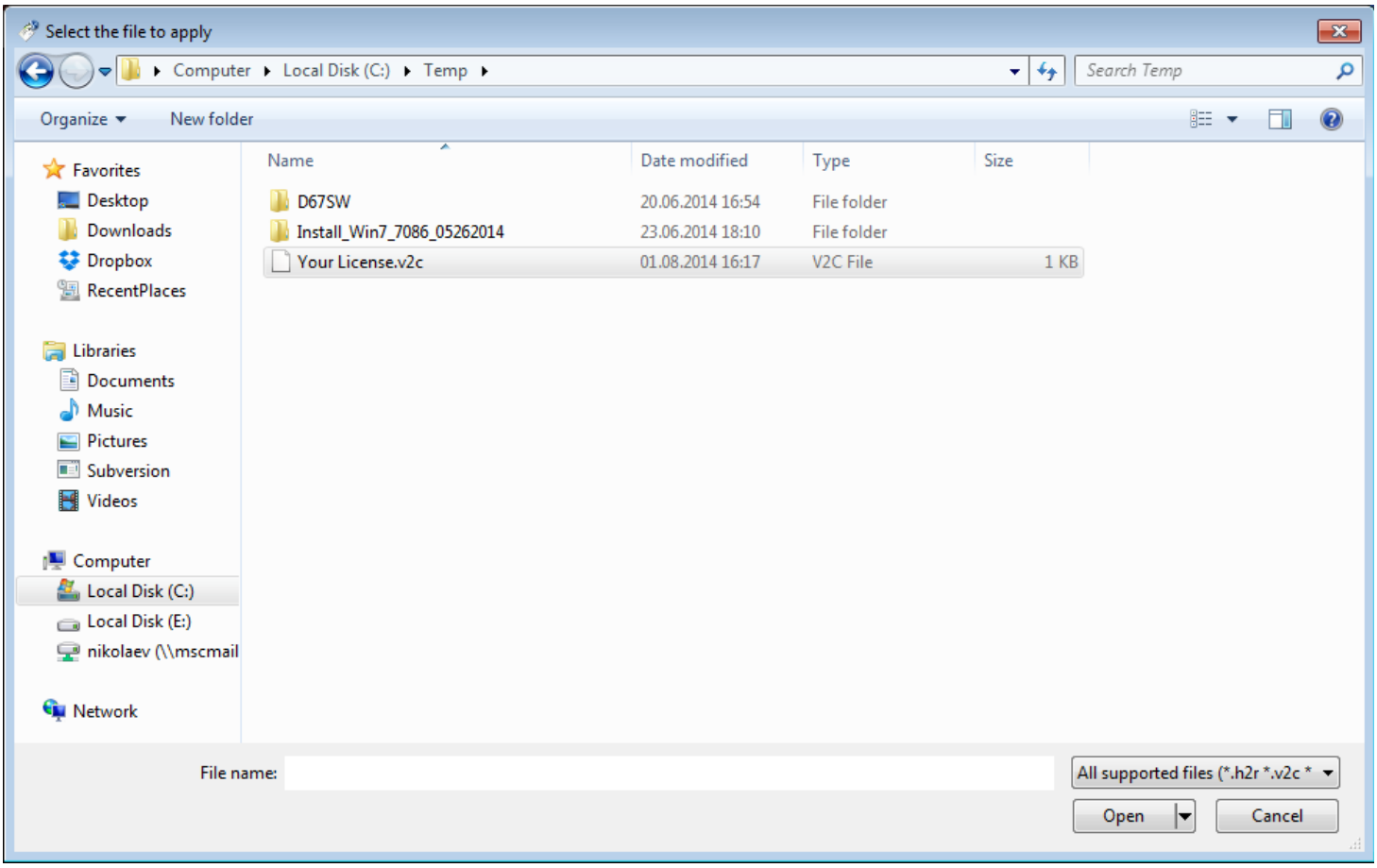
Installation and update of a CHARON license

Open the "Apply License File" tab.

Press "...":



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



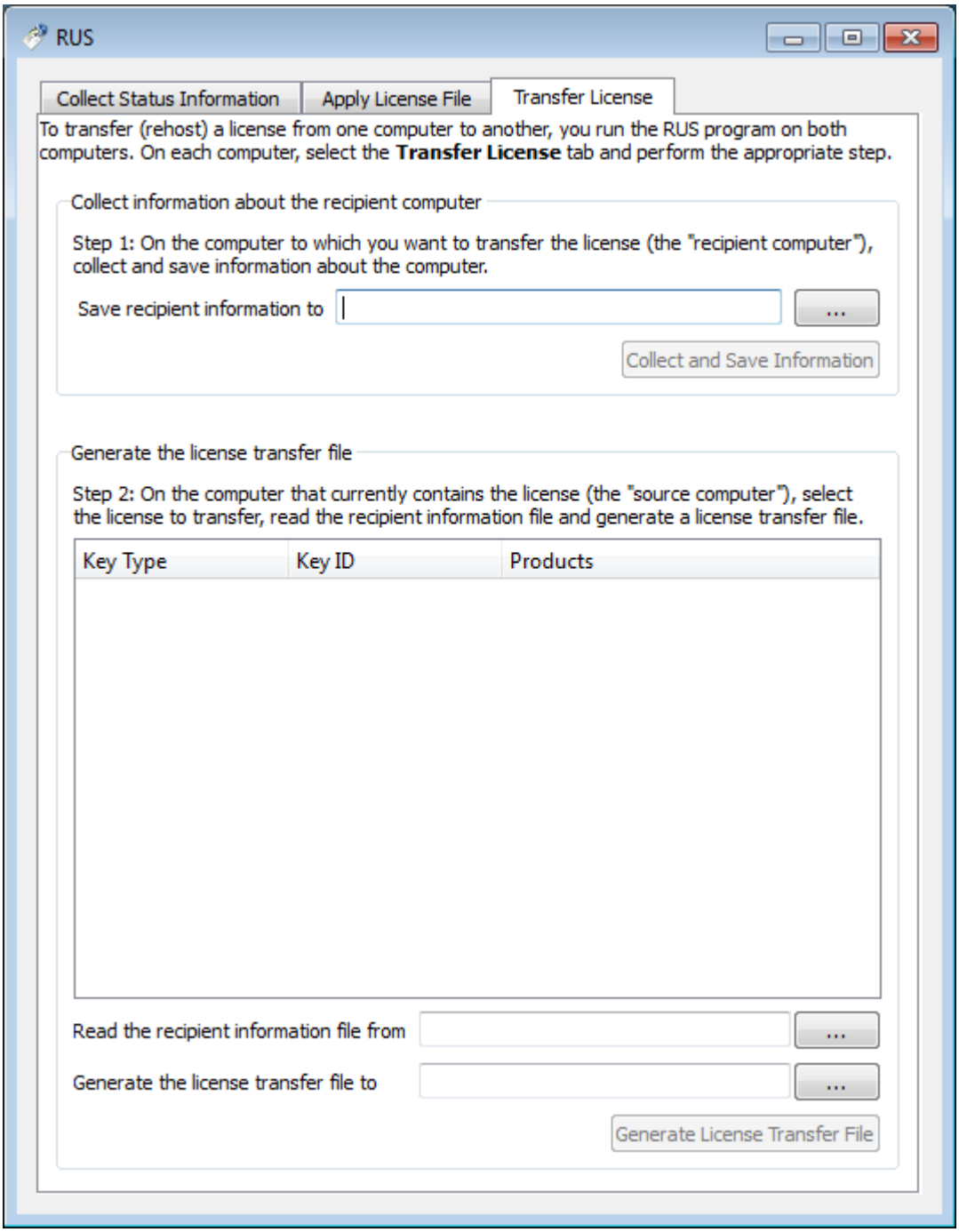
Press "Open" and then "Apply Update" in the main dialog box to apply the new license or to update an existing one.

Back to Table of Contents

© Stromasys, 2017

CHARON software license transfer

Open the "Transfer License" tab:



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

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

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

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

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

   Press the "..." beside the "Read the recipient information from file" textbox, choose the just transferred RecipientHost file, press "Open".
   Press the "..." beside the "Generate the license transfer file to" textbox, choose the target directory and enter the desired name of the file (*.h2h), press "Save".
   Press "Generate License Transfer File" to create the license transfer file in the specified folder.
4. Copy the resulting "*.h2h" file to the "RecipientHost".

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

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

Back to Table of Contents

Software License Removal

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

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

Back to Table of Contents

© Stromasys, 2017

mtd

General Description

"mtd" is a command line utility to:

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

Open "cmd.exe" in the Start menu, to start the utility, and switch to the CHARON x86 utilities directory (typically "C:\Program Files (x86)\CHARON\Utilities_X.X.XXXXX\x86" or "C:\Program Files\CHARON\Utilities_X.X.XXXXX\x86").

Back to Table of Contents

Usage

The following is an example of the usage syntax:

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

Options:

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

Example:

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

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

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

Example:

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

Back to Table of Contents

© Stromasys, 2017

CHARON Log Monitor and Dispatcher

Table of Contents

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

Back to Table of Contents

General Description

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

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

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

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

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

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

Starting in background mode

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

1. Open "cmd.exe" from the "Start" menu.
2. Change current directory to the folder (presumably) containing the LOG file.
   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 (x86)\CHARON\Utilities_X.X.XXXXX\x86\logmond" my_charon.log
   ```

   For rotating log file:

   ```
   ...> "C:\Program Files (x86)\CHARON\Utilities_X.X.XXXXX\x86\logmond" -l <log-directory> -p <log-prefix>
   ```

   where:

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

Example:

```
...> "C:\Program Files\CHARON\Utilities_1.0.XXXXX\x86\logmond" -l "C:\My CHARON logs" -p "MY_VAX"
```

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

© Stromasys, 2017

Back to Table of Contents

Installing as a service

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

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

```
...> "C:\Program Files\CHARON\Utilities_1.0.XXXXX\x86\logmond" -r
...> "C:\Program Files\CHARON\Utilities_1.0.XXXXX\x86\logmond" -i
```

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

Back to Table of Contents

Starting the Log Monitor and Dispatcher service

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

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

```
...> "C:\Program Files\CHARON\Utilities_1.0.XXXXX\x86\logmond" -u
```

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

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

It does NOT monitor emulator instances started from Launcher or manually from command line prompt. 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).

Back to Table of Contents

Stopping the Log Monitor and Dispatcher service

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

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

```
...> "C:\Program Files\CHARON\Utilities_1.0.XXXXX\x86\logmond" -d
```

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

Back to Table of Contents

© Stromasys, 2017

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\Utilities_1.0.XXXXX\x86\logmond" -r
```

Back to Table of Contents

© Stromasys, 2017

Performance Improvement Package

Table of Contents

- General Description
- Usage

Back to Table of Contents

General Description

The CHARON Performance Improvement Package is a collection of the following utilities for VAX/VMS:

1. IDLE
2. SHUTDOWN
3. SHUTDOWN_R
4. SPEEDUP
5. SLOWDOWN
6. SLOWDOWN_R
7. SLOW_TEST

These utilities are located in the "idle_vms_pkg.vdisk" disk file in the "C:\Program Files (x86)\CHARON\Virtual Disk Images\idle_vms_pkg_v2.0" folder. (Windows x64; for Windows x86 the path is: "C:\Program Files \CHARON\Virtual Disk Images\idle_vms_pkg_v2.0").

Back to Table of Contents

Usage

Specify this image in the CHARON-VAX configuration file, boot from the system disk and mount the disk with the following VAX/VMS command:

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

© Stromasys, 2017

In the root of this disk you will see the following executable files and the COM file designed for the VAX/VMS system running on CHARON-VAX:

Utility	Description
IDLE.EXE	Significantly reduces the CHARON-VAX/XX host CPU usage whenever a VAX/VMS system running on CHARON-VAX is idle. To ensure this is automatically started, you may include a call to the IDLE.EXE in the system startup file. IDLE.EXE stalls the emulated CPU when it detects an OpenVMS idle condition. While IDLE.EXE is running the emulated CPU consumes, on average, less host system CPU time. It is not recommended to employ IDLE.EXE in real-time process control environments. Note: IDLE.EXE cannot be used with the VAX 6*xx*0 emulators.
IDLE_VMS55.EXE	IDLE.EXE version for the VMS 5.5-2 and VMS 5.5-2H4 instances.
SHUTDOWN.EXE	Stops the CHARON-VAX emulator 30 seconds after the utility is invoked. These 30 seconds are intended to be used to shutdown the VAX/VMS system running on CHARON-VAX.
SHUTDOWN3.EXE	Stops CHARON-VAX 3 minutes after the utility is called.
SHUTDOWN5.EXE	Stops CHARON-VAX/XX 5 minutes after the utility is called.
SHUTDOWN_R.EXE	Cancels a pending request to SHUTDOWN.
SLOWDOWN.EXE	Slows down the CHARON-VAX execution. Repeated execution further slows down the execution.
SLOWDOWN_R.EXE	Restores the normal CHARON-VAX/XX execution speed.
SPEEDUP.EXE	Raises CHARON-VAX execution speed by a certain amount of steps. Only works if SLOWDOWN.EXE has been applied.
SLOW_TEST.COM	A test that demonstrates the effect of the CHARON slowdown mechanism.

To run any of these utilities issue the following command:

```
$ RUN <name of the utility>.EXE
```

or

```
$ @<name of the utility>.COM
```

> The SHUTDOWN*n*.EXE routines **do not** initiate the shutdown of the VAX/VMS system running on CHARON-VAX, only of the emulator itself.
>
> Establish a VMS shutdown procedure that can complete within the specified time

Back to Table of Contents

© Stromasys, 2017

HOSTPrint

Table of Contents

- General Description
- Usage

Back to Table of Contents

General Description

The HOSTPrint utility is used to print CHARON output to Windows printers. It receives data from 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.

Back to Table of Contents

Usage

HOSTPrint calling is specified in the CHARON configuration file. The call implement one or more of the following parameters:

Parameter	Description
-host=<hostname>	Name of the host - source of the printing data.
-port=<connection port number>	Port on the host to get the information to print from.
-delay=<delay for automatically buffer flush in seconds>	Flushing delay, 0 - wait infinite, 5..10800 - timeout for flush

-printer=[PrinterDeviceName]	Host name for the printer used.
	
	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.

© Stromasys, 2017

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.

Back to Table of Contents

© Stromasys, 2017

VT-30TV

Table of Contents

- General Description
- Usage

Back to 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" located in the "C:\Program Files (x86)\CHARON\Utilities_1.0.15804\x86" (Windows x64; for Windows x86 use the path "C:\Program Files\CHARON\Utilities_1.0.15804\x86" instead).

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
```

Back to Table of Contents

© Stromasys, 2017

CHARON-VAX and CHARON-PDP for Windows configuration details

Introduction

This chapter describes, in detail, all of the configuration parameters of the devices emulated by CHARON-VAX / CHARON-PDP for Windows, with corresponding examples and parameters.

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

Example:

```
load 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

Back to Table of Contents

© Stromasys, 2017

General Settings

Table of Contents

- Session
  - hw_model
  - configuration_name
  - log
  - log_method
  - log_show_messages
  - log_file_size
  - log_rotation_period
  - log_repeat_filter
  - license_key_id
  - affinity
  - n_of_io_cpus
  - license_key_lookup_retry

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 apply.

Example:

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

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

© Stromasys, 2017

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 (ir must be unique):
	
	`set session configuration_name="MSCDV1"`
	
	The value of this parameter is used as a prefix to the event log file name if the log file name is not forced, either not set or specified as a folder (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`
	
	xxxxxxxxx is an increasing decimal number starting from 000000000 to separate log files with the same time of creation (in case the log is being written faster than one log file per second).
	
	⚠ It is strictly recommended to use the "configuration_name" parameter if more than one CHARON instance runs on the same server.

log

Parameter	log
Type	Text string
Value	The log file or directory name is where the log file for each CHARON-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 "log_method" parameter.

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 sessi on. 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 log_file_si ze) and/or when the log file is older than a specified number of days (see log_rotation_period).

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-2016-10-13-10-00-00-0 00000000.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	• "overwrite" (default) • "append" Determines if the previous log information is maintained. This parameter must be specified only in addition to “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 size less then 64K effectively makes the LOG File 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 next log file based on period of time passed. If enabled the switching to next log file appears 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" (default) and "off". If the value is "on", immediately following messages with the same identifier and system error code are not listed in the log, but they are counted. When a different log message is generated, the repeat count of the earlier log message is reported with "The previous message has been repeated N times.", and the counter is cleared. **Example**: `set session log_repeat_filter="on"`

© Stromasys, 2017

license_key_id

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

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

affinity

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

© Stromasys, 2017

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-VAX / CHARON-PDP instance reserves one third of all available host CPUs for I/O processing (round down, at least one). The “n_of_io_cpus” parameter overrides that by specifying the number of CHARON I/O CPUs explicitly. **Example**: `set session n_of_io_cpus=2`

license_key_lookup_retry

Parameter	license_key_lookup_retry
Type	Text String

Value	In case the CHARON-VAX / CHARON-PDP license connection is not present when the guest starts up, this parameter specifies how many times CHARON-VAX / CHARON-PDP will try to establish the connection and, optionally, a period of time between retries.
	
	Syntax:
	
	`set session license_key_lookup_retry = "N [, T]"`
	
	Options:
	
	• N - Number of retries to look for license keys.
	• T - Time between retries in seconds. If not specified 60 seconds are used
	
	Example 1:
	
	`set session license_key_lookup_retry = 1`
	
	If license key is not found during initial scan, do only one more attempt after 60 seconds.
	
	Example 2:
	
	`set session license_key_lookup_retry = "1,30"`
	
	Same as above but retry in 30 seconds.
	
	Example 3:
	
	`set session license_key_lookup_retry = "3,10"`
	
	If license key is not found during initial scan, do 3 more attempts waiting 10 seconds between them.
	
	Example 4:
	
	`set session license_key_lookup_retry = "5"`
	
	If license key is not found during the initial scan, do 5 more attempts waiting 60 seconds between them.

Back to Table of Contents

© Stromasys, 2017

Core Devices

Table of Contents

- CPU
  - ace_mode
- RAM
  - size
- TOY
  - container
- KW11-L and KW11-P Timers (PDP-11)
  - kw11 frequency
- ROM
  - container
- PDP-11 Boot ROM
  - ext_rom
  - ext_rom_address
- EEPROM
  - container
- Auto Boot
  - MicroVAX3100, VAXstation 4000, VAX6310 and VAX 4000
  - MicroVAX II, MicroVAX 3600/3900 and VAXserver 3600/3900
  - PDP11/93 and PDP11/94
  - VAX66x0

Back to Table of Contents

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.

Back to Table of Contents

© Stromasys, 2017

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

© Stromasys, 2017

VAX_6640	128	3584	128	128
VAX_6650	128	3584	128	128
VAX_6660	128	3584	128	128

Back to Table of Contents

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.

Back to Table of Contents

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.

© Stromasys, 2017

Back to Table of Contents

ROM

The System Flash ROM file conserves specific parameters between reboots.

container

Parameter	container
Type	Text string
Value	Specifies the name of a file in which CHARON-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. it is recommended to specify the full path to this file

Example:

```
set rom container="C:\Charon\my_virtual_system.rom"
```

Back to Table of Contents

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.

Back to Table of Contents

© Stromasys, 2017

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. it is recommended to specify the full path to this file

Example:

```
set eeprom container="C:\Charon\my_virtual_system.rom"
```

Back to Table of Contents

© Stromasys, 2017

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.

© Stromasys, 2017

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.

Back to Table of Contents

© Stromasys, 2017

Serial lines

Table of Contents

- General Description
- Console
  - rts
  - dsr
  - communication
  - line
- Serial line controllers
  - address
  - vector
  - line
  - communication
  - rts
  - dsr
  - tx_q_max_depth
- Mapping Serial line controllers to system resources
  - Types of serial line mapping
  - physical_serial_line
    - line
    - baud
    - break_on
    - stop_on
    - log
    - log_file_size
  - virtual_serial_line
    - host
    - port
    - application
    - break_on
    - stop_on
    - log
    - log_file_size
- Linking serial controller port to host connection

Back to 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.

Back to Table of Contents

© Stromasys, 2017

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

© Stromasys, 2017

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.

Back to Table of Contents

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

© Stromasys, 2017

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

© Stromasys, 2017

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

© Stromasys, 2017

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
```

Back to Table of Contents

Mapping Serial line controllers to system resources

Types of serial line mapping

Type	Function
physical_serial_line	This type of mapping associates 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
```

Back to Table of Contents

© Stromasys, 2017

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.

© Stromasys, 2017

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.

© Stromasys, 2017

Example: mapping a console line to a host physical serial line

```
load physical_serial_line/chserial OPA0
set OPA0 line="\\.\COM1"
```

Back to Table of Contents

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 OPA 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"`

© Stromasys, 2017

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"`

© Stromasys, 2017

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.

© Stromasys, 2017

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.

Back to Table of Contents

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
```

Back to Table of Contents

© Stromasys, 2017

Disks and tapes

Contents

- MSCP and TMSCP Controllers
- SCSI Controllers
- DSSI Subsystem
- CI Subsystem

© Stromasys, 2017

MSCP and TMSCP Controllers

Table of Contents

- Introduction
- RQDX3 Controller
  - address
  - max_n_of_units
  - container
  - media_type
  - geometry
  - use_io_file_buffering
- TQK50 and TUK50 Controllers
  - address
  - container
  - media_type
  - geometry
- KDM70 Controller
  - xmi_node_id
  - container
  - media_type
  - geometry
  - use_io_file_buffering
- KDB50 Storage Controller
  - vax_bi_node_id
  - container
  - media_type
  - geometry
  - use_io_file_buffering

Back to 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).

Back to Table of Contents

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>
```

© Stromasys, 2017

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.

© Stromasys, 2017

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:
	
	• **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)"`
	
	• **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>"

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.

© Stromasys, 2017

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 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.

© Stromasys, 2017

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.

Back to Table of Contents

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.

© Stromasys, 2017

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>"
	
	• **Physical SCSI tape drive connected to a CHARON host, but not recognized by Windows**
	  ◦ "\\.\Scsi<X>:<Y>:<Z>:<N>"
	
	To find the values of X,Y,Z and N use the utility "Host Device Check"

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:...
```

Back to Table of Contents

© Stromasys, 2017

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.

© Stromasys, 2017

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)"`
	
	• **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, …
	
	• **Physical tape drives not recognized by Windows**
	  • "\\.\Scsi<X>:<Y>:<Z>:<N>"
	
	  Use the utility "Host Device Check" to find the values of X,Y,Z and N
	
	• **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"

© Stromasys, 2017

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.

© Stromasys, 2017

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 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:"
```

© Stromasys, 2017

> 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.

Back to Table of Contents

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.

© Stromasys, 2017

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)"`
	
	• **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.

© Stromasys, 2017

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 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.

Back to Table of Contents

© Stromasys, 2017

SCSI Controllers

Table of Contents

- Introduction
- Mapping to host resources
  - scsi_bus
  - scsi_id
  - virtual_scsi_disk
    - container
    - media_type
    - geometry
    - use_io_file_buffering
    - removable
  - floppy_scsi_device
    - container
  - virtual_scsi_tape
    - container
    - media_type
    - geometry
  - virtual_scsi_cdrom
    - container
    - media_type
    - geometry
    - use_io_file_buffering
  - physical_scsi_device
    - container
    - media_type
    - geometry
    - removable
    - disconnect_timeout

Back to 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 "*x*K*c*t0*n*:", 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)

© Stromasys, 2017

- "*n*" stands for a particular logical unit number, LUN.

Most of the 'normal' SCSI devices have only one logical unit - 0. Therefore, under normal conditions, disks in VAX/VMS appear as DKA0 (which is really DKA000), DKA100, DKA200, ..., tapes as MKA0 (which is really MKA000), MKA100, MKA200, ...

As soon as there is a disk/tape device with LUNs 0 and 1, VMS identifies them as, for example, DKA300 and DKA301 (MKA300 and MKA301) respectively.

The boot ROM of CHARON-VAX detects SCSI devices with multiple LUNs and builds proper device names for them.

To display a list of devices on the VAX console (SRM), enter ">>> `show device`" .

This list is passed to VAX/VMS at boot time.

VAX/VMS creates devices only for logical unit 0 for each device detected in the boot ROM. To add additional logical units, use the following SYSGEN command:

```
$ MCR SYSGEN CONNECT DKxxx/NOADAPTER
```

where DKxxx (or MKxxx) stands for the correct VAX/VMS name of the logical unit to be connected. You can find its name from the SRM console using the ">>> `show scsi`" command.

This command is not boot persistent, so it must be included in the VAX/VMS "SYSTARTUP_VMS.COM" file to ensure it is executed with each startup.

Also note: that the following rules are applied for logical units.

1. Each SCSI device must implement logical unit 0.
2. A SCSI device must implement all logical unit numbers between the highest and the lowest numbers implemented.

Empty disk images can be created with the "MkDisk" utility.

CHARON-VAX is able to boot from disk images of any VAX/VMS version, starting with 4.5 or higher for MicroVAX II or VAX 3600 and VMS 5.5-2 or higher for the VAX4000.

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

Back to Table of Contents

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"

© Stromasys, 2017

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.cfg
```

"kzdda.cfg" loads the second SCSI adapter.

OpenVMS version 5.5-2H4, or above, is required to use the "pkb" controller.

Back to Table of Contents

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:

© Stromasys, 2017

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").
	
	• **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)"`
	
	• **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.

© Stromasys, 2017

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
```

Back to Table of Contents

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:"
```

Back to Table of Contents

© Stromasys, 2017

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").

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.

© Stromasys, 2017

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"
```

Back to Table of Contents

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.

© Stromasys, 2017

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**: `set pka_0 use_io_file_buffering[0]=true`

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".

Back to Table of Contents

© Stromasys, 2017

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, …
	
	• **Physical tape drives and other SCSI devices not recognized by Windows**
	  • "\\.\Scsi<X>:<Y>:<Z>:<N>"
	
	Use the utility "Host Device Check" to find the values of X,Y,Z and N
	
	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.

© Stromasys, 2017

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 `set pka_0 disconnect_timeout[1] = 0x400000`

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".

Back to Table of Contents

© Stromasys, 2017

DSSI Subsystem

Table of Contents

- Introduction
- SHAC host adapter
  - port
  - host
  - scs_node_name
  - scs_system_id
  - mscp_allocation_class
  - container
  - media_type
  - geometry
  - use_io_file_buffering
- HSD50 storage controller
  - dssi_host
  - dssi_node_id
  - scs_node_name
  - scs_system_id
  - mscp_allocation_class
  - container
  - media_type
  - geometry
  - use_io_file_buffering

Back to 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.

Back to Table of Contents

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.

© Stromasys, 2017

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.

© Stromasys, 2017

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.

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)"`
	
	• **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, …
	
	• **Physical tape drives not recognized by Windows**
	  • "\\.\Scsi<X>:<Y>:<Z>:<N>"
	
	Use the utility "Host Device Check" to find the values of X,Y,Z and N
	
	• **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.

© Stromasys, 2017

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.

© Stromasys, 2017

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 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 dDevice                  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
```

Back to Table of Contents

© Stromasys, 2017

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).

© Stromasys, 2017

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)"`
	• **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, …
	• **Physical tape drives not recognized by Windows**
	  • "\\.\Scsi<X>:<Y>:<Z>:<N>"Use the utility "Host Device Check" to find the values of X,Y,Z and N
	• **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]=OPA0load 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:

© Stromasys, 2017

```
>>> BOOT DUA0
```

After logging into VMS, the “SHOW DEVICE D” and "SHOW DEVICE M" command displays the following:

```
$ show devices dDevice                  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 mDevice                  Device           Error    Volume         Free  Trans Mnt
 Name                    Status          Count     Label        Blocks Count Cnt
TAPES$MUA3:             Online              0
```

Back to Table of Contents

© Stromasys, 2017

CI Subsystem

Table of Contents

- Introduction
- CIXCD host adapter
  - port
  - host
  - xmi_node_id
  - ci_node_id
- HSJ50 storage controller
  - ci_host
  - ci_node_id
  - scs_node_name
  - scs_system_id
  - mscp_allocation_class
  - container
  - media_type
  - geometry
  - use_io_file_buffering

Back to 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.

Back to Table of Contents

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:

© Stromasys, 2017

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:

© Stromasys, 2017

```
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"
```

Back to Table of Contents

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).

© Stromasys, 2017

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).

© Stromasys, 2017

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)"`
	
	• **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, …
	
	• **Physical tape drives not recognized by Windows**
	  • "\\.\Scsi<X>:<Y>:<Z>:<N>"
	
	Use the utility "Host Device Check" to find the values of X,Y,Z and N
	
	• **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

© Stromasys, 2017

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.

© Stromasys, 2017

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`

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).

© Stromasys, 2017

Then via the storage controller with CI node id B (decimal 11), DU0 is reached (defined as container[0]) and the boot command is executed for the associated file on the host system.

After logging into VMS, the “SHOW DEVICE” command displays the following:

```
$ show devices

Device                  Device           Error    Volume         Free  Trans Mnt
 Name                   Status           Count     Label        Blocks Count Cnt
DISKS$DUA0:             Mounted              0     DSSI01        32022   147   1
DISKS$DUA1:             Online               0

Device                  Device           Error
 Name                   Status           Count
OPA0:                   Online               0
FTA0:                   Offline              0

Device                  Device           Error
 Name                   Status           Count
PAA0:                   Online               0
```

When a tape or disk image connected to an emulated HSJ50 controller is dismounted in VAX/VMS, it is disconnected from CHARON-VAX and can be manipulated. It may even be replaced with a different disk image if it keeps the same name. This capability may be useful when designing back-up and restore procedures.

Back to Table of Contents

© Stromasys, 2017

Networking

Table of Contents

- Introduction
- SGEC Ethernet Controller
  - interface
  - station_address
  - rx_fifo_size
- DEQNA / DESQA / DELQA Ethernet Controller
  - address
  - interface
  - station_address
  - rx_fifo_size
- DEMNA Ethernet Adapter
  - xmi_node_id
  - interface
  - station_address
  - rx_fifo_size
- DEBNI Ethernet Adapter
  - vax_bi_node_id
  - interface
  - station_address
  - rx_fifo_size
- PMAD-AA TurboChannel Ethernet Adapter
  - interface
  - station_address
  - rx_fifo_size
- Packet Port
  - interface
  - port_enable_mac_addr_change
  - port_retry_on_tx
  - port_pending_rx_number
  - port_pending_tx_number
  - suspend_msg_on_mac_change
  - log
  - legacy_mode
- Supported Ethernet Q-bus Adapters for DECnet OSI (DECnet Plus)
- Ethernet Q-bus Adapter and Cluster configuration rules

Back to Table of Contents

Introduction

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

© Stromasys, 2017

interface name.

Example:

```
set NIC interface = NDIS
```

Back to Table of Contents

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"
```

© Stromasys, 2017

Back to Table of Contents

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

© Stromasys, 2017

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"
```

Back to Table of Contents

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:

© Stromasys, 2017

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.

© Stromasys, 2017

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"
```

Back to Table of Contents

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"`

© Stromasys, 2017

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"
```

Back to Table of Contents

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

© Stromasys, 2017

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"
```

Back to Table of Contents

© Stromasys, 2017

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`

© Stromasys, 2017

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`

© Stromasys, 2017

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 logs Recv and Xmit packets at the emulated port layer.
	
	If an existing directory is specified, CHARON automatically enables creation of individual log files, one for each session using the same scheme as used for the generation of the rotating log files. If the "log" parameter is omitted, CHARON does not create log.
	
	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

© Stromasys, 2017

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.

Back to Table of Contents

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.

Back to Table of Contents

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.

Back to Table of Contents

© Stromasys, 2017

Specific Controllers

Table of Contents

- LPV11 Line Printer Controller
  - address
  - vector
  - parameters
- DUMMY_VCB02 device

Back to Table of Contents

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 from the STROMASYS web site.

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.

© Stromasys, 2017

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\'"
```

Back to Table of Contents

© Stromasys, 2017

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.

Back to Table of Contents

© Stromasys, 2017

Sample configuration files

Contents

- PDP-11/93 configuration file
- VAX 4000 Model 108 configuration file
- VAX 6310 configuration file
- VAX 6610 configuration file

© Stromasys, 2017

PDP-11/93 configuration file

```
#
# Copyright (C) 1999-2014 STROMASYS
# All rights reserved.
#
# The software contained on this media is proprietary to and embodies
# the confidential technology of STROMASYS. Posession, use, duplication,
# or dissemination of the software and media is authorized only pursuant
# to a valid written license from STROMASYS.
#
#============================================================================
#
# Sample configuration file for PDP-11/93 machines.
#
#----------------------------------------------------------------------------

set session hw_model = PDP1193

#============================================================================
#
# Select name of the instance to differentiate it among other instances
# runnig on the same host.
#
#----------------------------------------------------------------------------

#set session configuration_name = PDP1193

#============================================================================
#
# Disable rotating LOG and enable single file LOG. Select either appending or
# overwriting it on every instance start. Then specify desired log file name
# and path to it.
#
#----------------------------------------------------------------------------

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

#============================================================================
#
# Select connection for the console serial line TT0.
#
#----------------------------------------------------------------------------

#load physical_serial_line TT0 line = "\\.\COM<N>"
#load virtual_serial_line TT0 port = 10003
load virtual_serial_line TT0 port = 10003 application = "putty -load TT0 -P 10003"
set UART line = TT0

#============================================================================
#
# Uncomment to enable emulation of RQDX3 disk controller.
#
#----------------------------------------------------------------------------

#load RQDX3 DU

#============================================================================
#
# Uncomment to connect the emulator's DU0 to the disk image.
#
#----------------------------------------------------------------------------
```

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

#================================================================================
#
# Uncomment to connect the emulator's DU1 to host's disk drive.
#
#--------------------------------------------------------------------------------

#set DU container[1] = "\\.\PhysicalDrive0"
#set DU container[1] = "\\.\PhysicalDrive<N>"

#================================================================================
#
# Uncomment to connect the emulator's DU2 to host's CD/DVD-ROM drive or ISO
# disk image.
#
#--------------------------------------------------------------------------------

#set DU container[2] = "\\.\CdRom0"
#set DU container[2] = "\\.\CdRom<N>"
#set DU container[2] = "<file-name>.iso"

#================================================================================
#
# Uncomment to connect the emulator's DU3 to host's 3.5" FDD.
#
# Either "A:" or "B:" may be used in container specification.
#
#--------------------------------------------------------------------------------

#set DU container[3] = "\\.\A:"

#================================================================================
#
# Boot from DU0 MSCP disk.
#
#--------------------------------------------------------------------------------

#set cpu_0 auto_boot = "DU0"

#================================================================================
#
# Uncomment to enable emulation of TQK50 tape controller.
#
#--------------------------------------------------------------------------------

#load TQK50 MU

#================================================================================
#
# Uncomment to connect the emulator's MU0 to the tape image.
#
#--------------------------------------------------------------------------------

#set MU container[0] = "<file-name>.vtape"

#================================================================================
#
# Uncomment to connect the emulator's MU1 to host's tape drive.
#
#--------------------------------------------------------------------------------

#set MU container[1] = "\\.\Tape0"
#set MU container[1] = "\\.\Tape<N>"

#================================================================================
#
# Load optional DEQNA QBUS Ethernet Adapter (QNA).
#
#--------------------------------------------------------------------------------

#load DEQNA QNA interface = QNA0
#load packet_port QNA0 interface = (disabled)
#set QNA0 interface = "connection:<connection-name>"

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

© Stromasys, 2017

Back to Top

© Stromasys, 2017

VAX 4000 Model 108 configuration file

```
#
# Copyright (C) 1999-2014 STROMASYS
# All rights reserved.
#
# The software contained on this media is proprietary to and embodies
# the confidential technology of STROMASYS. Possession, use, duplication,
# or dissemination of the software and media is authorized only pursuant
# to a valid written license from STROMASYS.
#
#=============================================================================
#
# Sample configuration file for 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.
#
#-----------------------------------------------------------------------------

#set session configuration_name = VAX_4000_Model_108

#=============================================================================
#
# 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 GKA200 to an unknown SCSI device.
#
#------------------------------------------------------------------------------

#load physical_scsi_device pka_2 scsi_bus=pka scsi_id=2
#set pka_2 container="\\.\ScsiN:X:Y:Z"

#==============================================================================
#
# 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.
#
#------------------------------------------------------------------------------
```

© Stromasys, 2017

```
#include kzdda.cfg

#================================================================================
#
# 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"
```

© Stromasys, 2017

```
#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 ##################################
```

Back to Top

© Stromasys, 2017

VAX 6310 configuration file

```
#
# Copyright (C) 1999-2014 STROMASYS
# All rights reserved.
#
# The software contained on this media is proprietary to and embodies
# the confidential technology of STROMASYS. Possession, use, duplication,
# or dissemination of the software and media is authorized only pursuant
# to a valid written license from STROMASYS.
#
#===============================================================================
#
# Sample configuration file for 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.
#
#-------------------------------------------------------------------------------

#set session configuration_name = VAX_6310

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

Back to Top

© Stromasys, 2017

VAX 6610 configuration file

```
#
# Copyright (C) 1999-2014 STROMASYS
# All rights reserved.
#
# The software contained on this media is proprietary to and embodies
# the confidential technology of STROMASYS. Possession, use, duplication,
# or dissemination of the software and media is authorized only pursuant
# to a valid written license from STROMASYS.
#

#
# Sample configuration file for:
#
# * Charon-VAX/6610
#
# Emulator of the VAX model:
#
# * VAX 6610
#
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.
#
#--------------------------------------------------------------------------------

#set session configuration_name = VAX_6610

#================================================================================
#
# 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

#
# TOY date/time, boot options, ...
#

set toy container="vx6k610.dat"

#
# EEPROM settings, e.g. default boot drive
#

set eeprom container="vx6k610.rom"

#
#
# Specify the size of RAM (default 32MB). 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, ... 3584,

# 32bit systems only: Please note, that to be able to run the ~2Gb
```

```
# (2048) of simulated memory you should enable /3GB switch
#

set ram size=512
#set ram size=768
#set ram size=1024
#set ram size=2048
#set ram size=3584

#
# Now assign built-in console serial line. Currently the emulator
# offers two possible ways of using built-in serial lines. First of
# them is to attach a third party terminal emulator (virtual_serial_line).
# The second is connection to COM ports (via physical_serial_line).
#
# Once the way of the connection is defined corresponding mapping
# to the host resources can be performed.
#
# Connection to a third party terminal emulator:
#

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"

#
# Connection to a physical COM port of the host:
#

# load physical_serial_line/chserial OPA0
# set OPA0 line="COM1:"

#
# Load the MSCP controller with the name DUA
#

load KDM70/KDM70 DUA xmi_node_id=11

#
# Map the storage units for loaded MSCP controller DUA.
#
# In VMS, the DUA0 will correspond to the 'DUA container[0]', etc...
#
# Examples:
#
# set DUA container[0]="C:\Disk-Images\charon-6000\dua0.vdisk"
# set DUA container[1]="\\.\PhysicalDrive1"
# set DUA container[2]="\\.\CdRom0"
# set DUA container[3]="\\.\A:"
#
# It is also possible to map a tape drive identified by the host
# operating system to the DUA controller.
#
# Example:
#
# set DUA container[3]="\\.\Tape0"
#

#set DUA container[0]="..."

#
# 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 controller to the CIXCD adapter PAA.
#

#load HSJ50 PUA ci_node_id=0x0B mscp_allocation_class=1
```

© Stromasys, 2017

```
#
# Map the storage units for the loaded HSJ50 controller PUA.
#
# In VMS, the PUA will correspond to the 'DUA container[0]', etc...
#
# Examples:
#
# set PUA container[0]="C:\Disk-Images\charon-6000\dua0.vdisk"
# set PUA container[1]="\\.\PhysicalDrive1"
# set PUA container[2]="\\.\CdRom0"
# set PUA container[3]="\\.\A:"
#
# It is also possible to map a tape drive identified by the host
# operating system to the PUA controller.
#
# Example:
#
# set PUA container[3]="\\.\Tape0"
#

#set PUA container[0]="..."

#
# IMPORTANT NOTE:
#
# Make sure that DUA and PUA do not conflict in the VMS environment, since
# the disks on those controllers have the same name "DUA[x]" preceded by
# their allocation class (for the HSJ50 controller it is set in the
# configuration file). Assign the allocation class in such a way to avoid
# name conflicts or enable only either DUA or PUA.
#
# See CHARON-66x0 User Guide for more details.
#

#
# Load Ethernet Controllers
#
# 2 Ethernet controllers are loaded in this sample configuration file: EXA
# and EXB. Both of them are connected to corresponding network interfaces:
# EXA_0 and EXB_0. The initial settings for the interfaces are "disabled".
# The exact mapping to the host ethernet interfaces will be defined later
# in this configuration file.
#

load DEMNA/demna EXA xmi_node_id=13 interface=EXA_0
load packet_port/chnetwrk EXA_0 interface="(disabled)"

load DEMNA/demna EXB xmi_node_id=14 interface=EXB_0
load packet_port/chnetwrk EXB_0 interface="(disabled)"

#
# Map the Network connections.
#
# Each connection is instantiated by the two objects. One is DEMNA controler
# emulator EXn where n is A or B, and second is packet port EXn_0 where the
# n is A or B.
#
# In the emulated VMS, the network devices EXn will correspond to the
# EXn, EXn_0 pair in the configuration file.
#
# To assign the spesific permanent address to the desired EXn controller
# ucomment the corresponding line below.
#
# Example:
#
# set EXA station_address="00-63-50-1A-A3-1E"
#

#set EXA station_address="..."
#set EXB station_address="..."
```

© Stromasys, 2017

```
#
# To connect the corresponding adapter to the external device, uncomment
# the line below and put the valid interface definition.
#
# Example:
#
# set EXA_0 interface="Cabletron DE500B PCI Fast Ethernet Adapter (21143-PC)"
#

#set EXA_0 interface="connection:<connection-name>"
#set EXB_0 interface="connection:<connection-name>"


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

Back to Top

© Stromasys, 2017

CHARON-VAX and CHARON-PDP for Windows deinstallation

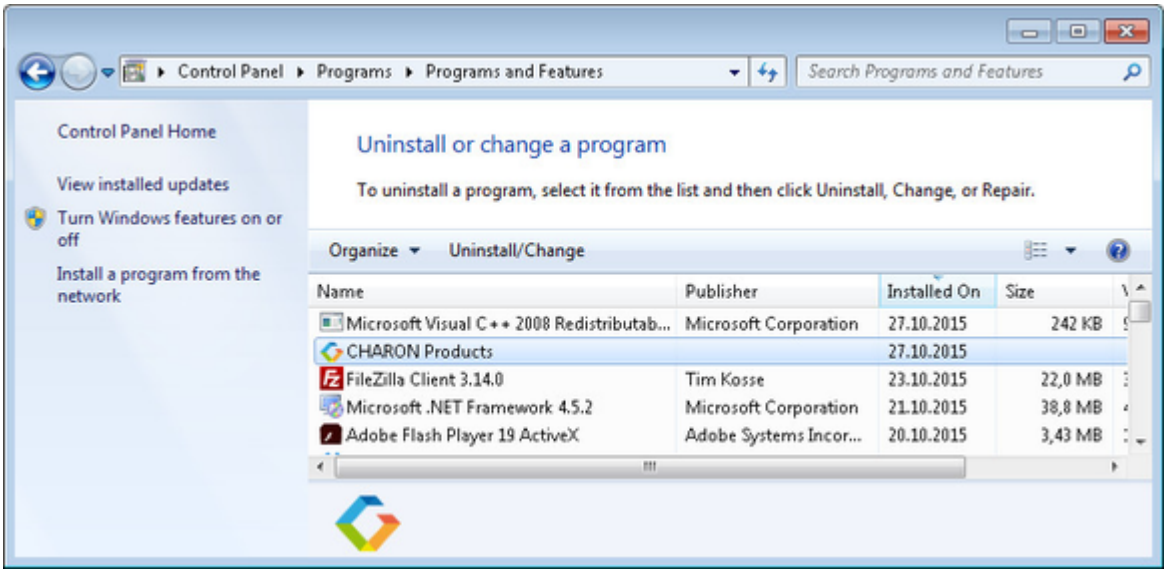
Deinstallation procedure

Perform a clean shutdown of the running guests, stop all running CHARON-VAX / CHARON-PDP instances and remove the corresponding CHARON-VAX / CHARON-PDP services for the emulated VAX/PDP11 models to be removed.
In case of a complete deinstallation:

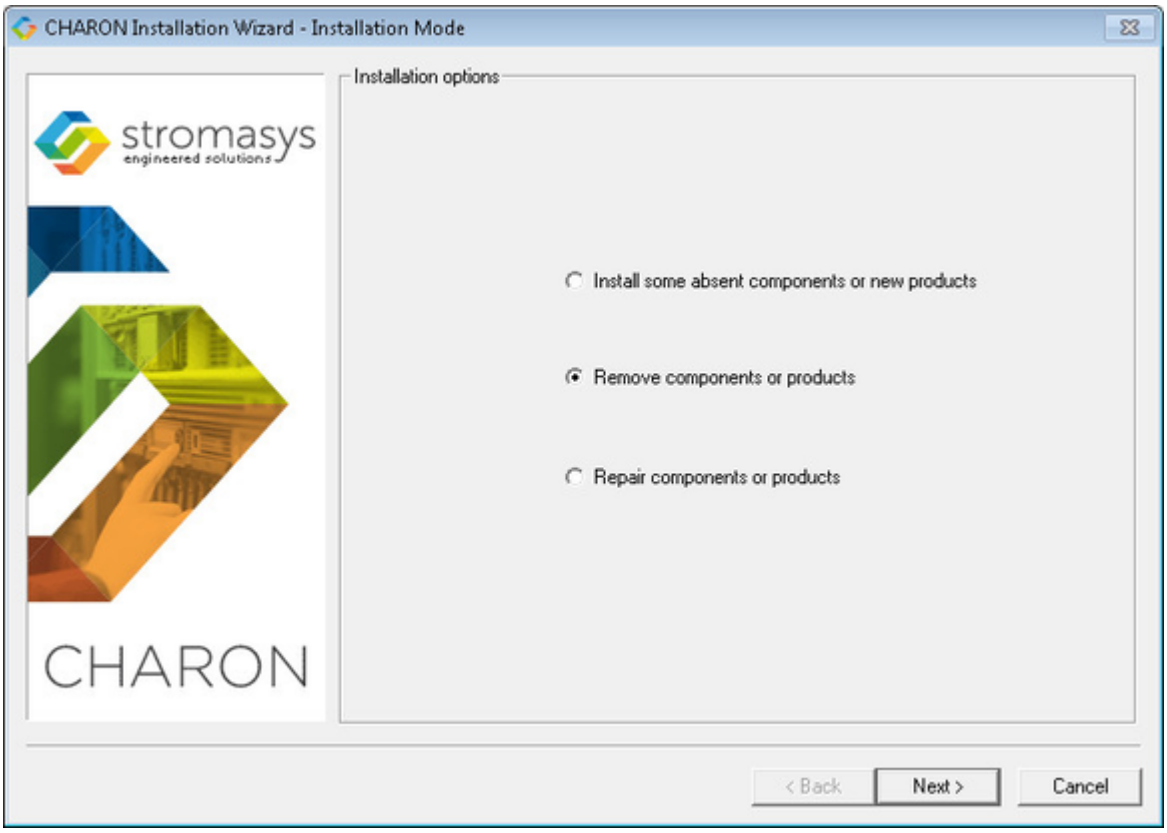
- Remove all CHARON-VAX / CHARON-PDP services.
- Using the "Network Control Center" utility, release all the network interfaces back to the CHARON host.
- Remove any specific CHARON drivers installed manually (if any).

Open up the "Control Panel", select "Uninstall a program":

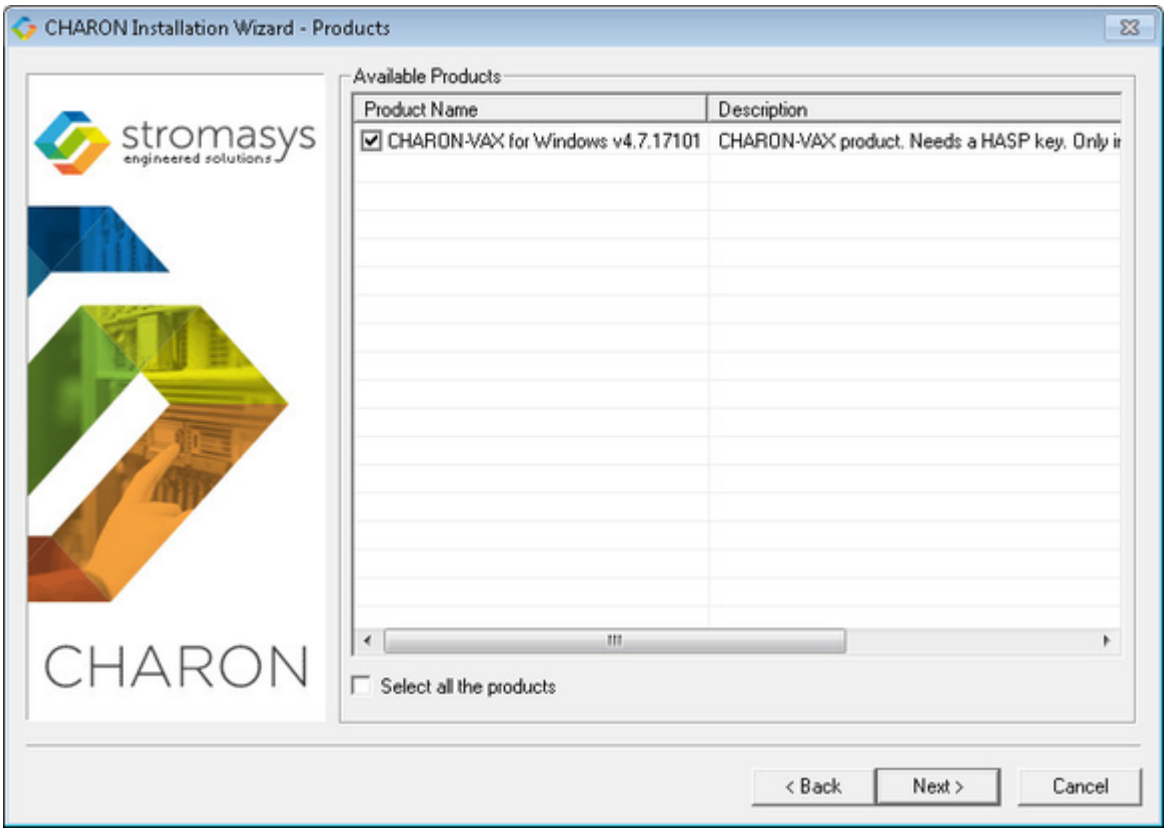
In the list of applications, double click on "CHARON Products":



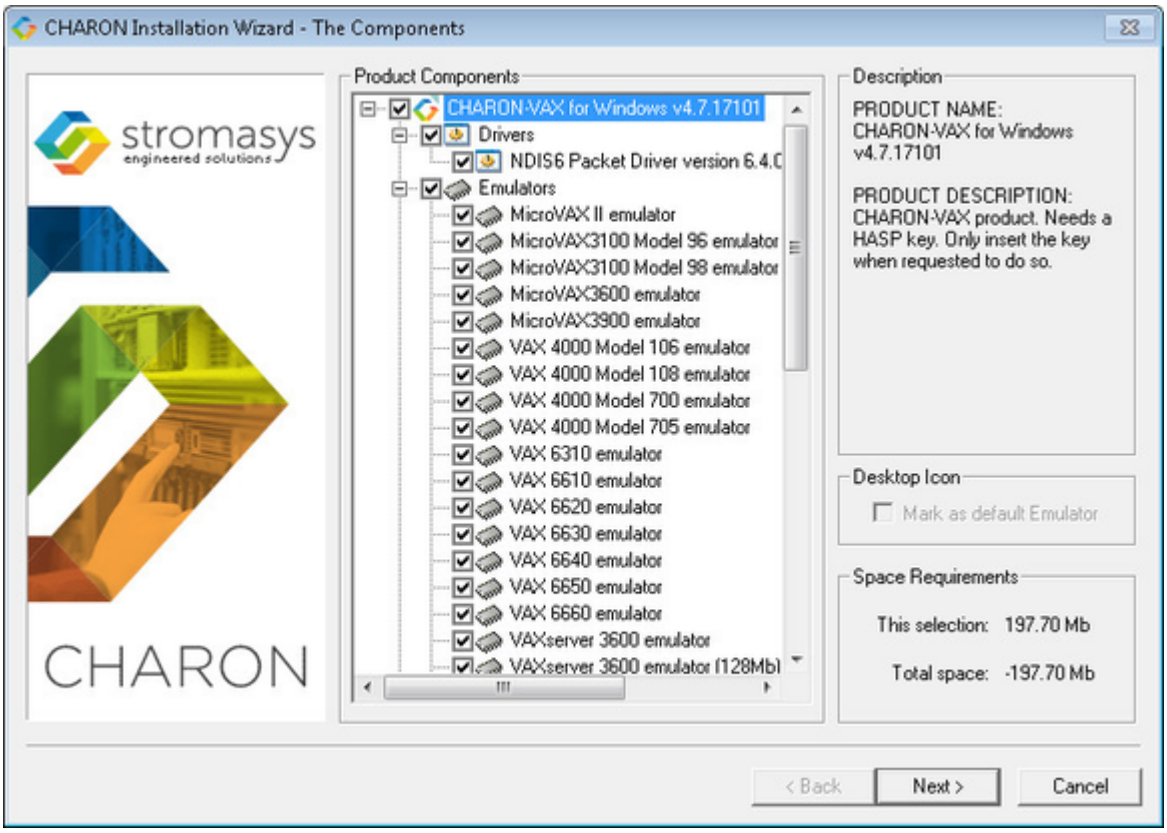
In the appearing CHARON installation dialog, select "Remove components or products" and press "Next":



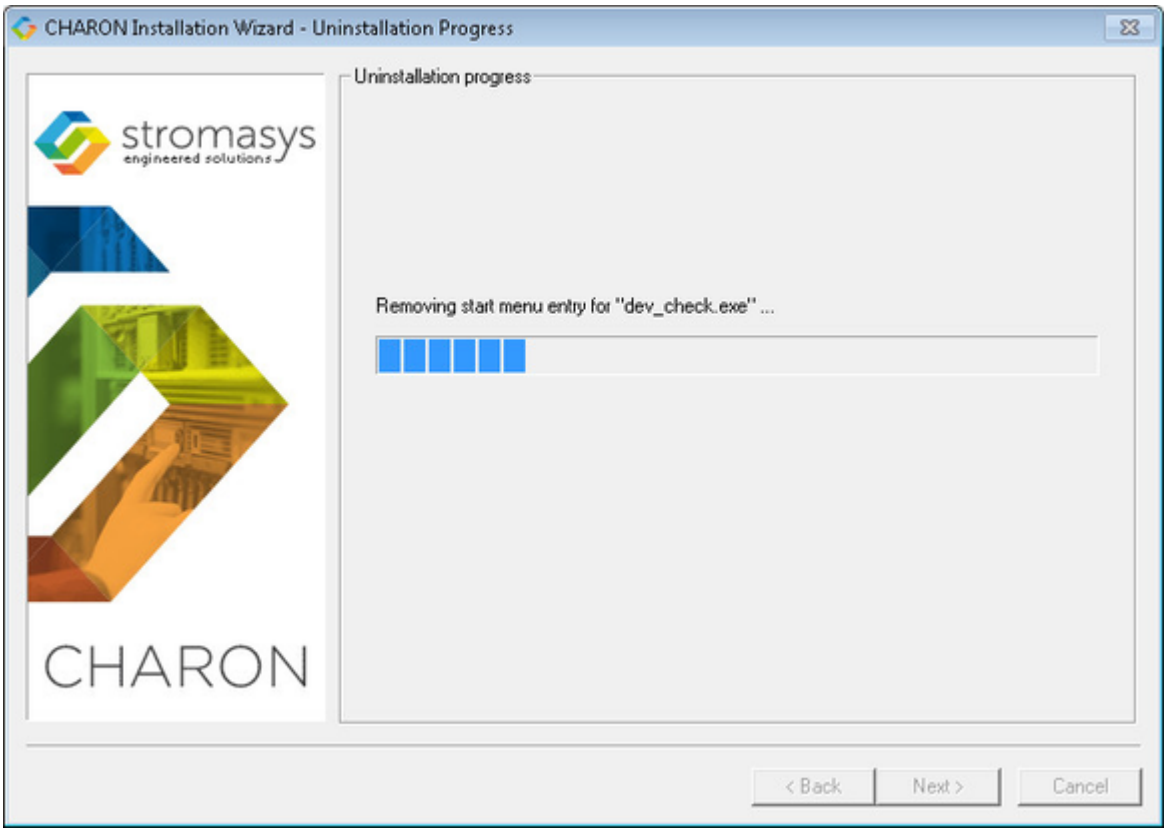
In the following dialog, select the CHARON product to be removed (or changed). Check "Select all the products" for deinstallation (or changing) of all installed CHARON products.
Press "Next":



Select the particular components to be deinstalled. Selection of all the components means that entire CHARON product will be removed.
Press "Next".

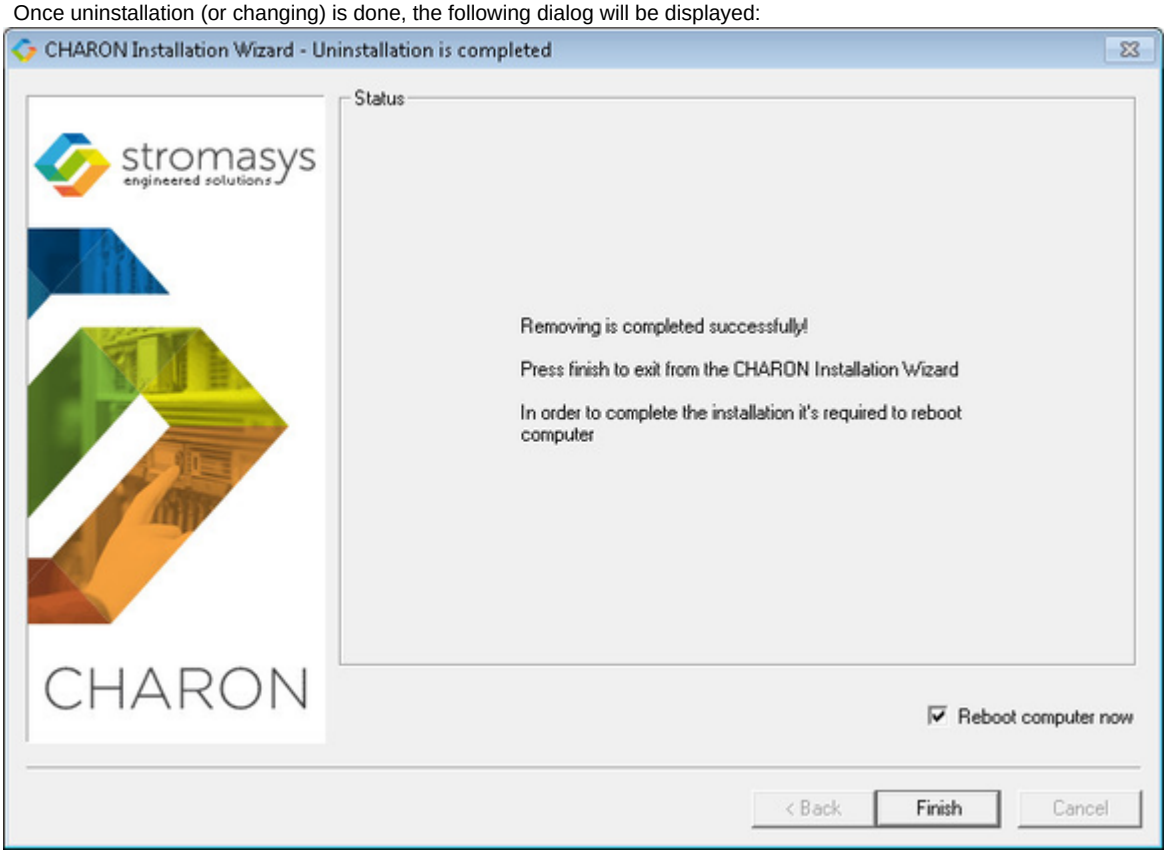


The uninstallation process will begin:



© Stromasys, 2017

Once uninstallation (or changing) is done, the following dialog will be displayed:



Press "Finish". The CHARON host will reboot automatically.

Back to Top

© Stromasys, 2017