

CHARON-AXP for Windows, version 4.5

CHARON-AXP for Windows, version 4.5

Publication date 02 June 2014

Table of Contents

1. Preface	1
1.1. Ownership Notice	1
1.2. Trademarks	1
1.3. Life support applications	1
1.4. End User License Agreement	1
1.4.1. Grant of License	1
1.4.2. Ownership of Software	2
1.4.3. Access and Transfers	2
1.4.4. Term	2
1.4.5. Limited Warranty	2
1.4.6. Intellectual Property Infringement	3
1.4.7. Export	3
1.4.8. Applicable Law; Claims and Disputes	3
2. Overview	5
2.1. General Description	5
2.2. CHARON-AXP User Guide Structure	5
2.3. CHARON-AXP hardware compatibility	7
3. CHARON-AXP hosting and performance	8
3.1. Host operating system requirements	8
3.2. Host operating system requirements	8
3.2.1. Common requirements	8
3.2.1.1. CPU selection	8
3.2.1.2. Host system hardware platform recommendations	9
3.2.1.3. Host system memory	9
3.2.1.4. Disk storage	9
3.2.2. Windows specific requirements	9
3.2.2.1. Ethernet adapters	9
3.2.2.2. Other host system requirements	10
3.3. General performance considerations	10
3.3.1. AXP CPU performance	11
3.3.2. Disk I/O subsystem	11
3.3.3. Network connections	11
3.3.4. Enhancing virtualization layer reliability	11
4. CHARON licensing	13
4.1. General description	13
4.1.1. General parameters	13
4.1.2. Products parameters	14
4.1.3. Optional parameters	14
4.2. CHARON licensing models	15
4.2.1. Licensing by usage of locally installed Sentinel HASP keys	15
4.2.2. Licensing by usage of the specific Network Sentinel HASP keys	15
4.2.3. Licensing by software license (SL)	15
4.3. Multiple licenses configuration	16
4.4. License installation	16
4.4.1. Installation from scratch	16
4.4.2. Replacement of currently installed Sentinel run-time to other one	17
4.4.3. Installation of CHARON Software License	17
4.5. License management	17
4.5.1. Sentinel Admin Control Center	18
4.5.1.1. General Description	18
4.5.1.2. Disabling remote keys access via Sentinel Admin Control Center	18
4.5.2. License management specifics	19
4.6. Switch to backup key in CHARON	20
4.7. Software Licenses Transfer	20

4.8. License Deinstallation	21
4.9. Backup license keys	22
4.10. Important Notes	22
5. Installing CHARON for Windows	23
5.1. License key driver installation	23
5.2. Installing files and components	24
5.3. Configuring network adapters	25
5.4. Installation of Software License (optional)	25
5.5. Configuring CHARON virtualization layer	26
5.6. Uninstallation	26
5.7. Default installation	26
5.7.1. File locations	26
5.7.2. Start Menu	27
5.7.3. Desktop icon	27
6. Configuring Virtual HP Alpha	28
6.1. The HP Alpha system architecture	28
6.2. The configuration command syntax	29
6.3. The virtual AXP models specifics	29
6.3.1. AlphaServer 400 (DECchip 21072, 3 PCI slots)	30
6.3.2. AlphaServer 800 (DECchip 21172, 4 PCI slots)	30
6.3.3. AlphaServer 1000 (DECchip 21072, 3 PCI slots)	30
6.3.4. AlphaServer 1000A (DECchip 21072, 7 PCI slots)	31
6.3.5. AlphaServer 1200 (1 IOD, 6 PCI slots)	31
6.3.6. AlphaServer 2000 (T2, 3 PCI slots)	32
6.3.7. AlphaServer 2100 (T2, 3 PCI slots)	32
6.3.8. AlphaServer 4000 (2 IODs, 16 PCI slots)	33
6.3.9. AlphaServer 4100 (1 IOD, 8 PCI slots)	33
6.3.10. AlphaServer DS10L (1 PCI bus, 4 PCI slot)	34
6.3.11. AlphaServer DS15 (2 Pchips, 4 PCI slots)	34
6.3.12. AlphaServer DS20 (2 Pchips, 6 PCI slots)	35
6.3.13. AlphaServer DS25 (2 Pchips, 6 PCI slots)	36
6.3.14. AlphaServer ES40 (2 Pchips, 10 PCI slots)	37
6.3.15. AlphaServer ES45 (2 Pchips, 10 PCI slots)	37
6.3.16. AlphaServer GS80 (2 QBBs, 8 PCI busses, 27 PCI slots)	38
6.3.17. AlphaServer GS160 (4 QBBs, 16 PCI busses, 55 PCI slots)	39
6.3.18. AlphaServer GS320 (8 QBBs, 32 PCI busses, 111 PCI slots)	41
6.4. Multi instance support	45
6.4.1. General description	45
6.4.2. Running several instances of CHARON	46
6.5. General configuration parameters	47
6.5.1. Common parameters	47
6.5.2. Specific configuration parameters	50
6.5.3. Examples	51
6.6. Console interface	51
6.6.1. Types of serial line emulation	52
6.6.2. "physical_serial_line" parameters	52
6.6.3. "virtual_serial_line" parameters	53
6.7. Specifying emulated memory	56
6.7.1. Syntax	56
6.7.2. Parameters of emulated RAM for various hardware models of virtual HP Alpha system	56
6.8. System time and date	57
6.9. Virtual HP Alpha SRM console environment	58
6.9.1. Firmware and console environment parameters	58
6.10. CPU Architecture	59
6.11. Virtual HP Alpha interval timer	60
6.12. Data storage in the virtualization layer	61
6.12.1. Types of data storage	61

6.12.1.1. Physical disks and disk images	61
6.12.1.2. Physical tapes and tape images	61
6.12.2. Virtual Acer Labs 1543C IDE/ATAPI controller	62
6.12.3. Virtual KZPBA PCI SCSI adapter	62
6.12.3.1. Attaching virtual KZPBA PCI SCSI Adapter to virtual system	62
6.12.3.2. Configuring virtual KZPBA PCI SCSI Adapter	63
6.12.3.2.1. KZPBA general parameters	63
6.12.3.2.2. KZPBA mapping to system resources	67
6.12.4. Virtual DEC-KGPSA-CA (EMULEX LP8000) PCI Fibre Channel adapter	69
6.12.4.1. Attaching virtual KGPSA PCI Fibre Channel Adapter to virtual system	70
6.12.4.2. Configuring virtual KGPSA PCI Fibre Channel Adapter in Fabric virtualization mode	70
6.12.4.2.1. KGPSA general parameters	70
6.12.4.2.2. KGPSA mapping to system resources	74
6.12.4.3. Configuring virtual KGPSA PCI Fibre Channel Adapter for CHARON PCI Pass Through	75
6.12.4.3.1. Defining parameters for the "host_bus_location"	75
6.12.4.3.2. Supported physical EMULEX LightPulse PCI/PCI-X/PCIe FC adapters	77
6.13. Virtual PCI Ethernet controllers	78
6.13.1. Virtual DE435, DE450, DE500AA and DE500BA network adapters	78
6.13.1.1. Attaching virtual DE435, DE450, DE500AA and DE500BA to virtual system	78
6.13.1.2. Configuring virtual DE435, DE450, DE500AA and DE500BA network adapters	79
6.13.2. Virtual DEFPA PCI FDDI adapter	82
6.13.2.1. Attaching virtual DEFPA PCI FDDI Adapter to virtual system	82
6.13.2.2. Configuring virtual DEFPA PCI FDDI Adapter	82
6.13.2.3. Defining parameters for the "host_bus_location"	85
6.13.3. NDIS5/NDIS6 CHARON Packet Port	87
6.13.3.1. Attaching NDIS5/NDIS6 CHARON Packet Port to virtual system	87
6.13.3.2. Configuring NDIS5/NDIS6 CHARON Packet Port	87
6.13.3.2.1. CHARON Packet Port general parameters	87
6.13.3.2.2. NDIS5/NDIS6 CHARON Packet Port mapping	88
6.14. Serial lines support in the virtualization layer	90
6.14.1. Virtual PBXDA-xx series PCI serial adapters based on DIGI AccelePort adapters	90
6.14.1.1. Attaching virtual PBXDA-xx series PCI serial Adapter to virtual system	91
6.14.1.2. Configuring virtual DEFPA PCI FDDI Adapter	91
6.14.1.3. Defining parameters for the "host_bus_location"	93
6.14.1.4. Compatibility	95
7. Operating CHARON	96
8. CHARON-AXP Utilities	97
8.1. Overview	97
8.2. CHARON Launcher	97
8.2.1. Overview	97
8.2.2. Installing CHARON as service	98
8.3. CHARON Service Manager	100
8.3.1. Overview	100
8.3.2. Interface description	100
8.3.3. Controlling CHARON Service Manager via system tray menu	103

8.4. CHARON Network Control Center	103
8.4.1. Overview	103
8.4.2. Install and upgrade CHARON network driver	104
8.4.3. Configure NIC for CHARON	105
8.4.4. Troubleshoot NIC dedicated to CHARON	107
8.4.5. CHARON Network Control Center log	110
8.5. MKDISK utility	111
8.6. "MkDskCmd" utility	112
8.6.1. Transferring disk images	113
8.7. License Update Service	114
8.8. HASP View Utility	117
8.8.1. Resolving problems with multiple licenses installed	118
8.9. Host Device Check utility	119
8.9.1. An alternative way to find out SCSI configuration details	119
8.10. MTD Utility	121
8.11. HOSTprint utility	122
8.12. "idle" utility	123
A. Installing and transferring an original host software to CHARON	125
A.1. Using Local Area Network	125
A.2. Using a physical disk drive	125
A.3. Using a tape	125
B. Configuration file examples	126
B.1. Virtual HP AlphaServer ES40 configuration template. (e.g. <i>es40.cfg</i>)	126
C. Specification of "system_name" parameter	132
D. Required Windows Standard Services for CHARON	139
D.1. Services disabling	139
D.1.1. In all cases	139
D.1.2. Windows services that may be disabled	140
D.1.3. Windows services that may be disabled optionally	141
D.2. Windows services that must NOT be disabled	141
D.3. Exceptions	143

Chapter 1. Preface

1.1. Ownership Notice

Stromasys SA, Geneva, Switzerland, owns all rights, including proprietary rights, copyrights, trademarks, and world-wide distribution rights to a methodology for the execution of HP Alpha applications and system software by means of a software virtualization layer, henceforth referred to as CHARON-AXP. The right to use CHARON-AXP software is governed by a license allowing the execution of the software on a single computer system. The CHARON-AXP license does not transfer ownership of the CHARON-AXP encrypted binary executable, nor does it provide any rights to decrypt, view, analyze, copy or reverse engineer the CHARON-AXP binary or source code. Possession and use of the software described in this publication is authorized only pursuant to a valid license.

Stromasys makes no representations that the use of the CHARON-AXP software as described in this publication will not infringe on existing or future patent rights, nor do the descriptions contained in this publication imply the granting of licenses to make, use, or sell equipment or software in accordance with the description.

1.2. Trademarks

The CHARON name with logo is a trademark of Stromasys. AXP, XMI, VMS and OpenVMS are trademarks of the Hewlett-Packard Company. Pentium, Xeon are registered trademarks in the United States and other countries, licensed exclusively through Intel Corporation, USA. Athlon and Opteron are registered trademarks of Advanced Micro Devices. All other trademarks and registered trademarks are the property of their respective holders.

1.3. Life support applications

The CHARON products of Stromasys are not designed for use in systems where malfunction of a CHARON product can reasonably be expected to result in a personal injury. Stromasys' customers using or selling our CHARON products for use in such applications do so at their own risk and agree to fully indemnify Stromasys for any damages resulting from such improper use or sale.

1.4. End User License Agreement

This is an agreement between Stromasys SA of Geneva, Switzerland, Licensor, and you, the end user, Licensee;

YOU ACKNOWLEDGE THAT YOU HAVE READ THIS SOFTWARE LICENSE CONTRACT AND LIMITED WARRANTY, UNDERSTAND IT, AND AGREE TO BE BOUND BY ITS TERMS AND CONDITIONS.

1.4.1. Grant of License

The Licensor grants to the Licensee, a non-exclusive right to use the licensed functionality of the Software Product (hereinafter the "SOFTWARE") in accordance with the terms contained in this License. Unless the contrary is specifically indicated in the product specification, this License permits the Licensee to run a single instance of the SOFTWARE on the computer.

1.4.2. Ownership of Software

STROMASYS retains the copyright, title and ownership of the SOFTWARE and the written materials regardless of the form or media in or on which the original and other copies may exist.

1.4.3. Access and Transfers

The Licensee defines who may access the licensed SOFTWARE. The Licensee is permitted to transfer the SOFTWARE from one of its computers to another one of its computers provided the SOFTWARE is transferred without modification and the other computer's configuration is appropriate for the SOFTWARE as per its Software Product Description. The Licensee is not permitted to transfer the SOFTWARE to a third party (a person or a company).

1.4.4. Term

If the term is limited in time, this License is valid as long as the system date of the computer used is set to the correct date according to the Gregorian calendar. This License commences upon the installation of the SOFTWARE and expires at the time indicated by either the hardware License key, the License certificate or as embedded in the SOFTWARE by means of a termination date or run-time limitation. This License terminates automatically without notice from STROMASYS upon the expiration of its term or if the Licensee fails to comply with any provision of this License. If the term of the License is classified as unlimited or perpetual and paid in full, the License will only terminate if the Licensee fails to comply with any provision of this License. Upon termination of the License, the Licensee shall remove the SOFTWARE from its computer.

1.4.5. Limited Warranty

STROMASYS warrants the media on which the SOFTWARE is furnished to be free of defects in material and workmanship, under normal use, for a period of ninety (90) days following the date of delivery to the Licensee. In the event of defects, STROMASYS' shall replace the defective media that has been returned to STROMASYS or the supplier with the Licensee's dated invoice and is shown to be defective. In the event that STROMASYS is unable to replace defective media or functionality, STROMASYS can refund the price paid by the Licensee for the product upon return of the License key and media.

This SOFTWARE and accompanying documentation (including instructions for use) are provided "as is" without warranty of any kind. STROMASYS does not warrant, guarantee, or make any representations regarding the use, or the results of use of the SOFTWARE or documentation in terms of correctness, accuracy, reliability or otherwise. The entire risk as to the results and performance of the SOFTWARE is assumed by the Licensee.

STROMASYS disclaims all other warranties, either express or implied, including but not limited to the implied warranties of merchantability and fitness for a particular purpose.. No oral or written information or advice given by STROMASYS, its dealers, distributors, agents or employees shall create a warranty or in any way increase the scope of this warranty and the Licensee may not rely on any such information or advice.

Neither STROMASYS nor anyone else who has been involved in the creation, production or delivery of this product shall be liable for any direct, indirect, consequential or incidental damages (including damages for loss of business profits, business interruption, loss of business information, and the like) arising out of the use or inability to use such product even if STROMASYS has been advised of the possibility of such damages.

Should any other warranties be found to exist, such warranties shall be limited in duration to ninety (90) days following the date of delivery to the Licensee. In no event will STROMASYS'

liability for any damages to the Licensee or any other person exceed the amount paid for the license to use the SOFTWARE.

You acknowledge that you understand that this software is not designed or licensed for use in applications in hazardous environments such as operation of nuclear facilities, aircraft navigation or control or life critical applications. STROMASYS expressly disclaims any liability resulting from use of the software in any such applications and accepts no liability in respect of any actions or claims based on the use of the software in any such applications by you. For the purpose of this paragraph the term "Life critical application" means an application in which the functioning or malfunctioning of the software may result directly or indirectly in physical injury or loss of human life.

1.4.6. Intellectual Property Infringement

STROMASYS shall defend, indemnify and hold the Licensee harmless from and against any third party claim alleging the infringement of any patent, copyright, trademark or other intellectual property right asserted against the Licensee by a third party based upon Licensee's authorized use of the SOFTWARE. If Licensee's use of any of the SOFTWARE is, or in STROMASYS' opinion is likely to be, enjoined due to the type of infringement specified above, or if required by settlement, STROMASYS will either: (a) substitute for the SOFTWARE substantially functionally similar programs and documentation; (b) procure for the Licensee the right to continue using the SOFTWARE; or if (a) and (b) are commercially impracticable, (c) terminate the Agreement and refund the license fees and maintenance fees paid by the Licensee as reduced to reflect the use of the SOFTWARE from the applicable license purchase date prorated over a three (3) year period.

The foregoing indemnification obligation of STROMASYS shall not apply: (1) if the SOFTWARE is modified by any party other than STROMASYS without STROMASYS prior written consent, but solely to the extent the alleged infringement is caused by such modification; (2) the SOFTWARE is combined with other non-STROMASYS products or process not contemplated by the Documentation, but solely to the extent the alleged infringement is caused by such combination; (3) to any use of the Software that is not authorized by the Documentation.

If a claim under this Section is received by the Licensee, the Licensee will provide STROMASYS: (i) prompt notice of such claim giving (but in any event notice in sufficient time for STROMASYS to respond without prejudice, but not later than 5 (five) days from receipt of such claim); (ii) the exclusive right to control and direct the investigation, defense, and settlement of such claim; and (iii) all reasonable necessary cooperation, at STROMASYS expense.

1.4.7. Export

The Licensee agrees not to export or re-export products or any part thereof including media in any form without obtaining the appropriate government licenses, if required.

1.4.8. Applicable Law; Claims and Disputes

This License shall be governed and construed in accordance with the laws of Switzerland.

Any claim or dispute between the Licensee and STROMASYS or against any agent, employee, successor or assignee of STROMASYS, whether related to this Agreement shall be resolved by binding arbitration in Geneva in accordance with the Swiss Rules of International Arbitration.

This Agreement constitutes the entire agreement between the end user and STROMASYS.

YOU ACKNOWLEDGE THAT YOU HAVE READ THIS SOFTWARE LICENSE AND LIMITED WARRANTY, UNDERSTAND IT, AND AGREE TO BE BOUND BY ITS TERMS AND CONDITIONS. YOU ALSO AGREE THAT THIS IS THE COMPLETE AND EXCLUSIVE

STATEMENT OF AGREEMENT BETWEEN THE PARTIES AND SUPERSEDES ALL PROPOSALS OR PRIOR AGREEMENTS, ORAL OR WRITTEN, AND ANY OTHER COMMUNICATIONS BETWEEN THE PARTIES RELATING TO THE SUBJECT MATTER OF THIS SOFTWARE LICENSE AND LIMITED WARRANTY.

Chapter 2. Overview

2.1. General Description

Modern software operating systems contain a hardware abstraction layer or HAL. The HAL creates a software layer on top of the hardware to "virtualize" the functionality of the hardware components. The CHARON-AXP products are essentially HALs of complete HP Alpha systems, including the HP Alpha I/O devices. They are mathematically precise models of HP Alpha hardware, and contain modules of HP ALPHA CPUs, the console subsystem, the buses and I/O adapters, disks and tapes.

After installation of CHARON-AXP on a general purpose host platform, it provides an exact model of a working HP Alpha system. On this 'virtual' system you install your HP Alpha operating system and HP Alpha applications, just as if you had purchased new HP Alpha hardware. No conversion or sources are needed, and you boot your HP Alpha system as usual. The CHARON-AXP systems execute the same binary HP Alpha code and the same I/O drivers as on the original hardware. We tested with the original HP Alpha hardware diagnostics to verify compatibility.

What you obtain is an HP Alpha, typically running at comparable speed and with a significantly smaller footprint, a reduction in cost of maintenance and energy consumption. An additional advantage of CHARON-AXP over HP Alpha hardware is the scalability with its host system. CHARON-AXP performance is proportional to the host system performance, and every time you move to a faster host system your 'virtual Alpha' will also get faster.

Another improvement over the hardware is the amount of memory each model of CHARON-AXP supports; most emulated models supports up to 32 GB of operating memory (up to 64 Gb for GS80, up to 128 GB for GS160, and up to 256 GB for GS320).

This guide covers:

- The selection of a suitable host system, essentially a multi-core server configured for the specific requirements of a CHARON-AXP product. Each product has its optimal host platform to get the best HP Alpha system performance. Ask Stomasys or one of its Resellers for configuration details for your specific system requirements.
- The installation process of the CHARON-AXP product, which is not significantly different from the installation of any other applications.
- The CHARON-AXP configuration settings that allow you to specify the HP Alpha system configuration of your choice
- The HP Alpha software installation process is not described in detail, since it is identical to HP Alpha hardware, and your HP Alpha software documentation applies. Solutions are provided to transfer the contents of the existing HP Alpha system and user disks, avoiding a complete system re-installation in most cases.

Like the original HP Alpha system CHARON-AXP can run the same supported operating systems, such as Tru64 and OpenVMS. Windows NT and Linux are not supported.

2.2. CHARON-AXP User Guide Structure

Stomasys has been building cross platform computer system virtualization products since 1999. The CHARON-AXP product line, which provides Hewlett-Packard Alpha hardware functionality as a virtualization layer on industry standard servers, has followed a development path comparable to the original DEC (now HP) Alpha hardware.

For even higher performance, CHARON-AXP can be clustered with shared storage or network clusters. With this technology, it is possible to replace large DEC Alpha data centers with a single rack of modern servers.

To get the best performance from the CHARON-AXP virtualization layer, it is essential to use a high performance 64-bit host system. This manual provides the guidelines for host system selection, CHARON-AXP installation and operation.

The Stomasys products and virtual AXP systems covered in this guide are:

- Product: **CHARON-AXP/4100**, includes the following virtual AXPs:
 - *CHARON-AXP/AS400*, a single 64-bit CPU HP AlphaServer replacement.
 - *CHARON-AXP/AS800*, a single x64 CPU HP AlphaServer replacement.
 - *CHARON-AXP/AS1000*, a single x64 CPU HP AlphaServer replacement.
 - *CHARON-AXP/AS1000A*, a single x64 CPU HP AlphaServer replacement.
 - *CHARON-AXP/AS1200*, a 2 64-bit CPU HP AlphaServer replacement.
 - *CHARON-AXP/AS2000*, a 2 64-bit CPU HP AlphaServer replacement.
 - *CHARON-AXP/AS2100*, a 4 64-bit CPU HP AlphaServer replacement.
 - *CHARON-AXP/AS4000*, a 2 64-bit CPU HP AlphaServer replacement.
 - *CHARON-AXP/AS4100*, a 4 64-bit CPU HP AlphaServer replacement.
- Product: **CHARON-AXP/DS10**, includes the following virtual AXPs:
 - *CHARON-AXP/DS10L*, a single 64-bit CPU HP AlphaServer replacement.
 - *CHARON-AXP/DS15*, a single 64-bit CPU HP AlphaServer replacement.
- Product: **CHARON-AXP/DS20**, includes the following virtual AXPs:
 - *CHARON-AXP/DS20*, a 2 64-bit CPU HP AlphaServer replacement.
 - *CHARON-AXP/DS25*, a 2 64-bit CPU HP AlphaServer replacement.
- Product: **CHARON-AXP/ES40**, includes the following virtual AXPs:
 - *CHARON-AXP/ES40*, a 4 64-bit CPU HP AlphaServer replacement.
 - *CHARON-AXP/ES45*, a 4 64-bit CPU HP AlphaServer replacement.
- Product: **CHARON-AXP/GS80**, includes the following virtual AXPs:
 - *CHARON-AXP/GS80*, an 8 64-bit CPU HP AlphaServer replacement.
- Product: **CHARON-AXP/GS160**, includes the following virtual AXPs:
 - *CHARON-AXP/GS160*, a 16 64-bit CPU HP AlphaServer replacement.
- Product: **CHARON-AXP/GS320**, includes the following virtual AXPs:
 - *CHARON-AXP/GS320*, a 32 64-bit CPU HP AlphaServer replacement.

2.3. CHARON-AXP hardware compatibility

The CHARON-AXP virtualization layers are tested with the UETP set of tests. HP has verified that the CHARON-AXP test results correspond to correctly functioning HP Alpha hardware.

HP provides OpenVMS and layered product licenses for the transfer from a hardware (HP Alpha) to CHARON-AXP; see the following web page:

<http://h71000.www7.hp.com/openvms/vax-emulator.html>

When CHARON-AXP is running on HP products, the transfer licenses maintain the HP OpenVMS/Alpha and layered software warranties.

The HP Alpha components represented in CHARON-AXP are designed to operate like their hardware equivalents. In addition to AXE like set of the CPU tests (physical Alpha CPU was sampled with billions of the instruction test cases covering various instruction/operand forms and compared with emulation up to 100% binary equivalence which gives conformance with Alpha architecture) we use HP Alpha hardware diagnostics to verify that a virtual HP Alpha component corresponds to its hardware. To avoid adding unnecessary complexity, the virtual components do not include diagnostic logic that was not used in normal hardware operation. Wherever possible without compromising compatibility, the virtual devices were 'redesigned' to avoid hardware limitations. For example, some virtual HP Alphas support a total emulated memory of up to 256 GB, despite restrictions of particular hardware models.

The purpose of HP Alpha virtualization is to replace HP Alpha hardware and any HP Alpha operating system or binary application that runs on HP Alpha hardware. Depending on emulated hardware model of virtual HP Alpha system OpenVMS 6.2-1H3, 7.1, 7.1-1H1, 7.2, 7.2-1, 7.2-2, 7.3, 7.3-1, 7.3-2, 8.3, and 8.4 and Tru64 3.2C, 4.0a – 4.0g, 5.0, and 5.1 with various patch levels were specifically tested.

Since the performance of a virtual HP Alpha depends on the host system hardware, its components operate at a different speed compared to the equivalent HP Alpha hardware. This is similar to moving a HP Alpha operating system and its applications to a HP Alpha with faster hardware components. The HP Alpha operating system will schedule the various application requests as before and applications will simply complete faster. Virtual real-time components, for instance the HP Alpha system clock, receive the correct timing interrupts and will operate as expected, provided the host system meets the specified minimum system requirements.

Every effort has been made to handle unusual HP Alpha coding sequences correctly. Self-modifying HP Alpha binary code, as is used in Oracle RDB, is part of the verification tests and executes correctly. Note that (generally undesirable) coding techniques like using NOOPs for software delay loops can give unexpected results as Virtual HP Alpha CPU executes NOOPs very quickly.

Chapter 3. CHARON-AXP hosting and performance

This chapter describes the minimum hardware and software requirements the host system must meet for the CHARON-AXP virtualization layer to work properly. Some requirements are checked during installation and/or execution time. If these limits are not met, CHARON-AXP will simply not install or operate. Other limits are 'soft' and invoke a performance reduction ('safe mode') as described in this manual below.

3.1. Host operating system requirements

The CHARON-AXP virtualization layers are designed for a Windows™ x64 server platform. Supported Windows 64bit versions:

- Windows 7 Ultimate and Professional
- Windows Server 2012 Standard
- Windows Server 2008 R2 Standard

Those versions of Windows could be installed on the physical hardware, or in a VMware virtual appliance. Supported VMware hypervisor versions are ESXi 4.1 update 1 and ESXi 5.0.

Not supported host operating systems are: Windows XP, Windows 2003 Server and Professional, Windows 2003 R2 Server and Professional, Windows Server 2008, Windows Vista, Windows Server 2003, Windows 2000 Server and Professional, and earlier versions of Windows.

3.2. Host operating system requirements

3.2.1. Common requirements

3.2.1.1. CPU selection

The CHARON-AXP products require a multi-processor host system for their operation. The host system must have a physical CPU core available for each virtual Alpha CPU. CHARON-AXP uses extra host CPU cores to perform I/O and DIT (Dynamic Instruction Translation) compilation tasks. Thus, the number of extra CPU (cores) required depends on the particular configuration and operation conditions. The optimal configuration is achieved when on top load you have at least one host CPU (core) idle 100% available for the host operation system use. Leaving too less number of the host CPU (cores) to the I/O and ACE (DIT) will result in performance reduction and malfunction especially in SMP environment. The recommended host configurations for the specific CHARON-AXP products are as follows:

- For systems with light load, number of available physical CPU cores should be greater than or equal to 1.5 times number of emulated AXP CPUs. For example, for lightly loaded GS80 system with 8 AXP cores a hosting server with at least 12 CPU cores are required.
- For systems with medium to heavy load, number of available physical CPU cores should be greater than or equal to 2 times number of emulated AXP CPUs. For example, for heavy loaded GS80 system with 8 AXP cores a hosting server with at least 16 CPU cores are required.
- CPU type recommendations:

- Generally Intel CPUs give advantage to CHARON-AXP over AMD CPUs.
- For configurations with 8 or less virtual AXP CPU cores: Intel Xeon 5600 series or newer, at least 3GHz
- For configurations with 7 or more virtual AXP CPU cores: Intel Xeon 7500 series or newer, at least 2.26 GHz
- If AMD CPUs are the only available option, Opteron 6100 series or newer, at least 2.2 GHz. (AMD CPUs older than K10 do not support cmpxchg16b instruction required for normal CHARON-AXP SMP operations)

Please refer to the SPDs for the additional information.

3.2.1.2. Host system hardware platform recommendations

HP Proliant server products (ML-series towers, DL-series rack mount or BL-series blade servers) with sufficient CPU cores, memory, storage, and network adapter capacity are recommended

For predictable HP Alpha performance the host system must be dedicated to the CHARON-AXP virtualization layer, with the possible exception of a co-resident HP Alpha console terminal or X-terminal emulator

3.2.1.3. Host system memory

The minimum host memory size depends on the amount of HP Alpha memory that is requested from the HP Alpha virtualization layer and on the number of CHARON instances running on one host. As a rule of thumb, the minimum host memory is the amounts of HP Alpha memory multiplies by the number of the instances +2 GB, with a minimum of 2048Mb (a recommended amount of memory is 4096Mb).

The maximum amount of HP Alpha memory that can be created in the CHARON-AXP products and is supported by OpenVMS/Alpha is 32 GB. For details, see the HP Alpha memory size specification

3.2.1.4. Disk storage

The CHARON-AXP virtualization layer requires approximately 30 MB disk space, not counting any (virtual) HP Alpha disks. HP Alpha disks can be in the form of physical disks (locally or on an external storage subsystem) or as HP Alpha disk images, which appear as standard files. When HP Alpha disk images are used to represent HP Alpha disk drives, the disk image files have the same size as the equivalent HP Alpha disk hardware, regardless of their degree of utilization

When physical disks are used for the virtual Alpha, these disks are connected as SCSI devices to the host platform (locally, via FibreChannel or iSCSI), regardless of the disk architecture configured in the HP Alpha environment. These physical disks are formatted by the HP Alpha operating system and cannot be used by the host system.

3.2.2. Windows specific requirements

3.2.2.1. Ethernet adapters

In CHARON, each Ethernet adapter has its own device driver placed on top of a dedicated physical Ethernet adapter. The adapter must support dynamic MAC address changes (in essence does not require a reboot of the host system to reload a MAC address). Most modern

adapters support necessary functionality. There are two reasons for the requirement of dedicated Ethernet adapters:

- A host system protocol of the same type (e.g. TCP/IP) would interfere with the same protocol running on its virtual instance.
- For security reasons, the virtual network adapter uses special code that excludes access from the external network to anything but the Ethernet drivers running on the virtual CHARON system. This prevents penetration of malicious code into the host system from the external network

The exceptions to this rule are:

- If the host system uses iSCSI services, in which case a dedicated LAN with a separate adapter for the iSCSI storage services is recommended.
- If the host system requires Windows remote desktop connectivity for remote management, in which case separate network adapter and protected VLAN is recommended.

3.2.2.2. Other host system requirements

The host system must provide a USB port for the USB license key. The license key is used constantly by CHARON during operation; it is recommended to connect the key directly to the system USB hub and not via an external USB hub which can cause access problems. Key disconnection causes termination of operation within a few minutes. Note that a quick reconnection of the key might not cancel termination.

The CHARON virtualization layer interacts directly in several areas with the host system hardware. Where possible without compromising reliability, virtual peripherals are 'mapped' through Windows drivers to the local hardware. Some host peripherals that work in Windows will not function correctly with the CHARON layer. Some examples are Ethernet adapters that cannot change their MAC address without a power cycle and some classes of peripherals connected to the host system via USB or Firewire connections.

External SCSI devices (i.e. tapes and disks) are accessed directly by the drivers of the operating system running on CHARON and do not need the installation of a Windows driver (legacy HP devices like a TK50 tape drive for example), although they can operate through a Windows driver if that one is available.

3.3. General performance considerations

The configurations referred to earlier in this chapter was the target systems used for best performance during product design. The *functionality* of the HP Alpha virtualization layer is the same (in essence an accurate model of the corresponding HP Alpha system) for every host platform that meets the minimum requirements. The effective HP Alpha system performance delivered by CHARON-AXP depends on the host system. This allows for instance CHARON-AXP/ES40 to meet or exceed HP AlphaServer ES40 performance when executing on an HP Proliant. CHARON-AXP delivers approximately 380 SPEC2000 INT and 370 SPEC2000 FP per CPU when run on a Xeon 5680 host.

For lower performance requirements, CHARON-AXP can be used on smaller platforms. Since each of Virtual HP Alpha components puts its own requirements on the host system, it is important to look at your specific requirements before deciding what type of host system to use.

Experience shows that the three main areas of performance consideration are: HP Alpha CPU performance, disk I/O speed and network connections.

3.3.1. AXP CPU performance

The component in the virtualization layer that creates a HP Alpha CPU runs several concurrent tasks using a complex proprietary algorithm to optimize performance.

Above 2 GHz host CPU frequency, the memory bandwidth and latency becomes an important virtualization layer performance factor in the current host CPU architecture. Important parameters are host cache memory size (the larger the better) and host memory latency (the lower the better). In addition, the HP Alpha CPU floating point performance is quite dependent on the host CPU design.

The multiple CPU emulation processes that can run in the virtualization layer require a significant amount of host system memory, as specified earlier in this chapter. If less host memory is momentarily available (for example, because another application has started on the same host system), the CPU emulation process becomes less effective and can shut down completely, reducing performance. Therefore, concurrent operation of the CHARON-AXP virtualization layer with other applications on the same system is not recommended.

3.3.2. Disk I/O subsystem

CHARON-AXP Disk I/O throughput scales with the host I/O bandwidth and can exceed that of hardware HP Alpha systems with an order of magnitude. In general disk I/O is rarely a bottleneck.

3.3.3. Network connections

On a high performance host platform Virtual HP ALPHA Ethernet adapter operates approximately the same speed (1 Gbps) as counterpart, but it will not always reach the full 1 Gbps throughput of modern adapters. 1 Gbps host adapters can be used in most cases, and multiple adapters can be configured.

The use of multiple adapters will not necessary increase aggregate throughput beyond that of a single 1 Gbps host adapter. Dropping incoming packets due to temporary overload is acceptable (this happens on hardware HP Alpha systems as well) if the communications protocol can retransmit lost packets in time. For sensitive protocols, (i.e. the communication between instances of the OpenVMS distributed lock manager), configuring a separate Ethernet link reduces the risk of critical packet loss.

3.3.4. Enhancing virtualization layer reliability

CHARON-AXP executes a number of interrelated processes; each needs sufficient host system performance to provide a stable system. At several levels CHARON-AXP protects itself against a lack of host system capabilities:

- If the frequency of any of the host CPUs is below 1400 MHz, CHARON-AXP will not install. If an installed executable is started on a system below that frequency, execution will terminate. Note that laptop or desktop systems in low power mode often reduce the clock frequency of their CPU(s) below their rated speed. Disable this through the power management control panel.
- If the number of host CPUs is less than requested, execution stops and the virtual layer shuts down completely.
- When insufficient HP Alpha memory can be locked in physical host memory, safe mode is entered to reduce memory requirements. Below a critical size, the virtualization layer shuts down, dependent on the model being virtualized.

- Additional host system load due to other applications running concurrently can prevent timely access to the USB license key, causing CHARON-AXP not to start or to shut down.
- It is possible to run two or more CHARON-AXP virtualization layers on the same host system, once the number of the host CPU (cores) permits the multi-instance operation as well as product license.

As far as possible, a lack of host system resources is reported in the CHARON-AXP log file

For production use, CHARON-AXP should use a dedicated host system.

Chapter 4. CHARON licensing

4.1. General description

CHARON products are protected with licenses, issued by STROMASYS for each customer individually. The CHARON license contains all the specifics of the particular CHARON distribution.

The license is implemented in form of a hardware dongle, namely Sentinel HASP key or a software license. Please be careful with your license key since in case of its loss/damage CHARON will not start anymore unless the license key is replaced. For extra protection STROMASYS recommends to use additional backup license keys (purchased separately) that may replace the main license key for restricted period of time in case of its damage/loss.

It is also important to connect HASP license keys to computer from time to time even if CHARON is not used, since the keys contain build-in accumulator that needs to be charged. If the accumulator is completely discharged license keys may be fatally damaged.

CHARON license is read on start of each instance of CHARON and then it is re-checked with some frequency defined by the license content. In case if CHARON detects absence (or malfunction) of license key / software license it displays a warning requesting to connect the license key (enable software license) again in some given period of time. If the time is run out CHARON exits. Note that if the time-restricted license is used and it is suddenly expired CHARON tries to find its replacement (if available, i.e. connected to the host or present on network in case of network license) automatically and if it is found CHARON proceeds with using that license.

Note

The present CHARON implementation assumes that the expired license must be removed firstly to allow running CHARON to switch to some other (valid) one.

Note

CHARON software license is not distributed in case of Proof-of-Concept and evaluation installations. Only hardware dongles are used in this case.

Update of CHARON license can be performed w/o CHARON stopping ("on fly"). On next license check CHARON will use the updated license normally.

The following sections list all the main parameters of CHARON licensing mechanism.

4.1.1. General parameters

- 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

4.1.2. Products parameters

CHARON license can contain a number of product sections licensing different CHARON products. Each of them contains the following parameters:

- Commercial product name
- Commercial product code
- Commercial product version and range of build numbers suitable for running
- Range of CHARON 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 no sufficient memory.
- Number of CHARON instances that can be run in the same time
- 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)
- Number of hosts that may run CHARON in the same time (in case of networking license)
- Level of support (if any), end date of any support contract, the "First Line" Service Provider
- Frequency of CHARON license checking during CHARON execution

4.1.3. Optional parameters

CHARON license may also contain some optional parameters defining possible restrictions/extensions and additional information:

- 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 emulation of IEQ11-A IEEE488 Controller (on top of DCI-3100 IEEE488 Controller) (this parameter is applicable only for CHARON-VAX/PDP11 products)

- Parameter that enables emulation of DRV11-WA I/O controller (on top of DCI-1100 I/O controller) (this parameter is applicable only for CHARON-VAX/PDP11 products)

4.2. CHARON licensing models

CHARON licensing models are divided in 3 groups:

4.2.1. Licensing by usage of locally installed Sentinel HASP keys

This is most common way of CHARON licensing. CHARON license is embedded in Sentinel HASP dongle. It is applicable only on the host where the dongle is physically installed. CHARON installation procedure takes care of the Sentinel HASP run-time (driver) installation, so once CHARON product has been installed it is possible to plug-in the license key and start CHARON usage.

Number of CHARON instances to be run on a particular host may be restricted by the license content (see above).

4.2.2. Licensing by usage of the specific Network Sentinel HASP keys

The network Sentinel HASP key can be shared between several hosts running CHARON (including the host on which the network license is installed). If CHARON is running on the host where the network key is installed no additional steps are required in this case. If the host does not have CHARON installed it can distribute the connected network license to CHARON instances running on other hosts - in this case the Sentinel driver must be installed on that host manually.

Find the Sentinel drivers as part of CHARON kit in the folder "*hasp_install*". Just unzip the only archive it contains and run the file "*haspdinst.exe*" with parameter "*-install*" to install the driver.

Once the driver is installed it allows running CHARON on all the hosts in this particular network segment using a locally connected network license.

Note

The network license key contains a specific parameter for restriction of the number of hosts allowed to run CHARON at the same time. Together with a parameter defining the number of CHARON instances running at the same time the network license sets the total number of running CHARON instances on allowed number of hosts.

4.2.3. Licensing by software license (SL)

CHARON software license does not require any hardware to be connected to license host, but it still assumes that the Sentinel run-time must be installed. SL is a "virtual" key and it has exactly the same functionality as the hardware dongles.

Software licenses can be local and network-wide. In case if they are network-wide they behave the same way as the network HASP keys.

There is also a special type of SL license called "Provisional" (demo). It has restricted period of validity.

4.3. Multiple licenses configuration

Despite a type of licensing CHARON can use **only one valid ("active") license (of given vendor code) at the time**. This active license is displayed by HASP View utility. The utility provides its number and ID / IP address of the hosts where the active license is installed.

The current conception is that CHARON cannot check all the available license keys / SL, choose needed one, switch from one key to another one etc. This functionality is not supported at the moment

If multiple licenses (with the same vendor code) are installed in a given network segment at the time, CHARON (and the Sentinel run-time) uses the following algorithm (*not fully tested*):

1. Firstly CHARON software licenses (if any) are accessed.
2. If the software licenses are not found one of the locally installed keys are accessed. The particular accessed key is defined by internal number of USB port, so to the end user this choice may look almost as random.
3. If there is no locally installed license keys the network keys are accessed. The particular accessed key is defined by internal logic of Sentinel run-time, so it this choice is hardly predictable

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

If there is a need to extend existing license with some new CHARON products it can be done by requesting STROMASYS to provide license update for existing hardware or software license.

In case if it is absolutely impossible to avoid usage of multiple licenses there are some recommendations:

- For the hosts intended to use only locally installed licenses disable an ability to use remote licenses with a help of Sentinel Admin Control Center (see below)
- Avoid connection of multiple license keys to one host. If it impossible plug in the license key to be accessed first in the firstly checked USB slot (can be defined experimentally)
- Disable not needed licenses via Sentinel Admin Control Center (see below)

4.4. License installation

4.4.1. Installation from scratch

Installation of CHARON license consists of:

- Installation of Sentinel run-time. By default it is done automatically by CHARON installation on Windows and by installing of "aksusbcd" RPM package on Linux (this RPM package is included in each CHARON for Linux distribution).
- Physically connecting HASP license key in case of hardware dongle protection
- Collecting system fingerprint (*.c2v file), sending it to STROMASYS and applying update (*.v2c file) in case of software license. See the details below.

Sometimes a manual installation of Sentinel run-time may be required. In this case open up CHARON kit folder and proceed the following way:

- Open up the folder *"hasp_install"*
- Unzip the only archive it contains (the name can alter for different CHARON kits)
- Open up *"cmd.exe"* and run the file the archive contains: **"haspdinst.exe -install"**

4.4.2. Replacement of currently installed Sentinel run-time to other one

Replacement of currently installed Sentinel Run-time can be needed in case of:

- Upgrade to newer version
- 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:
 1. Extract content of the only ZIP file resided in *"hasp_install"* directory of the CHARON distribution to the same folder.
 2. Run **"cmd.exe"** from "Start" menu of host computer
 3. **"cd"** to the *"hasp_install"* directory
 4. Issue the following command: **"haspdinst.exe -fr -kp"**
- Install other Sentinel run-time:
 1. **"cd"** to the directory whether the target run-time resides
 2. Issue the following command: **"haspdinst.exe -install"**

4.4.3. Installation of CHARON Software License

- Run the "License Update Service" utility
- Using the "Collect Status Information" section get a system fingerprint *.c2v file
- Collect a v2c file provided by STROMASYS (in return on the system fingerprint *.c2v file) and put it somewhere on CHARON host.
- Apply the v2c file with "Apply License File" section of the utility

In case of "Provisional" (demo) license there is no need to collect system fingerprint. Just proceed with applying the v2c license file as described above.

Please see more details in the "License Update Service" utility section of this Guide.

4.5. License management

The CHARON license management is performed by usage of Sentinel Admin Control Center and specific utilities described in sub-sections below.

4.5.1. Sentinel Admin Control Center

4.5.1.1. General Description

Sentinel Admin Control Center is a web-interface to the Sentinel run-time. It allows viewing/managing available keys, enable/disable them, allow/prohibit usage of remote keys etc.

Note

Sentinel Admin Control Center is not able to display CHARON licenses - there are specific utilities for that. They will be described later.

To access Sentinel Admin Control Center start any web browser, enter `http://localhost:1947` and press **Enter**: The browser will display web interface of Sentinel Admin Control Center.

The screenshot below gives an example of its interface:

Example 4.1. Sentinel Admin Control Center, Sentinel Keys part

The screenshot shows the Sentinel Admin Control Center interface. On the left is a navigation menu with options like Sentinel Keys, Products, Features, Sessions, Update/Attach, Access Log, Configuration, Diagnostics, Help, and About. The main area is titled 'Sentinel Keys Available' and contains a table with the following data:

#	Location	Vendor	Key ID	Key Type	Configuration	Version	Sessions	Actions
1	XEON4WAYW7	68704	961833018	HASP HL NetTime 50	-	3.25	-	Browse Net Features
2	Local	68704	354850588	HASP HL NetTime 50	-	3.25	-	Products Sessions Features Blink on
3	Local	68704	1351199824	HASP HL Time	-	3.25	-	Products Sessions Features Blink on
4	RH64	DEMOMA - evaluation	464243137687019632	HASP SL AdminMode Rehostable	-	2.31	1	Browse Net Features

Below the table, there are details for the selected key (ID: 961833018):

```

Details for HL NetTime 50 (ID:961833018) on 192.168.1.22
Key Hardware Version: 6.2
Sentinel License Manager Version: 12.50 Build 1.16926
Uptime: 7 days 23 hours 45 minutes
Host: XEON4WAYW7 running Windows 7 Ultimate Build 7601 Service Pack 1 (x86 Family 15 Model 2 Stepping 5)

```

This example demonstrates that 4 license keys are available:

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

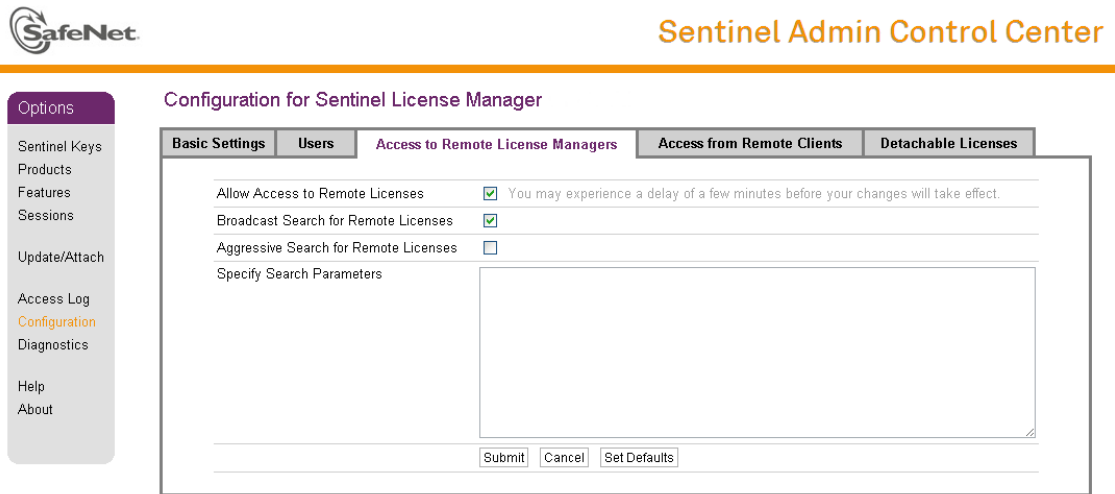
Sentinel Admin Control Center reports that there is one open session on the key (4), the other ones are not used at the moment

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

4.5.1.2. Disabling remote keys access via Sentinel Admin Control Center

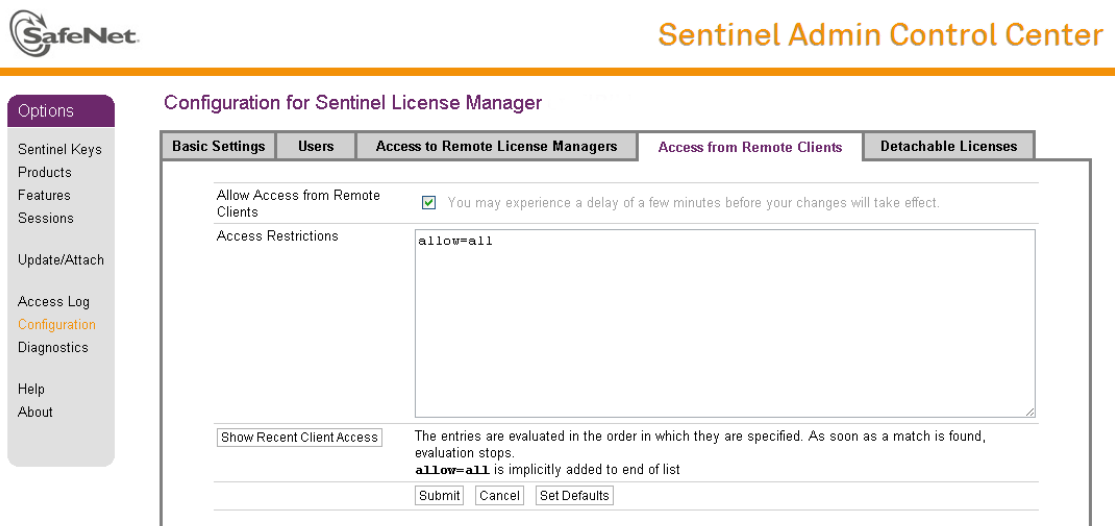
The most helpful feature of Sentinel Admin Control Center is an ability to disable access to remote keys and (if network key is installed locally) cut off license provision for remote hosts. The following examples demonstrate how it could be done:

Example 4.2. Disabling / enabling access to remote license keys via Sentinel Admin Control Center



The screenshot shows the Sentinel Admin Control Center interface. The left sidebar contains a menu with 'Options' highlighted, and sub-items: Sentinel Keys, Products, Features, Sessions, Update/Attach, Access Log, Configuration (highlighted), Diagnostics, Help, and About. The main content area is titled 'Configuration for Sentinel License Manager' and has several tabs: Basic Settings, Users, Access to Remote License Managers (selected), Access from Remote Clients, and Detachable Licenses. Under the selected tab, there are three checkboxes: 'Allow Access to Remote Licenses' (checked), 'Broadcast Search for Remote Licenses' (checked), and 'Aggressive Search for Remote Licenses' (unchecked). Below these is a text area labeled 'Specify Search Parameters' which is empty. At the bottom are buttons for 'Submit', 'Cancel', and 'Set Defaults'.

Example 4.3. Disabling / enabling access to local network license keys from remote hosts via Sentinel Admin Control Center



The screenshot shows the Sentinel Admin Control Center interface. The left sidebar is the same as in Example 4.2. The main content area is titled 'Configuration for Sentinel License Manager' and has tabs: Basic Settings, Users, Access to Remote License Managers, Access from Remote Clients (selected), and Detachable Licenses. Under the selected tab, there are two checkboxes: 'Allow Access from Remote Clients' (checked) and 'Access Restrictions' (unchecked). Below these is a text area containing the text 'allow=all'. At the bottom left is a button 'Show Recent Client Access'. Below the text area is a note: 'The entries are evaluated in the order in which they are specified. As soon as a match is found, evaluation stops. allow=all is implicitly added to end of list'. At the bottom are buttons for 'Submit', 'Cancel', and 'Set Defaults'.

4.5.2. License management specifics

CHARON kits for Windows provide 2 specific utilities for license management (see "CHARON Utilities for Windows" for detailed description):

- **HASP View.** This utility is intended for displaying the license used by CHARON. So in case of any problems with licenses it is recommended to check the license content with a help of this utility. If the license is empty probably there is something wrong with license keys themselves - in this case Sentinel Admin Control Center should be used to check the keys
- **License Update Service (HASPRUS).** The utility is provided by Sentinel Inc. It is intended for collecting host/license key information (creating "*.c2v" file) and applying updates ("*.v2c" file). It can work with all types of license keys including the software licenses.

4.6. Switch to backup key in CHARON

It is possible to specify a backup license (both hardware and software) to be used by CHARON if the main license becomes not accessible.

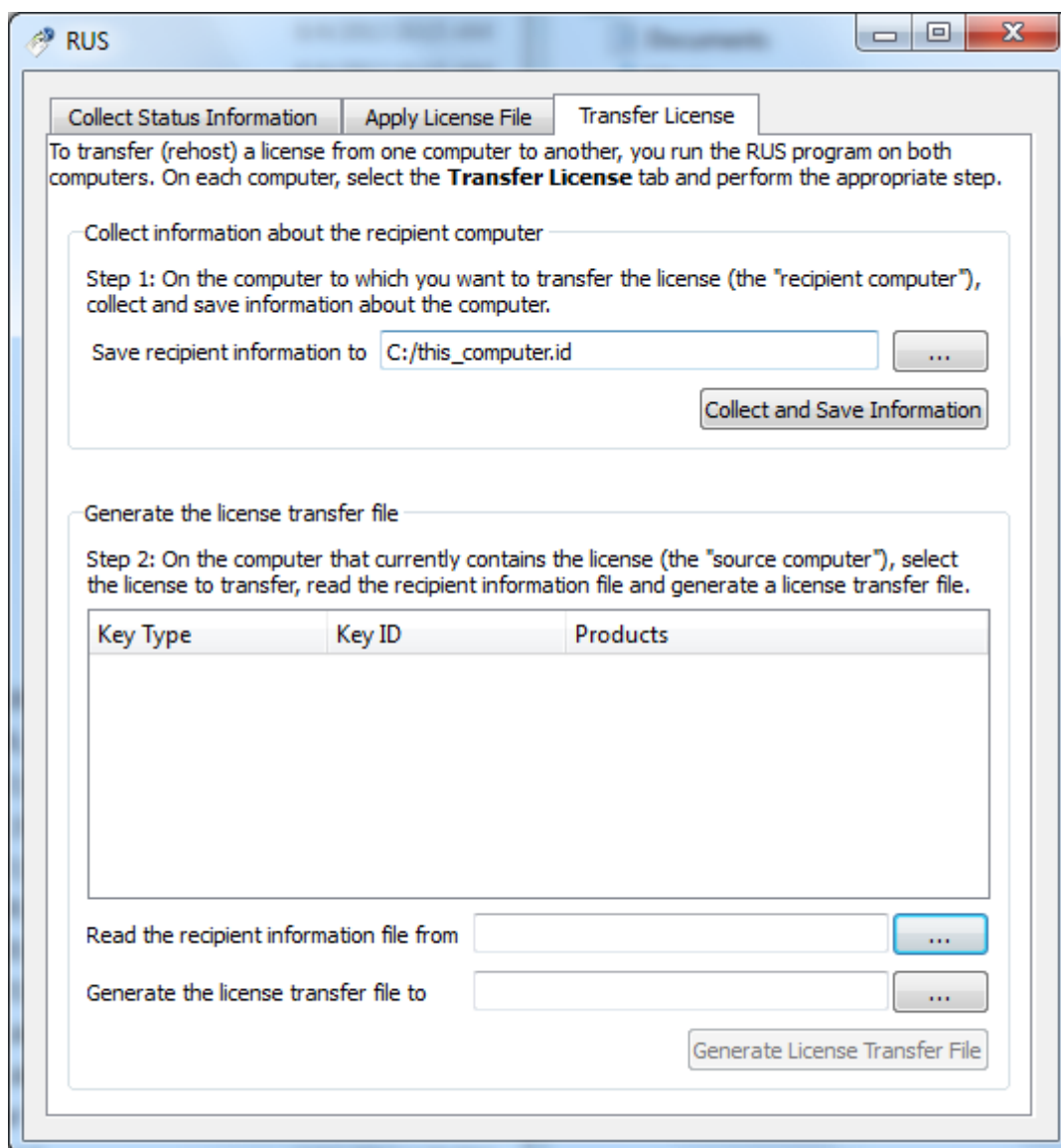
CHARON provides the following parameter to manage backup license:

"set session" parameters	Type	Value
license_key_id[N], N=0 or 1	Numeric	<p>A number (decimal Sentinel key ID) specifying the regular (N=0) and backup (N=1) license key to be used by CHARON.</p> <p>Example 4.4.</p> <pre>set session license_key_id[0] = 1877752571 set session license_key_id[1] = 354850588</pre> <p>it is also possible to specify both regular and backup key in one line:</p> <pre>set session license_key_id[0] = 1877752571 li- cense_key_id[1] = 354850588</pre> <p>Depending on presence of the regular and/or backup license key IDs in the configuration file CHARON behaves differently:</p> <ol style="list-style-type: none"> No keys are specified CHARON behaves as usual (performs unqualified search for any suitable key). If no keys are found, CHARON exits. Both keys are specified CHARON performs qualified search for regular license key. If it is not found, CHARON performs qualified search for backup license key. If it is not found, CHARON exits. Only regular key is specified CHARON performs qualified search for regular license key. If it is not found, CHARON performs unqualified search for any suitable key. If it is not found, CHARON exits. Only backup key is specified CHARON behaves as usual (performs unqualified search for any suitable key). If no keys are found, CHARON exits.

4.7. Software Licenses Transfer

It is possible to transfer CHARON Software License from one host to other with a help of "License Update Service" utility in the following way:

Open up the "License Update Service" utility in **Start->CHARON products-><product>->Utilities**, then switch to "Transfer License" tab to see available options:



To transfer installed software license to the following steps:

1. Run the **License Update Service** on the target host, select “Transfer License” tab and collect the host information into a specific *.id* file
2. Copy the target host information *.id* file to the source host
3. Run the **License Update Service** on the source host, select “Transfer License” tab, choose the license to transfer, specify the target host information file with the “...” button, choose some filename to store the license transfer file with the lower “...” button and press “Generate License Transfer File” button
4. Copy the license v2c file back to the target host and apply it as described in the utility description section.

4.8. License Deinstallation

To deinstall CHARON license follow the steps below:

- Open up the folder *"hasp_install"*

- Unzip the only archive it contains (the name can alter for different CHARON kits)
- Open up "*cmd.exe*" and run the file the archive contains: "**haspdinst.exe -fr -kp**"

Actually no manual deinstallation is required on Windows, since InstallShell is able to remove Sentinel run-time automatically on deinstallation of last CHARON kits.

4.9. Backup license keys

Backup keys are provided by STROMASYS along with standard license dongles. It is strongly recommended to order one to have instant backup for situation of damage / loss of the main license key. Please note that the backup keys may have restricted functionality:

- Run time is typically limited to 720 hours in total. It should be enough time to get replacement from STROMASYS.
- Backup license may be valid only by certain date. Please check it with STROMASYS management.

4.10. Important Notes

Please note that license key has built-in battery which must not be completely discharged. So it is strongly recommended to connect not used license keys to USB ports of some computer from time to time for charging.

Chapter 5. Installing CHARON for Windows

Before installing CHARON for Windows:

1. Make sure your Microsoft Windows OS system is brought up to the levels of the Service Pack and updates recommended in the CHARON Release Notes
2. Make sure that your system is configured as a standalone server, or if a part of a Microsoft domain, the domain policies do not block the local admin rights of CHARON account.
3. All the installation procedure executables do not have any "compatibility" settings. Otherwise the operating system may be recognized incorrectly.
4. Make sure that the local admin has full amounts of system administrative rights. Any restriction of the rights may badly affect the installation process and function of CHARON.
5. Disable Firewall on the NICs which will be used for CHARON networking

The CHARON distribution kit contains a numbered CHARON USB license key. The latest versions of the CHARON manual and Release Notes are available online on Stromasys ftp. We recommend that you read the release notes before starting the installation of CHARON. The release notes indicate any changes to the documentation, software or installation procedure since the release of this manual.

Your CHARON license key is numbered and represents the full value of your CHARON product purchase. It will not be replaced free of charge if lost; we recommend that you establish an appropriate security procedure for this high value item.

In the very unlikely case that the CHARON key fails, **DO NOT DISCARD THE KEY** and contact Stromasys immediately for replacement and recovery of the key's internal information. For very high availability requirements, a runtime limited backup key can be purchased.

Some CHARON kits may be protected with so-called "Software License" (SL), a virtual license key installed on the system. It is managed the same way as the hardware Sentinel HASP key - for example it can be updated with the License Update Service and viewed with HASP View Utility. The only difference is the process of the license installation.

5.1. License key driver installation

To run the license key driver installation and CHARON installation you must be logged in as local Administrator or as a user having local administrator privileges. CHARON should be installed and operated from the same account.

Since normal running of CHARON depends on the CHARON license key, its driver must be installed first. Insert the CHARON installation CD in the CD drive of your Windows system. **Do not connect the CHARON license key yet.** Download and unpack CHARON installation kit. The Installation tool (*InstallShell.exe*) can be found in the top folder.

Select the product(s) that you have a license(s) for and press "Next" button. After successful installation of the Visual C++ redistributables, WinHelp32 kit, CHARON Service Manager and the HASP license key driver, the installation will prompt you to choose the products components for installation. It is recommended to leave all the components selected.

The CHARON USB license key must always be connected during the use of the HP Alpha virtualization layer.

It is possible to update the license content “on fly”, w/o stopping CHARON. See the “HAS-PRUS” utility for details on how to do it. Once the license is updated CHARON will read and use it automatically.

CHARON fully supports the functionality provided by the Sentinel HASP NET license key.

Note: If you inserted the license key before its driver is installed, the Windows operating system will automatically look for the USB driver. In that case proceed as follows:

- *Cancel the Windows initiated driver installation.*
- *Browse the subfolder 'hasp_install' of the CHARON installation folder; find the HASP_SRM_driver_cmdline_X.XX.zip file, which contains the "haspdinst.exe" driver installation utility). Open up Windows console (Start->Run->cmd.exe), switch to the folder containing the HASP driver installation procedure and execute the command “haspdinst --install”.*
- *Once the USB driver is installed, continue with the CHARON installation procedure.*
- *If there is a driver installation problem, check the web page:*

<http://www.aladdin.com/support/hasp/vendor.aspx>

for the latest version of "haspdinst.exe"

5.2. Installing files and components

1. Run the CHARON installation tool (*InstallShell.exe*) from the installation source folder.
2. InstallShell guides you through the installation process as follows:
 - It informs about available products. Select the product you have a license for and then click “Next”. If there are no suitable products on the media for the host operating system the installation procedure will warn you. In this case please check that the platform is suitable for these particular products.
 - The InstallShell displays a dialog containing the End User License Agreement text in a scrolling text box. Read it carefully and select “Agree” radio button if you agree with the terms. Once it is done the “Next” button becomes available.
 - Select the folder to install CHARON (if requested), or accept the default folder (recommended) `<SystemDrive>:\Program Files\CHARON`. Click “Next”. Note that once folder is chosen it cannot be changed since this folder is considered as CHARON base directory which is used for all CHARON installations. If it is needed to change the CHARON base directory all the CHARON products should be uninstalled and then the installation should be executed again. In this case it will ask to select a folder for new location of the CHARON base directory.
 - Select components to be installed, including network driver and the Network Diagnostic Center Utility. It’s recommended to leave default selections untouched unless you have certain reasons to additionally deselect some components. Click “Next”.
 - Click “Finish” upon completion and reboot the system (see the following description in additional).
3. If the installation tool reports success and no other users than the current user are going to run CHARON in future, proceed straightly with reboot (notice that the installation procedure adds the current user to the just created CHARON-GRP user group).
4. If some additional users are going to use the CHARON virtualization layer, do the following:

- Uncheck "Reboot the system" checkbox to avoid rebooting. Press "Finish" button.
- Open up the "Start->Programs->Administrative Tools->Computer Management->Local Users and Groups->Groups->CHARON-GRP"
- Add the desired user with a right click and choose "Properties, Add"
- Make sure those users have local administrator rights.
- Reboot the system.

Unless directed elsewhere, the default installation location is the directory *C:\Program Files\CHARON\Build_XXXXX\x64* or *C:\Program Files\CHARON\Build_XXXXX\x64*, where XXXXX is a build number. A start menu item will be installed automatically and a shortcut will be added to the desktop, if requested.

5.3. Configuring network adapters

During the installation procedure, a CHARON specific NDIS5/6 packet protocol driver is installed.

Note

CHARON installs the proper NDIS driver for your operating system and abstracts out the configuration to allow you to easily move between OS versions. This means that it is acceptable to use either the NDIS5 or NDIS6 syntax regardless of your OS version.

CHARON requires a dedicated host Ethernet adapter for each virtual HP Alpha Ethernet adapter you configure; so for each adapter dedicated to CHARON, please execute the following steps:

- Start up the "Network Control Center" utility from **Start -> Programs-> CHARON -> <product> -> Utilities -> Network Control Center version 1.xx**
- Choose "Configure NIC" and press "Next"
- Choose the adapters to be used by CHARON and dedicate them to the emulator.
- Press "Cancel" to finish.
- Use only that dedicated adapters for all CHARON networking

For more information please refer to the Network Control Center chapter of this Guide.

5.4. Installation of Software License (optional)

If your CHARON kit is going to be protected with Software License (SL) the following steps should be performed in addition:

1. Open up the License Update Service
2. Choose the first tab called "Collect Status Information"
3. Select "Installation of new protection key" radio button and press "Collect Information" button. If some software license has been already installed only "Update of the existing protection key" option will be available and selected by default. In this case just press the "Collect Information" button and proceed with the following step.

4. The utility will ask you to choose a directory where to create ".C2V" file. Name this file somehow and press Ok. This file stores specific information about your system.
5. Provide the created ".C2V" file to STROMASYS Licensing Center. In return a ".V2C" file will be sent back.
6. Copy the ".V2C" file received from STROMASYS Licensing Center to the host system
7. Open up the License Update Service
8. Choose the second tab called "*Apply License File*"
9. Press "..." button and choose the ".V2C" file
10. Press "Apply Update" button. Note the status information displayed. In case of any problems or impossibility to apply the update please contact STROMASYS Licensing Center

5.5. Configuring CHARON virtualization layer

After installation you should edit a configuration file for the chosen emulator from CHARON package and install a HP operating system, for instance OpenVMS. The configuration procedure is described in the next chapters.

5.6. Uninstallation

You can remove CHARON using the same CHARON installation tool (InstallShell.exe). It can be called through Add/Remove programs applet:

Start -> Settings -> Control Panel -> Add/Remove programs

5.7. Default installation

5.7.1. File locations

The default Windows installation folder for CHARON-AXP products is "*C:\Program Files\CHARON*" (assuming "C" is the Windows system disk). The default installation puts the following files in the default installation folders:

- *Build_XXXXX\i64* folder (where XXXXX is a product build number) contains the main emulator executables and their associated DLLs; *putty.exe* (the default terminal emulator), some REG files for creating predefined PuTTY sessions, the example configuration files, etc.
- *Utilities_1.0.XXXXX* folder (where XXXX is a product build number) contains CHARON Launcher, CHARON Service Manager, MkDisk, CHARON Network Control Center, HASP-RUS, HASP View, MTD and HOSTprint utilities.
- *Drivers* folder contains the following drivers: NDIS5/NDIS6 CHARON Packet Port, EMULEX, DEFPA, DIGI and DCI1100 CHARON PCI Pass Through.
- *WebUI* folder contains files of CHARON Web User Interface. See a separate User's Guide for its description.

5.7.2. Start Menu

A Start menu item **Program** → **CHARON** → **CHARON-AXP version X.X.XXXXX** is added, with the following submenu items:

- **AlphaServer <XX>** starts the CHARON-AXP/XX virtualization layer with the default configuration file for the model "XX".
- **Utility** folder contains various CHARON utilities and reporting tools.

5.7.3. Desktop icon

During installation a shortcut "**AlphaServer <XX>**" is placed on the desktop. This shortcut starts the CHARON-AXP/XX virtualization layer with the default configuration file for the model "XX"

Chapter 6. Configuring Virtual HP Alpha

6.1. The HP Alpha system architecture

In hardware HP Alpha system, the CPU, memory, peripheral controllers and adapters are connected through the central system buses.

CHARON-AXP implements these central system buses, the HP Alpha CPU(s), memory, disk/tape controllers and the Ethernet components. When CHARON-AXP starts, it follows a configuration script and assembles a virtual HP Alpha system by combining models of the buses, the HP Alpha CPU, memory and controllers into a working unit and loading this into the host system.

The virtual peripheral devices are mapped in the configuration script to a device or service on the host system. For instance, a virtual DE500BA Ethernet adapter is associated with a dedicated physical Ethernet controller in the host system, thus connecting Virtual HP Alpha to a physical Ethernet network.

When the configuration script is fully executed, CHARON-AXP has created a complete virtual HP Alpha system. It then hands over control to the HP Alpha CPU, which will boot the HP Alpha system software in the same manner as would happen on HP Alpha hardware.

The configuration script consists of one or more text files with a *.cfg* extension. To facilitate structuring of large configurations, a part of the configuration can be stored in a separate file. Such file is incorporated in the main script with the **include** command.

Follow the steps below for a quick start with a custom configuration:

1. Choose an emulated Alpha model you would like to run, and copy the relevant default configuration file to your private configuration file (for example: *my_es40.cfg*)
2. Write a comment at the top of the *my_es40.cfg* to define the purpose for this configuration
3. Define the amount of RAM you require (for example: **set ram size=1024**). By default the memory amount is set to different values depending on the HP Alpha models
4. Define the virtual operator console mapping (for example for PuTTY: **set load virtual_serial_line OPA0 port=10003 application="putty.exe -load OPA0"**)
5. Define storage units (for example: "**set PKA container[0]="file-name.vdisk"**). This first unit will appear in CHARON-AXP as DKA0 in CHARON-AXP SRM console
6. Define the Ethernet NIC device connection in the configuration file, for example:

```
load DE500BA/dec21x4x EWA interface=EWA0
```

```
load packet_port/chnetwrk EWA0 interface="connection:CHARON"
```

It is necessary to ensure that CHARON NDIS5/NDIS6 driver is installed for the selected adapter, and there are no other protocols active for it. To ensure, run "Network Control Center" utility, select "Configure NIC" radio button, press "Next", select the defined Ethernet NIC device, press "Designate adapter to CHARON". This device will appear under Open-VMS as EWx device

7. Now you can run the configured system

6.2. The configuration command syntax

There are two types of configuration commands:

- The **load** command instructs CHARON to add a component to a system bus
- The **set** command defines the characteristics of a loaded component

To be able to load and manipulate more than one copy of a particular component a logical name is assigned to each loaded component as the following example shows:

Example 6.1.

```
load "component A" NAME1
```

```
load "component B" NAME2
```

NAME1 and NAME2 are freely chosen names that are only relevant within the configuration file. These names have no meaning and will never show up in the operating system running on the virtual environment

Using their logical names the two identical components (for instance two DE500BA Ethernet adapters) can be given individual parameters (for instance the IDs of the host adapters they should use). The example configuration files show how the logical names are used.

The parameters used with the set commands are typically assigned to a value, which can be **true/false**, a **number** or a **text string**. Numbers can be expressed in different formats, as it can be more convenient to use octal or hexadecimal formats:

- For octal use a number starting with 0; use the symbols 0 – 7. Example: 07665
- For decimal use a number starting with 1 - 9. Example: 12345
- For hexadecimal use a number starting with 0x; 0 - 9 and a – f. Example: 0x1234abc

The **set** commands are listed separately in this guide; however any load command can be extended with one or more of relevant set commands to get a more compact configuration file. For instance:

Example 6.2.

```
load "component A" NAME1 <parameter>="abcd"
```

Is equivalent to:

```
load "component A" NAME1
```

```
set NAME1 <parameter>="abcd"
```

6.3. The virtual AXP models specifics

All the emulators included to the CHARON-AXP have specific PCI bus configuration and peripherals. This specific reflects original HP Alpha system hardware configurations and is implemented in CHARON-AXP to provide better compatibility with original HP Alpha operating systems (presumably old versions of HP Tru64 UNIX Operating System)

6.3.1. AlphaServer 400 (DECchip 21072, 3 PCI slots)

In addition to 3 PCI vacant slots there are 2 PCI positions occupied by on-board devices. All 5 PCI positions are listed in the following table in the order in which Alpha SRM console enumerates them.

Slot	PCI	Device	Function	IRQ	
pci_0					
-	0	6	0	11	NCR 53C810 PCI SCSI Adapter
-	0	7	0	-	Intel i82378 PCI ISA Bridge (SATURN)
0	0	11	0	10	<i>option</i>
1	0	12	0	15	<i>option</i>
2	0	13	0	9	<i>option</i>

The IRQ stands for ISA IRQ Number because all interrupts are routed through the Intel i82378 PCI ISA Bridge (SATURN) resident cascade of Intel i8259 interrupt controllers.

So far the CHARON-AXP emulators do not support virtual NCR 53C810 PCI SCSI adapter. Instead, virtual QLOGIC ISP1040B PCI SCSI adapter is used.

6.3.2. AlphaServer 800 (DECchip 21172, 4 PCI slots)

In addition to 4 PCI vacant slots there are 3 PCI positions occupied by on-board devices. All 7 PCI positions are listed in the following table in the order in which Alpha SRM console enumerates them.

Slot	PCI	Device	Function	IRQ	
pci_0					
-	0	5	0	0	QLOGIC ISP1020 PCI SCSI Adapter
-	0	6	0	0	S3 Trio32/64 Display Adapter
-	0	7	0	-	Intel i82375 PCI EISA Bridge (MERCURY)
0	0	11	0	10	<i>option</i>
1	0	12	0	15	<i>option</i>
2	0	13	0	9	<i>option</i>
3	0	14	0	7	<i>option</i>

The IRQ stands for input line of ASIC interrupt controllers. It has nothing to do with "EISA" style interrupts. So far, the CHARON-AXP emulators do not emulate S3 Trio32/64 Display Adapter. So position of the device 6, function 0 on the PCI 0 remains empty.

6.3.3. AlphaServer 1000 (DECchip 21072, 3 PCI slots)

In addition to 3 PCI vacant slots there are 2 PCI positions occupied by on-board devices. All 5 PCI positions are listed in the following table in the order in which Alpha SRM console enumerates them.

Slot	PCI	Device	Function	IRQ	
pci_0					

Slot	PCI	Device	Function	IRQ	
-	0	6	0	12	NCR 53C810 PCI SCSI Adapter
-	0	7	0	-	Intel i82375 PCI EISA Bridge (MERCURY)
0	0	11	0	0	<i>option</i>
1	0	12	0	4	<i>option</i>
2	0	13	0	8	<i>option</i>

The IRQ stands for input line of ASIC interrupt controllers. It has nothing to do with "EISA" style interrupts. So far, the CHARON-AXP emulators do not emulate NCR 53C810 PCI SCSI adapter. Instead, emulation of QLOGIC ISP1040B PCI SCSI adapter is used.

6.3.4. AlphaServer 1000A (DECchip 21072, 7 PCI slots)

In addition to 7 PCI vacant slots there are 3 PCI positions occupied by on-board devices. All 10 PCI positions are listed in the following table in the order in which Alpha SRM console enumerates them.

Slot	PCI	Device	Function	IRQ	
pci_0					
-	0	7	0	-	Intel i82375 PCI EISA Bridge (MERCURY)
-	0	8	0	-	DECchip 21050 PCI-to-PCI Bridge)
0	0	11	0	1	<i>option</i>
1	0	12	0	2	<i>option</i>
2	0	13	0	3	<i>option</i>
pci_0					
-	1	0	0	0	NCR 53C810 PCI SCSI Adapter
3	1	1	0	7	<i>option</i>
4	1	2	0	9	<i>option</i>
5	1	3	0	11	<i>option</i>
6	1	4	0	13	<i>option</i>

The IRQ stands for input line of ASIC interrupt controllers. It has nothing to do with "EISA" style interrupts. So far, the CHARON-AXP emulators do not emulate NCR 53C810 PCI SCSI adapter. Instead, emulation of QLOGIC ISP1040B PCI SCSI adapter is used.

6.3.5. AlphaServer 1200 (1 IOD, 6 PCI slots)

In addition to 6 PCI vacant slots there are 2 PCI positions occupied by on-board devices. All 8 PCI positions are listed in the following table in the order in which Alpha SRM console enumerates them.

Slot	PCI	Device	Function	IRQ	
pci_1					
-	1	1	0	4	NCR 53C810 PCI SCSI Adapter
0	1	2	0	8	<i>option</i>
1	1	3	0	12	<i>option</i>

Slot	PCI	Device	Function	IRQ	
2	1	4	0	16	<i>option</i>
pci_0					
-	0	1	0	-	Intel i82375 PCI EISA Bridge (MERCURY)
3	0	2	0	8	<i>option</i>
4	0	3	0	12	<i>option</i>
5	0	4	0	16	<i>option</i>

So far, the CHARON-AXP emulators do not emulate NCR 53C810 PCI SCSI adapter. Instead, emulation of QLOGIC ISP1040B PCI SCSI adapter is used.

6.3.6. AlphaServer 2000 (T2, 3 PCI slots)

In addition to 3 PCI vacant slots there are 3 PCI positions occupied by on-board devices. All 6 PCI positions are listed in the following table in the order in which Alpha SRM console enumerates them.

Slot	PCI	Device	Function	IRQ	
pci_0					
-	0	0	0	2	DEC TULIP PCI Ethernet adapter
-	0	1	0	1	NCR 53C810 PCI SCSI Adapter
-	0	2	0	-	Intel i82375 PCI EISA Bridge (MERCURY)
0	0	6	0	0	<i>option</i>
1	0	7	0	4	<i>option</i>
2	0	8	0	5	<i>option</i>

The IRQ stands for input line of T2 resident cascade of Intel i8259 interrupt controllers. It has nothing to do with "EISA" style interrupts.

So far the CHARON-AXP emulators do not support virtual NCR 53C810 PCI SCSI adapter. Instead, virtual QLOGIC ISP1040B PCI SCSI adapter is used.

6.3.7. AlphaServer 2100 (T2, 3 PCI slots)

In addition to 3 PCI vacant slots there are 3 PCI positions occupied by on-board devices. All 6 PCI positions are listed in the following table in the order in which Alpha SRM console enumerates them.

Slot	PCI	Device	Function	IRQ	
pci_0					
-	0	0	0	2	DEC TULIP PCI Ethernet adapter
-	0	1	0	1	NCR 53C810 PCI SCSI Adapter
-	0	2	0	-	Intel i82375 PCI EISA Bridge (MERCURY)
0	0	6	0	0	<i>option</i>
1	0	7	0	4	<i>option</i>
2	0	8	0	5	<i>option</i>

The IRQ stands for input line of T2 resident cascade of Intel i8259 interrupt controllers. It has nothing to do with "EISA" style interrupts.

So far the CHARON-AXP emulators do not support virtual NCR 53C810 PCI SCSI adapter. Instead, virtual QLOGIC ISP1040B PCI SCSI adapter is used.

6.3.8. AlphaServer 4000 (2 IODs, 16 PCI slots)

In addition to 16 PCI vacant slots there are 2 PCI positions occupied by on-board devices. All 18 PCI positions are listed in the following table in the order in which Alpha SRM console enumerates them.

Slot	PCI	Device	Function	IRQ	
pci_1					
-	1	1	0	4	NCR 53C810 PCI SCSI Adapter
-	1	2	0	8	<i>option</i>
-	1	3	0	12	<i>option</i>
-	1	4	0	16	<i>option</i>
-	1	5	0	20	<i>option</i>
pci_0					
-	0	1	0	-	Intel i82375 PCI EISA Bridge (MERCURY)
-	0	2	0	8	<i>option</i>
-	0	3	0	12	<i>option</i>
-	0	4	0	16	<i>option</i>
-	0	5	0	20	<i>option</i>
pci_3					
-	3	2	0	8	<i>option</i>
-	3	3	0	12	<i>option</i>
-	3	4	0	16	<i>option</i>
-	3	5	0	20	<i>option</i>
pci_2					
-	2	2	0	8	<i>option</i>
-	2	3	0	12	<i>option</i>
-	2	4	0	16	<i>option</i>
-	2	5	0	20	<i>option</i>

So far the CHARON-AXP emulators do not support virtual NCR 53C810 PCI SCSI adapter. Instead, virtual QLOGIC ISP1040B PCI SCSI adapter is used.

6.3.9. AlphaServer 4100 (1 IOD, 8 PCI slots)

In addition to 8 PCI vacant slots there are 2 PCI positions occupied by on-board devices. All 10 PCI positions are listed in the following table in the order in which Alpha SRM console enumerates them.

Slot	PCI	Device	Function	IRQ	
pci_1					

Slot	PCI	Device	Function	IRQ	
-	1	1	0	4	NCR 53C810 PCI SCSI Adapter
-	1	2	0	8	<i>option</i>
-	1	3	0	12	<i>option</i>
-	1	4	0	16	<i>option</i>
-	1	5	0	20	<i>option</i>
pci_0					
-	0	1	0	-	Intel i82375 PCI EISA Bridge (MERCURY)
-	0	2	0	8	<i>option</i>
-	0	3	0	12	<i>option</i>
-	0	4	0	16	<i>option</i>
-	0	5	0	20	<i>option</i>

So far the CHARON-AXP emulators do not support virtual NCR 53C810 PCI SCSI adapter. Instead, virtual QLOGIC ISP1040B PCI SCSI adapter is used.

6.3.10. AlphaServer DS10L (1 PCI bus, 4 PCI slot)

In addition to 4 PCI vacant slots there are 5 PCI positions occupied by on-board devices. All 9 PCI positions are listed in the following table in the order in which Alpha SRM console enumerates them.

Slot	PCI	Device	Function	IRQ	
pci_0					
-	0	7	0	-	ALi M1543C PCI ISA bridge
-	0	9	0	29	DECchip 21143 PCI Ethernet Adapter
-	0	11	0	30	DECchip 21143 PCI Ethernet Adapter
-	0	13	0	-	ALi M1543C PCI IDE/ATAPI controller
1	0	14	0	35	<i>option</i>
2	0	15	0	39	<i>option</i>
3	0	16	0	43	<i>option</i>
4	0	17	0	47	<i>option</i>
-	0	19	0	11	ALi M1543C PCI USB adapter

So far, the CHARON-AXP emulators do not support virtual ALi M1543C PCI USB adapter. The position of the device 19, function 0 on the PCI 0 remains empty.

6.3.11. AlphaServer DS15 (2 Pchips, 4 PCI slots)

In addition to 4 PCI vacant slots there are 7 PCI positions occupied by on-board devices. All 11 PCI positions are listed in the following table in the order in which Alpha SRM console enumerates them.

Slot	PCI	Device	Function	IRQ	
pci_2					

Slot	PCI	Device	Function	IRQ	
1	2	7	0	40	<i>option</i>
2	2	8	0	36	<i>option</i>
3	2	9	0	24	<i>option</i>
4	2	10	0	20	<i>option</i>
pci_0					
-	0	7	0	-	ALi M1543C PCI ISA bridge
-	0	8	0	-	Adaptec AIC-7899 (channel 0)
-	0	8	1	-	Adaptec AIC-7899 (channel 1)
-	0	9	0	-	Intel i82559 PCI Ethernet Adapter
-	0	10	0	-	Intel i82559 PCI Ethernet Adapter
-	0	13	0	-	ALi M1543C PCI IDE/ATAPI controller
-	0	19	0	-	ALi M1543C PCI USB adapter

The IRQ stands for bit position in DRIR of TITAN chip. It has nothing to do with "ISA" style interrupts which are routed to IRQ 55 (including ALi M1543C PCI IDE/ATAPI controller).

So far the CHARON-AXP emulators do not emulate Adaptec AIC-7899. Instead, emulation of QLOGIC ISP1040B is used.

So far the CHARON-AXP emulators do not emulate Intel i82559. Instead, emulation of DECchip 21143 is used.

So far the CHARON-AXP emulators do not emulate ALi M1543C PCI USB adapter. So position of the device 19, function 0 on the PCI 0 remains empty.

6.3.12. AlphaServer DS20 (2 Pchips, 6 PCI slots)

In addition to 6 PCI vacant slots there are 5 PCI positions occupied by on-board devices. All 11 PCI positions are listed in the following table in the order in which Alpha SRM console enumerates them.

Slot	PCI	Device	Function	IRQ	
pci_1					
4	1	7	0	47	<i>option</i>
5	1	8	0	43	<i>option</i>
6	1	9	0	39	<i>option</i>
pci_0					
-	0	5	0	-	ALi M1543C PCI ISA bridge
-	0	6	0	19	Adaptec AIC-7895 (channel 0)
-	0	6	1	18	Adaptec AIC-7895 (channel 1)
-	0	15	0	-	ALi M1543C PCI IDE/ATAPI controller
-	0	19	0	-	ALi M1543C PCI USB adapter
1	0	7	0	31	<i>option</i>
2	0	8	0	27	<i>option</i>
3	0	9	0	23	<i>option</i>

The IRQ stands for bit position in DRIR of Tsunami/Typhoon Chip. It has nothing to do with "ISA" style interrupts which are routed to IRQ 55 (including ALi M1543C PCI IDE/ATAPI controller).

Unless SCSI option is plugged into PCI slot 4, 5, or 6, the onboard SCSI controllers appear as PKA (pka7.0.0.6.0) and PKB (pkb7.0.0.106.0) respectively.

So far the CHARON-AXP emulators do not support virtual Adaptec AIC-7895 PCI SCSI adapter. Instead, virtual QLOGIC ISP1040B PCI SCSI adapter is used.

So far the CHARON-AXP emulators do not support virtual ALi M1543C PCI USB adapter. So position of the device 19, function 0 on the PCI 0 remains empty.

6.3.13. AlphaServer DS25 (2 Pchips, 6 PCI slots)

In addition to 6 PCI vacant slots there are 7 PCI positions occupied by on-board devices. All 13 PCI positions are listed in the following table in the order in which Alpha SRM console enumerates them.

Slot	PCI	Device	Function	IRQ	
pci_0					
-	0	7	0	-	ALi M1543C PCI ISA bridge
-	0	8	0	-	Intel i82559 PCI Ethernet Adapter
1	0	9	0	24	<i>option</i>
2	0	10	0	12	<i>option</i>
-	0	16	0	-	ALi M1543C PCI IDE/ATAPI controller
-	0	19	0	-	ALi M1543C PCI USB adapter
pci_1					
3	1	1	0	28	<i>option</i>
4	1	2	0	32	<i>option</i>
pci_2					
-	2	1	0	-	Adaptec AIC-7899 (channel 0)
-	2	1	1	-	Adaptec AIC-7899 (channel 1)
-	2	5	0	-	BroadCom BCM5703 PCI Ethernet Adapter
pci_3					
5	3	1	0	36	<i>option</i>
6	3	2	0	40	<i>option</i>

The IRQ stands for bit position in DRIR of TITAN Chip. It has nothing to do with "ISA" style interrupts which are routed to IRQ 55 (including ALi M1543C PCI IDE/ATAPI controller).

So far the CHARON-AXP emulators do not emulate Intel i82559. Instead, emulation of DECchip 21143 is used.

So far the CHARON-AXP emulators do not emulate ALi M1543C PCI USB adapter. So position of the device 19, function 0 on the PCI 0 remains empty.

Unless SCSI option is plugged into PCI slot 1, 2, 3, or 4, the onboard SCSI controllers appear as PKA (pka7.0.0.1.2) and PKB (pkb7.0.0.101.2) respectively.

So far the CHARON-AXP emulators do not emulate Adaptec AIC-7899. Instead, emulation of QLOGIC ISP1040B is used.

So far the CHARON-AXP emulators do not emulate BroadCom BCM5703. Instead, emulation of DECchip 21143 is used.

6.3.14. AlphaServer ES40 (2 Pchips, 10 PCI slots)

In addition to 10 PCI vacant slots there are 3 PCI positions occupied by on-board devices. All 13 PCI positions are listed in the following table in the order in which Alpha SRM console enumerates them.

Slot	PCI	Device	Function	IRQ	
pci_1					
5	1	1	0	24	<i>option</i>
6	1	2	0	28	<i>option</i>
7	1	3	0	32	<i>option</i>
8	1	4	0	36	<i>option</i>
9	1	5	0	40	<i>option</i>
10	1	6	0	44	<i>option</i>
pci_0					
1	0	1	0	8	<i>option</i>
2	0	2	0	19	<i>option</i>
3	0	3	1	16	<i>option</i>
4	0	4	0	20	<i>option</i>
-	0	5	0	-	ALi M1543C PCI ISA bridge
-	0	15	0	-	ALi M1543C PCI ISA bridge
-	0	19	0	-	ALi M1543C PCI USB adapter

The IRQ stands for bit position in DRIR of Tsunami/Typhoon chip. It has nothing to do with "ISA" style interrupts which are routed to IRQ 55 (including ALi M1543C PCI IDE/ATAPI controller).

So far the CHARON-AXP emulators do not support virtual ALi M1543C PCI USB adapter. So position of the device 19, function 0 on the PCI 0 remains empty.

6.3.15. AlphaServer ES45 (2 Pchips, 10 PCI slots)

In addition to 10 PCI vacant slots there are 3 PCI positions occupied by on-board devices. All 13 PCI positions are listed in the following table in the order in which Alpha SRM console enumerates them.

Slot	PCI	Device	Function	IRQ	
pci_0					
-	0	7	0	-	ALi M1543C PCI ISA bridge
1	0	8	0	20	<i>option</i>
2	0	9	0	24	<i>option</i>
3	0	10	0	12	<i>option</i>

Slot	PCI	Device	Function	IRQ	
4	0	11	0	16	<i>option</i>
-	0	16	0	-	ALi M1543C PCI IDE/ATAPI controller (DQA, DQB)
-	0	19	0	-	ALi M1543C PCI USB adapter
pci_1					
5	1	1	0	28	<i>option</i>
6	1	2	0	32	<i>option</i>
pci_2					
7	2	1	0	8	<i>option</i>
8	2	2	0	44	<i>option</i>
pci_3					
9	3	1	0	36	<i>option</i>
10	3	2	0	40	<i>option</i>

The IRQ stands for bit position in DRIR of TITAN chip. It has nothing to do with "ISA" style interrupts which are routed to IRQ 55 (including ALi M1543C PCI IDE/ATAPI controller).

So far the CHARON-AXP emulators do not support virtual ALi M1543C PCI USB adapter. So position of the device 19, function 0 on the PCI 0 remains empty.

6.3.16. AlphaServer GS80 (2 QBBs, 8 PCI busses, 27 PCI slots)

Slot	PCI	Device	Function	IRQ	
qbb_0_pca_0_pci_0					
0/1	0	1	0	36	QLOGIC ISP1040B PCI SCSI Adapter
2	0	2	0	40	<i>option</i>
3	0	3	0	44	<i>option</i>
-	0	7	0	-	ALi M1543C PCI ISA bridge
-	0	15	0	-	ALi M1543C PCI IDE/ATAPI controller
-	0	19	0	-	ALi M1543C PCI USB adapter
qbb_0_pca_0_pci_1					
4	1	4	0	48	<i>option</i>
5	1	5	0	52	<i>option</i>
6	1	6	0	56	<i>option</i>
7	1	7	0	60	<i>option</i>
qbb_0_pca_1_pci_0					
0/1	2	0	0	32	<i>option</i>
2	2	2	0	40	<i>option</i>
3	2	3	0	44	<i>option</i>
qbb_0_pca_1_pci_1					
4	3	4	0	48	<i>option</i>
5	3	5	0	52	<i>option</i>

Slot	PCI	Device	Function	IRQ	
6	3	6	0	56	<i>option</i>
7	3	7	0	60	<i>option</i>
qbb_1_pca_0_pci_0					
0/1	8	0	0	32	<i>option</i>
2	8	2	0	40	<i>option</i>
3	8	3	0	44	<i>option</i>
qbb_1_pca_0_pci_1					
4	9	4	0	48	<i>option</i>
5	9	5	0	52	<i>option</i>
6	9	6	0	56	<i>option</i>
7	9	7	0	60	<i>option</i>
qbb_1_pca_1_pci_0					
0/1	10	0	0	32	<i>option</i>
2	10	2	0	40	<i>option</i>
3	10	3	0	44	<i>option</i>
qbb_1_pca_1_pci_1					
4	11	4	0	48	<i>option</i>
5	11	5	0	52	<i>option</i>
6	11	6	0	56	<i>option</i>
7	11	7	0	60	<i>option</i>

PCI 2 and 3 on each QBB are not populated.

So far the CHARON-AXP emulators do not support virtual ALi M1543C PCI USB adapter. So position of the device 19, function 0 on the PCI 0 remains empty.

Total number of PCI devices configured through CFG file may not exceed 27.

6.3.17. AlphaServer GS160 (4 QBBs, 16 PCI busses, 55 PCI slots)

Slot	PCI	Device	Function	IRQ	
qbb_0_pca_0_pci_0					
0/1	0	1	0	36	<i>QLOGIC ISP1040B PCI SCSI Adapter</i>
2	0	2	0	40	<i>option</i>
3	0	3	0	44	<i>option</i>
-	0	7	0	-	ALi M1543C PCI ISA bridge
-	0	15	0	-	ALi M1543C PCI IDE/ATAPI controller
-	0	19	0	-	ALi M1543C PCI USB adapter
qbb_0_pca_0_pci_1					
4	1	4	0	48	<i>option</i>
5	1	5	0	52	<i>option</i>

Slot	PCI	Device	Function	IRQ	
6	1	6	0	56	<i>option</i>
7	1	7	0	60	<i>option</i>
qbb_0_pca_1_pci_0					
0/1	2	0	0	32	<i>option</i>
2	2	2	0	40	<i>option</i>
3	2	3	0	44	<i>option</i>
qbb_0_pca_1_pci_1					
4	3	4	0	48	<i>option</i>
5	3	5	0	52	<i>option</i>
6	3	6	0	56	<i>option</i>
7	3	7	0	60	<i>option</i>
qbb_1_pca_0_pci_0					
0/1	8	0	0	32	<i>option</i>
2	8	2	0	40	<i>option</i>
3	8	3	0	44	<i>option</i>
qbb_1_pca_0_pci_1					
4	9	4	0	48	<i>option</i>
5	9	5	0	52	<i>option</i>
6	9	6	0	56	<i>option</i>
7	9	7	0	60	<i>option</i>
qbb_1_pca_1_pci_0					
0/1	10	0	0	32	<i>option</i>
2	10	2	0	40	<i>option</i>
3	10	3	0	44	<i>option</i>
qbb_1_pca_1_pci_1					
4	11	4	0	48	<i>option</i>
5	11	5	0	52	<i>option</i>
6	11	6	0	56	<i>option</i>
7	11	7	0	60	<i>option</i>
qbb_2_pca_0_pci_0					
0/1	16	0	0	32	<i>option</i>
2	16	2	0	40	<i>option</i>
3	16	3	0	44	<i>option</i>
qbb_2_pca_0_pci_1					
4	17	4	0	48	<i>option</i>
5	17	5	0	52	<i>option</i>
6	17	6	0	56	<i>option</i>
7	17	7	0	60	<i>option</i>
qbb_2_pca_1_pci_0					
0/1	18	0	0	32	<i>option</i>
2	18	2	0	40	<i>option</i>

Slot	PCI	Device	Function	IRQ	
3	18	3	0	44	<i>option</i>
qbb_2_pca_1_pci_1					
4	19	4	0	48	<i>option</i>
5	19	5	0	52	<i>option</i>
6	19	6	0	56	<i>option</i>
7	19	7	0	60	<i>option</i>
qbb_3_pca_0_pci_0					
0/1	24	0	0	32	<i>option</i>
2	24	2	0	40	<i>option</i>
3	24	3	0	44	<i>option</i>
qbb_3_pca_0_pci_1					
4	25	4	0	48	<i>option</i>
5	25	5	0	52	<i>option</i>
6	25	6	0	56	<i>option</i>
7	25	7	0	60	<i>option</i>
qbb_3_pca_1_pci_0					
0/1	26	0	0	32	<i>option</i>
2	26	2	0	40	<i>option</i>
3	26	3	0	44	<i>option</i>
qbb_3_pca1_pci_1					
4	27	4	0	48	<i>option</i>
5	27	5	0	52	<i>option</i>
6	27	6	0	56	<i>option</i>
7	27	7	0	60	<i>option</i>

PCI 2 and 3 on each QBB are not populated.

So far the CHARON-AXP emulators do not support virtual ALi M1543C PCI USB adapter. So position of the device 19, function 0 on the PCI 0 remains empty.

Total number of PCI devices configured through CFG file may not exceed 27.

6.3.18. AlphaServer GS320 (8 QBBs, 32 PCI busses, 111 PCI slots)

Slot	PCI	Device	Function	IRQ	
qbb_0_pca_0_pci_0					
0/1	0	1	0	36	<i>QLOGIC ISP1040B PCI SCSI Adapter</i>
2	0	2	0	40	<i>option</i>
3	0	3	0	44	<i>option</i>
-	0	7	0	-	<i>ALi M1543C PCI ISA bridge</i>
-	0	15	0	-	<i>ALi M1543C PCI IDE/ATAPI controller</i>

Slot	PCI	Device	Function	IRQ	
-	0	19	0	-	ALi M1543C PCI USB adapter
qbb_0_pca_0_pci_1					
4	1	4	0	48	<i>option</i>
5	1	5	0	52	<i>option</i>
6	1	6	0	56	<i>option</i>
7	1	7	0	60	<i>option</i>
qbb_0_pca_1_pci_0					
0/1	2	0	0	32	<i>option</i>
2	2	2	0	40	<i>option</i>
3	2	3	0	44	<i>option</i>
qbb_0_pca_1_pci_1					
4	3	4	0	48	<i>option</i>
5	3	5	0	52	<i>option</i>
6	3	6	0	56	<i>option</i>
7	3	7	0	60	<i>option</i>
qbb_1_pca_0_pci_0					
0/1	8	0	0	32	<i>option</i>
2	8	2	0	40	<i>option</i>
3	8	3	0	44	<i>option</i>
qbb_1_pca_0_pci_1					
4	9	4	0	48	<i>option</i>
5	9	5	0	52	<i>option</i>
6	9	6	0	56	<i>option</i>
7	9	7	0	60	<i>option</i>
qbb_1_pca_1_pci_0					
0/1	10	0	0	32	<i>option</i>
2	10	2	0	40	<i>option</i>
3	10	3	0	44	<i>option</i>
qbb_1_pca_1_pci_1					
4	11	4	0	48	<i>option</i>
5	11	5	0	52	<i>option</i>
6	11	6	0	56	<i>option</i>
7	11	7	0	60	<i>option</i>
qbb_2_pca_0_pci_0					
0/1	16	0	0	32	<i>option</i>
2	16	2	0	40	<i>option</i>
3	16	3	0	44	<i>option</i>
qbb_2_pca_0_pci_1					
4	17	4	0	48	<i>option</i>
5	17	5	0	52	<i>option</i>
6	17	6	0	56	<i>option</i>

Slot	PCI	Device	Function	IRQ	
7	17	7	0	60	<i>option</i>
qbb_2_pca_1_pci_0					
0/1	18	0	0	32	<i>option</i>
2	18	2	0	40	<i>option</i>
3	18	3	0	44	<i>option</i>
qbb_2_pca_1_pci_1					
4	19	4	0	48	<i>option</i>
5	19	5	0	52	<i>option</i>
6	19	6	0	56	<i>option</i>
7	19	7	0	60	<i>option</i>
qbb_3_pca_0_pci_0					
0/1	24	0	0	32	<i>option</i>
2	24	2	0	40	<i>option</i>
3	24	3	0	44	<i>option</i>
qbb_3_pca_0_pci_1					
4	25	4	0	48	<i>option</i>
5	25	5	0	52	<i>option</i>
6	25	6	0	56	<i>option</i>
7	25	7	0	60	<i>option</i>
qbb_3_pca_1_pci_0					
0/1	26	0	0	32	<i>option</i>
2	26	2	0	40	<i>option</i>
3	26	3	0	44	<i>option</i>
qbb_3_pca1_pci_1					
4	27	4	0	48	<i>option</i>
5	27	5	0	52	<i>option</i>
6	27	6	0	56	<i>option</i>
7	27	7	0	60	<i>option</i>
qbb_4_pca_0_pci_0					
0/1	32	0	0	32	<i>option</i>
2	32	2	0	40	<i>option</i>
3	32	3	0	44	<i>option</i>
qbb_4_pca_0_pci_1					
4	33	4	0	48	<i>option</i>
5	33	5	0	52	<i>option</i>
6	33	6	0	56	<i>option</i>
7	33	7	0	60	<i>option</i>
qbb_4_pca_1_pci_0					
0/1	34	0	0	32	<i>option</i>
2	34	2	0	40	<i>option</i>
3	34	3	0	44	<i>option</i>

Slot	PCI	Device	Function	IRQ	
qbb_4_pca_1_pci_1					
4	35	4	0	48	<i>option</i>
5	35	5	0	52	<i>option</i>
6	35	6	0	56	<i>option</i>
7	35	7	0	60	<i>option</i>
qbb_5_pca_0_pci_0					
0/1	40	0	0	32	<i>option</i>
2	40	2	0	40	<i>option</i>
3	40	3	0	44	<i>option</i>
qbb_5_pca_0_pci_1					
4	41	4	0	48	<i>option</i>
5	41	5	0	52	<i>option</i>
6	41	6	0	56	<i>option</i>
7	41	7	0	60	<i>option</i>
qbb_5_pca_1_pci_0					
0/1	42	0	0	32	<i>option</i>
2	42	2	0	40	<i>option</i>
3	42	3	0	44	<i>option</i>
qbb_5_pca_1_pci_1					
4	43	4	0	48	<i>option</i>
5	43	5	0	52	<i>option</i>
6	43	6	0	56	<i>option</i>
7	43	7	0	60	<i>option</i>
qbb_6_pca_0_pci_0					
0/1	48	0	0	32	<i>option</i>
2	48	2	0	40	<i>option</i>
3	48	3	0	44	<i>option</i>
qbb_6_pca_0_pci_1					
4	49	4	0	48	<i>option</i>
5	49	5	0	52	<i>option</i>
6	49	6	0	56	<i>option</i>
7	49	7	0	60	<i>option</i>
qbb_6_pca_1_pci_0					
0/1	50	0	0	32	<i>option</i>
2	50	2	0	40	<i>option</i>
3	50	3	0	44	<i>option</i>
qbb_6_pca_1_pci_1					
4	51	4	0	48	<i>option</i>
5	51	5	0	52	<i>option</i>
6	51	6	0	56	<i>option</i>
7	51	7	0	60	<i>option</i>

Slot	PCI	Device	Function	IRQ	
qbb_7_pca_0_pci_0					
0/1	56	0	0	32	<i>option</i>
2	56	2	0	40	<i>option</i>
3	56	3	0	44	<i>option</i>
qbb_7_pca_0_pci_1					
4	57	4	0	48	<i>option</i>
5	57	5	0	52	<i>option</i>
6	57	6	0	56	<i>option</i>
7	57	7	0	60	<i>option</i>
qbb_7_pca_1_pci_0					
0/1	58	0	0	32	<i>option</i>
2	58	2	0	40	<i>option</i>
3	58	3	0	44	<i>option</i>
qbb_7_pca_1_pci_1					
4	59	4	0	48	<i>option</i>
5	59	5	0	52	<i>option</i>
6	59	6	0	56	<i>option</i>
7	59	7	0	60	<i>option</i>

PCI 2 and 3 on each QBB are not populated in emulator.

So far the CHARON-AXP emulators do not support virtual ALi M1543C PCI USB adapter. So position of the device 19, function 0 on the PCI 0 remains empty.

Total number of PCI devices configured through CFG file may not exceed 27.

6.4. Multi instance support

6.4.1. General description

CHARON supports several instances of the emulators running simultaneously on the same host. Number of instances allowed to run simultaneously is encoded into CHARON license key.

In order to run several instances simultaneously, please note the following steps:

1. The host system must have enough CPU cores and memory to cover the requirements of all the instances at the same time.

Each virtual CPU occupies one host CPU, so the total number of CPUs should be greater than a sum of all the emulated CPUs. Note that some CPUs needs to be used for I/O processing and at least one CPU – for the operating system housekeeping. Thus the total amount of the host CPUs depends on the number of the CPUs needed for I/O. The general recommendation is to leave at least 1/3 of the CPUs available to an instance for the instance I/O, but depending on data flow this number can be increased / decreased for each instance separately.

The minimal host memory is calculated as a sum of emulated memory of each CHARON instance plus at least 2 GB of additional memory.

2. Each instance must have its own configuration and log files, toy etc containers. Configuration file of each CHARON instance should exactly specify the following:
 - The number of CPUs to emulate (“**n_of_cpus**”). By default this parameter is equal to the number of the CPUs the particular emulated model supports. But this number can be reduced by changing the parameter or by the license restrictions
 - The number of CPUs chosen for I/O operations (“**n_of_io_cpus**”). By default this parameter is equal to 1/3 of the CPUs available for certain emulator (round by 1). It is possible to dedicate a chosen number of CPUs for I/O processing in case of intensive or, in opposite case, very shallow data flow.
 - Number of the CPUs the instance allocates. By default CHARON-AXP instance grabs as many CPUs as possible. To balance the number of host CPUs between different instances a special parameter “**affinity**” is provided. This parameter specifies what CPUs in particular each instance can allocate.

Using those 3 parameters it is possible to balance the hosting server resources for all running CHARON instances.

3. Each instance must use its own specific console port.
4. Once the configuration files are updated for each particular instance CHARON, it is recommended to test those configurations separately.

6.4.2. Running several instances of CHARON

At the moment the Launcher utility does not support multi-instance configurations, so it is necessary to select one of the following startup methods:

- Run each instance from a shortcut. In this case a shortcut should contain a proper CHARON configuration file name for each instance.
- Run each instance as a separate Windows Service.
- Run each instance from the command line. Open up the *cmd.exe* (**Start->Run**, type “**cmd.exe**”, **cd** to the directory where CHARON executables are installed, type “<**name of the executable**> <**name of the configuration file**>” and press **Enter**. Repeat the last actions for all the instances to be run in the same time. Note that the configuration files must be different for each one!

Example 6.3. CHARON-AXP

```
>gs160.exe gs160_first.cfg
```

```
>gs160.exe gs160_second.cfg
```

Example 6.4. CHARON-VAX

```
>mv3k6.exe mv3k6_first.cfg
```

```
>mv3k6.exe mv3k6_second.cfg
```

Please refer to **set session_name** parameter in order to name the CHARON instances

6.5. General configuration parameters

Some **set** parameters do not require the load counterpart; they set general CHARON instance parameters influencing startup and logging:

6.5.1. Common parameters

"set session" parameters	Type	Value
configuration_name	Text string	<p>A string specifying the name of the session (instance). This name will be showed if scrolling over the icon in the taskbar notification area.</p> <p>Example 6.5.</p> <p>set session configuration_name="MSCDV1"</p> <p>The value of this parameter is used as a prefix to the event log file name in case if the multiple log files approach is chosen.</p> <p>Example 6.6.</p> <p>set session configuration_name="SERVER"</p> <p>in this case the log file will have the following form:</p> <p><i>AS400-YYYY-MM-DD-hh-mm-ss-xxxxxxxxx.log</i></p> <p>xxxxxxxxx is an increasing decimal number starting from 000000000 to separate log files with the same time of creation (in case if the log is being written faster than one log file per second).</p>
log	Text string	<p>A string specifying a file name to store the log of the session or a directory where the log files for each session will be stored.</p> <p>If an existing directory is specified, CHARON automatically enables creation of individual log file for each session. If the log parameter is omitted CHARON will create logs for each session in the current directory.</p> <p>Example 6.7.</p> <p>set session log="log.txt"</p> <p>set session log="C:\CHARON Logs"</p> <p>In case if just a directory specified in the log parameter and the configuration_name parameter of the session is specified too the log file name is composed as follows:</p> <p><i>configuration_name-YYYY-MM-DD-hh-mm-ss-xxxxxxxxx.log</i></p> <p>otherwise if the configuration_name parameter is omitted, the log name will have the following format:</p>

"set session" parameters	Type	Value
		<p><i>hw_model-YYYY-MM-DD-hh-mm-ss-xxxxxxxx.log</i></p> <p>xxxxxxxx is an increasing decimal number starting from 00000000 to separate log files with the same time of creation (in case if the log is being written faster than one log file per second).</p>
log_method	Text string	<p>"overwrite" (default) or "append". Determines if previous log information is maintained. Note that this parameter must be specified only in addition to the "log" parameter on the same line with it.</p> <p>This parameter is applicable only if the log is stored in exactly specified file.</p> <p>Example 6.8.</p> <p>set session log="log.txt" log_method="append"</p>
log_show_messages	Text string	<p>Defines the message types that should be shown. The parameter is a string of comma delimited words: "all", "info", "warning" and "error" which defines which message types should be logged. The default value is "all" message types.</p>
log_repeat_filter	Text string	<p>Specifies if repeated messages should be filtered or not. Possible values are "on" (default) and "off".</p> <p>If the value is "on", immediately following messages with the same identifier and system error code are not listed in the log, but they are counted. When a different log message is generated, the repeat count of the earlier log message is reported with "The previous message has been repeated N times.", and the counter is cleared.</p>
log_locale	Text string	<p>Sets the language of message database. So far the following values are supported:</p> <ul style="list-style-type: none"> • "Dutch" • "English" • "Swedish" • "Spanish" • "Chinese-Simplified" <p>By default it is set to "English". If specified an unsupported value, "English" is used.</p> <p>Example 6.9.</p> <p>set session log_locale="Dutch"</p>
license_key_id[N], N=0 or 1	Numeric	<p>A number (decimal Sentinel key ID) specifying the regular (N=0) and backup (N=1) license key to be used by CHARON.</p>

"set session" parameters	Type	Value
		<p>Example 6.10.</p> <p>set session license_key_id[0] = 1877752571</p> <p>set session license_key_id[1] = 354850588</p> <p>it is also possible to specify both regular and backup key in one line:</p> <p>set session license_key_id[0] = 1877752571 li- cense_key_id[1] = 354850588</p> <p>Depending on presence of the regular and/or backup license key IDs in the configuration file CHARON behaves differently:</p> <ol style="list-style-type: none"> 1. No keys are specified CHARON behaves as usual (performs unqualified search for any suitable key). If no keys are found, CHARON exits. 2. Both keys are specified CHARON performs qualified search for regular license key. If it is not found, CHARON performs qualified search for backup license key. If it is not found, CHARON exits. 3. Only regular key is specified CHARON performs qualified search for regular license key. If it is not found, CHARON performs unqualified search for any suitable key. If it is not found, CHARON exits. 4. Only backup key is specified CHARON behaves as usual (performs unqualified search for any suitable key). If no keys are found, CHARON exits.
affinity	Text string	<p>Overrides initial process's affinity mask provided by host operating system.</p> <p>Once specified it allows binding the running instance of emulator to particular host CPUs. Might be used for soft partitioning host CPU resources, for isolating host CPUs for other applications.</p> <p>By default the emulator instance allocates as many host CPUs as possible. The "affinity" overrides that and allows explicit specification on which host CPU the instance shall run.</p> <p>Host CPUs are enumerated as comma separated list of host system assigned CPU numbers.</p> <p>Example 6.11.</p> <p>set session affinity="0, 2, 4, 6"</p>
n_of_io_cpus	Numeric	<p>Says how many host CPUs (of those specified by "affinity" parameter, if any) the emulator shall use for I/O handling.</p>

"set session" parameters	Type	Value
		<p>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 that by specifying number of I/O host CPUs explicitly</p> <p>Example 6.12.</p> <pre>set session n_of_io_cpus=2</pre>

6.5.2. Specific configuration parameters

Set parameters for the session	Type	Value
hw_model	Text string	<p>Virtual HP Alpha system hardware model for which the configuration file is created.</p> <p>Tip: use a default configuration template for a particular model as a starting point for a custom configuration. This would ensure that the parameter is set correctly.</p> <p>Example 6.13.</p> <pre>set session hw_model="AlphaServer_DS20"</pre> <p>See Appendix section for proper combination of "cpu_architecture", "hw_model" and "dsrdb[0]" parameters according to the chosen "system_name" value.</p>
n_of_cpus	Numeric	<p>Limits number of emulated CPUs.</p> <p>Example 6.14.</p> <pre>set session n_of_cpus=3</pre> <p>Maximum number of CPUs enabled by CHARON is specified by the license key, but cannot exceed the original hardware restrictions:</p> <ul style="list-style-type: none"> • AlphaServer_AS400 – 1 CPU • AlphaServer_AS800 – 1 CPU • AlphaServer_AS1000 – 1 CPU • AlphaServer_AS1000A – 1 CPU • AlphaServer_AS1200 – 2 CPUs • AlphaServer_AS2000 – 2 CPUs • AlphaServer_AS2100 – 4 CPUs • AlphaServer_AS4000 – 2 CPUs • AlphaServer_AS4100 – 4 CPUs

Set parameters for the session	Type	Value
		<ul style="list-style-type: none"> • AlphaServer_DS10L – 1 CPU • AlphaServer_DS15 – 1 CPU • AlphaServer_DS20 – 2 CPUs • AlphaServer_DS25 – 2 CPUs • AlphaServer_ES40 – 4 CPUs • AlphaServer_ES45 – 4 CPUs • AlphaServer_GS80 – 8 CPUs • AlphaServer_GS160 – 16 CPUs • AlphaServer_GS320 – 32 CPUs <p>At startup emulator adjusts the number of emulated CPUs accordingly to the number of available host CPU cores (enabled by “affinity” if any).</p> <p>This option overrides automatic adjustment.</p> <p><i>Note that in any case emulator reserves at least one host CPU core for I/O management, so that given N host CPU cores emulator supports up-to N-1 emulated CPUs.</i></p>

6.5.3. Examples

Example 6.15. hw_model example

```
set session hw_model="AlphaServer_ES40"
```

This command specifies HP Alpha Server model the configuration file is designed for. It must be the first command in a configuration file. Various CHARON products create specific virtual CPU models and have different configuration commands. This command helps to detect errors and prevents execution in case an incorrect virtual model is started. If the **set session hw_model="...."** statement is not found, the configuration file is ignored, and the emulator will not be activated.

Example 6.16. Logging

```
set session log="clipper.log" log_method="append"
```

Creates a log file in the directory where CHARON starts. Specify the full path to locate the log file elsewhere. The specified log file is created or overwritten at each start depending on the log_method parameter. The **log_method** parameter must be specified on the same line with the “log” parameter.

6.6. Console interface

Virtual HP Alpha system supports one serial console port, which in CHARON-AXP is identified with the logical name **OPA0**. To use the **OPA0** a physical or virtual serial line connection must be loaded in the configuration file.

Emulated Alpha models AS400 and DS10L also have a second console port, **TTA0**

Except for the console port(s), the virtual Alpha could provide additional physical serial ports with emulated PBXDA-xx adapter in Pass Through mode.

Terminals can also be connected to CHARON-AXP via TCP/IP or LAT terminal servers.

6.6.1. Types of serial line emulation

Type	Function
physical_serial_line	This command associates a COM port in the Windows host system with the OPA0 console port. The COM port can be a physical port part of the host system hardware or a logical COM port as created by, for example, an Ethernet serial port device.
virtual_serial_line	This command associates a network connection in the Windows host system with the OPA0 console port.

Example 6.17.

```
load physical_serial_line OPA0
```

6.6.2. "physical_serial_line" parameters

Parameter	Type	Description
line	Text string	"\\.\COMn" A defined COM port on the Windows host system.
baud	Numeric	Forces the baud rate of the corresponding COM port to the specified value. Variety of supported values depends on underlying physical communication resource (COM port that is). The most widely used values are: 300, 1200, 9600, 19200, 38400. Example 6.18. load physical_serial_line OPA0 baud=38400
log	Text string	A string specifying a file name to store content of the OPA0 or TT0 sessions or a directory where the log files for each OPA0 or TT0 individual session will be stored. If an existing directory is specified, CHARON automatically enables creation of individual log file for each OPA0 or TT0 session. If the log parameter is omitted CHARON will not create any console log. Example 6.19. set OPA0 log="log.txt" set OPA0 log="C:\CHARON Logs\Session Logs"
break_on	Text string	Defines a key (combination of keys) for the break operation.

Parameter	Type	Description
		<p>It works only for console line (For CHARON-VAX is it is the only line of the UART and the <i>line[3]</i> of QUART).</p> <ul style="list-style-type: none"> • "<i>Ctrl-P</i>" or/and "<i>Break</i>" – specify one of them or both separating with comma (",") • "<i>none</i>" – No break key is defined <p>The default value is "<i>Break</i>" for line 3 of QUART and "<i>none</i>" – for other lines.</p>

Provided that the physical serial line connects a terminal to CHARON, pressing the "**Break**" button on the terminal's keyboard will generate a SPACE condition on the serial line. Setting the **break_on** parameter value to "*Break*" in the configuration file will trigger the HALT (Reset) condition in CHARON upon detection of the SPACE condition on the associated COM port.

Set the **break_on** parameter value to "*Ctrl-P*" to trigger the HALT condition in the emulated Alpha/VAX/PDP-11 upon reception of **Ctrl-P** character (ASCII code 10 (hex)).

The *break_on* parameter is ignored for all the lines except the console line.

Example 6.20. Alpha Configuration

```
load physical_serial_line OPA0
```

```
set OPA0 line="\\.\COM1"
```

or in a more compact form:

```
load physical_serial_line OPA0 line="\\.\COM1"
```

Example 6.21. VAX Configuration

```
load physical_serial_line/chserial DEF
```

```
set DEF break_on="Ctrl-P,Break" line="\\.\COM2"
```

```
set quart line[3]=DEF
```

Note

In the examples above DEF is a logical name for the serial line emulation. That name is only used as a reference within a configuration file. It has no influence on the naming of the devices inside an Alpha/VAX/PDP-11 operating system. The names used can be helpful identifiers, use any character string you wish.

6.6.3. "virtual_serial_line" parameters

Parameter	Type	Description
host	Text string	<p>The remote host's IP address or host name and optionally remote TCP/IP port number for the virtual serial line to connect to. If omitted, the virtual serial line does not initiate connection to remote host while still listening for incoming connection requests.</p> <p>Specify the value in the following form:</p>

Parameter	Type	Description
		<p>host="<host-name>[:<port-no>]"</p> <p>If the <port-no> is not specified the virtual serial line uses TCP/IP port number specified by the "port" parameter (see below).</p>
port	Numeric	TCP/IP port number for the virtual serial line. The virtual serial line always listens on this port for incoming connection requests.
application	Text string	An application to run. At startup CHARON starts the specified application. For example a 3rd party terminal emulator connecting to a specified port. Specify a file known to the operating system. E.G. an *.ht file to invoke Putty terminal emulator with the parameters stored within that file.
break_on	Text string	<p>Specifies which byte sequences received over physical serial line shall trigger HALT command with switching to CHARON SRM console.</p> <p>Specify the following values: "<i>Crtl-P</i>", "<i>F5</i>", "<i>Break</i>" or "none" to disable triggering HALT condition.</p> <p>Example 6.22.</p> <p>break_on="Crtl-P"</p> <p>The default value is "<i>F5</i>" and "<i>Break</i>"</p>
stop_on	Text string	<p>Specifies which byte sequences received over virtual serial line shall trigger STOP condition. The STOP condition causes CHARON to terminate.</p> <p>Specify value as a comma separated combination of the following: "Application", "F6", or as "none" to disable triggering STOP condition.</p> <p>Example 6.23.</p> <p>stop_on="Application,F6"</p> <p>The default value is "none".</p> <p>Set to "Application" to trigger the STOP condition when the associated application terminates. Use this option only for virtual_serial_lines configured for automatic application invocation (where the Application parameter specifies a valid application).</p> <p>Set to "F6" to trigger the STOP condition upon reception of the sequence "<ESC>[17~". Terminal emulators may send these sequences when pressing the F6 button</p>
log	Text string	<p>A string specifying a file name to store content of the OPA0 or TT0 sessions or a directory where the log files for each OPA0 or TT0 individual session will be stored.</p> <p>If an existing directory is specified, CHARON automatically enables creation of individual log file for each</p>

Parameter	Type	Description
		<p>OPA0 or TT0 session. If the log parameter is omitted CHARON will not create any console log.</p> <p>Example 6.24.</p> <pre>set OPA0 log="log.txt" set OPA0 log="C:\CHARON Logs\Session Logs"</pre>

Example 6.25. Defining a local Putty terminal emulator session as the serial console terminal:

```
load virtual_serial_line OPA0 break_on="Ctrl-P,F5"
set OPA0 port=10003 stop_on="F6,Application"
set OPA0 application="putty.exe -load OPA0"
```

A more compact form is also possible:

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

The **OPA0** is a parameter for the Putty terminal emulator for loading settings for this particular session from its registry base.

Notes on the `virtual_serial_line` option:

1. Use the combination of parameters `port` and `application` as follows to start a 3rd party terminal emulator or similar program.

Example 6.26.

```
load virtual_serial_line/chserial TTA0 port=10000 application="putty.exe -load OPA0"
```

In this example CHARON creates port 10000 and waits for a connection. Then it immediately starts *Putty.exe* (using the configuration OPA0), which will connect to the port 10000. To connect CHARON to a port on a specific host use the parameters `host` and `port`:

Example 6.27.

```
load virtual_serial_line/chserial TTA0 host="192.168.1.1" port=10000
```

In this example CHARON connects to port 10000 of the host with TCP/IP address 192.168.1.1 and at the same time it accepts connections on local port 10000.

2. It is also possible to specify port on a remote host (note that CHARON always acts as a server). The syntax is:

Example 6.28.

```
load virtual_serial_line/chserial TTA0 host="192.168.1.1:20000" port=10000
```

In this example CHARON will accept connection on local port 10000 and connects to remote port 20000 of the host 192.168.1.1

Note that the two last examples are mainly used for inter-CHARON communications. They are used to connect CHARON to an application that communicates to CHARON as described below.

Example 6.29. Two CHARON systems are 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"
```

Both hosts execute CHARON, the two TXA0 lines connect to each other, thus creating a "serial" cable between the two emulated Alphas, VAXes/PDPs. The order in which the instances of CHARON are started makes no difference.

6.7. Specifying emulated memory

6.7.1. Syntax

The memory subsystem is permanently loaded and has the logical name **ram**. The effective amount of memory is determined in steps, starting with the **set ram size** statement in the configuration file.

ram parameter	Type	Description
size	Numeric	Size of emulated memory in MB.

Example 6.30.

```
set ram size = 512
```

creates 512 MB of emulated memory

Note

- Where applicable, the memory is capped to the maximum as defined in the CHARON license key
- In addition, CHARON generates an error message in the log file and reduce its effective memory size if the host system cannot allocate enough memory to map the requested emulated memory

6.7.2. Parameters of emulated RAM for various hardware models of virtual HP Alpha system

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

Hardware Model	RAM size (in MB)			
AlphaServer 4100	64	32768	512	64
AlphaServer DS10L	64	32768	512	64
AlphaServer DS15	64	32768	512	64
AlphaServer DS20	64	32768	512	64
AlphaServer DS25	64	32768	512	64
AlphaServer ES40	64	32768	512	64
AlphaServer ES45	64	32768	512	64
AlphaServer GS80	256	65536	512	256
AlphaServer GS160	512	131072	512	512
AlphaServer GS320	1024	262144	1024	1024

Note

If no set ram statement is found, the memory size is set to 512MB, except for the AlphaServer_GS320 for which it is set to 1024MB

6.8. System time and date

The virtual system maintains its time and date via **TOY** (time-of-year) component. In order to preserve time and date while virtual system is not running the **TOY** component uses a binary file on the host system. A name of the file is specified by "*container*" option of the **TOY** component.

ram parameter	Type	Description
container	Text string	Specifies the name of file in which the virtual system preserves its time and date during "offline" period. By default it is left unspecified.
sync_to_host	Text string	Specifies whether and how the guest OS time is synchronized with the CHARON host time. <i>Syntax:</i> set TOY sync_to_host = "{as_vms as_tru64 as_is}[, nowrite]" where: <ul style="list-style-type: none"> • as_vms - If the guest OS is OpenVMS/AXP and its date and time must be set to the host's date and time each time it boots • as_tru64 - If the guest OS is Tru64 UNIX and its date and time must be set to the host's date and time each time it boots • as_is - If the TOY date and time must be set to the host's UTC date and time • nowrite - Disable undesirable updates to the TOY from the guest OS.

ram parameter	Type	Description
		<p>Example 6.31.</p> <p>set TOY sync_to_host = "as_vms, nowrite"</p> <p>To synchronize the guest OS with TOY, use the following commands:</p> <p><i>On OpenVMS/AXP:</i></p> <p>\$ set time</p> <p><i>On Tru64 UNIX:</i></p> <p># date -u `consvar -g date cut -f 3 -d ' '`</p> <p>The default value is <i>not specified</i> - it means that by default CHARON does not synchronize its guest OS time with the CHARON host time, but collect date and time from the file specified with "container" parameter.</p>

Example 6.32.

```
set TOY container="my_virtual_system.dat"
```

```
set TOY sync_to_host="as_vms"
```

The virtual system may have its time and date different from system time and date of the host system, but relies on correctness of the host's system time and date to calculate duration of "offline" period (i.e. while virtual system is not running).

6.9. Virtual HP Alpha SRM console environment

Virtual HP Alpha system implements a subset of Alpha SRM console environment according to Alpha Architecture Reference Manual. Virtual HP Alpha SRM console environment is a part of virtual HP Alpha ROM (which also contains virtual HP Alpha firmware).

6.9.1. Firmware and console environment parameters

In order to preserve console environment settings (such as default boot device, boot OS flags, boot file name, etc.) while virtual system is not running, the ROM component ("**rom**") uses 2MB binary file on the host system. Name of the file is specified by "**container**" option of the ROM component.

rom parameter	Type	Description
container	Text string	<p>Specifies the name of file in which the virtual system preserves its firmware image and console environment during "offline" period.</p> <p>By default it is left unspecified.</p>
system_name	Text string	This parameter allows changing the system name.
system_serial_number	Text string	<p>This parameter allows changing the system serial number, for example:</p> <p>set rom system_serial_number = NY12345678</p>

rom parameter	Type	Description
		Any sequence of characters can be used as a serial number. Sequences longer than 16 symbols are truncated to 16 symbols. Serial Numbers should be according to DEC standard: 10 characters. First two characters are capital letters, remaining 8 characters are decimal digits. By default it is set to SN01234567
dsrdb[0], dsrdb[1]	Numeric	DSRB - Dynamic System Recognition Data Block. These parameters allow changing the emulated hardware model type
version	Text string	Sets Console and PALcode versions in the following way: 1. Set SRM Console version to X.Y-Z: set rom version[0] = x.y-z 2. Set OpenVMS PALcode version to X.Y-Z: set rom version[1] = x.y-z 3. Set Tru64 UNIX PALcode version to X.Y-Z: set rom version[2] = x.y-z

Example 6.33.

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

```
set rom system_name="Alpha Server 1000 4/200"
```

```
set rom dsrdb[0]=1090
```

```
set rom version[0] = 7.3-1 version[1] = 1.98-104 version[2] = 1.92-105
```

The same file also carries copy of virtual HP Alpha/VAX/PDP11 etc firmware. Each new version (new build number is considered as new version too) of CHARON software updates the firmware preserved in the file thus clearing console environment variables.

See Appendix section for proper combination of "cpu_architecture", "hw_model", "dsrdb[0]" and "dsrdb[1]" parameters according to the chosen "system_name" value.

6.10. CPU Architecture

The virtual Alpha CPU architecture can be configured in the following way:

ace parameter	Type	Description
cpu_architecture	Identifier	Specifies the architecture of the virtual Alpha CPU. Can be one of the following: EV4, EV45, EV5, EV56, EV6, EV67, EV68

Example 6.34.

```
set ace cpu_architecture = EV6
```

Apart from that, nice to keep the System Manufacturing Model (SMM, a number identifying a particular Alpha model) and the System Name in sync:

```
set rom dsrdb[0] = <SMM> system_name = "<System Name>"
```

Example 6.35.

```
set session hw_model = AlphaServer_ES40
```

```
set ace cpu_architecture = EV67
```

```
set rom dsrdb[0] = 1820 system_name = "AlphaServer ES40 6/667"
```

See Appendix section for proper combination of "cpu_architecture", "hw_model" and "dsrdb[0]" parameters according to the chosen "system_name" value.

6.11. Virtual HP Alpha interval timer

The CHARON-AXP virtualization layer provides interval timer interrupts to virtual Alpha CPU(s) at frequency 100Hz (100 interrupts a second). This is default behavior which may be changed through "clock_period" configuration parameter of virtual ISA or EISA bus, depending on emulated hardware model of virtual HP Alpha system. Value of the parameter is interval timer period in microseconds. By default it is set to 10000. By changing it to 1000 frequency of virtual interval timer interrupts may be increased to 1000Hz (1000 interrupts a second).

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

Example 6.36. Example for AlphaServer 400, DS, ES, GS

```
set ISA clock_period=1000
```

Example 6.37. Example for AlphaServer 800, 1000, 1000A, 1200, 2000, 2100, 4000, 4100

```
set EISA clock_period=1000
```

Note

Higher interval timer frequency creates higher load for virtual Alpha CPU which may cause degradation of overall virtual system performance.

6.12. Data storage in the virtualization layer

6.12.1. Types of data storage

6.12.1.1. Physical disks and disk images

The following options are supported for the disk storage in CHARON environment:

1. **Disk images**, which are essentially binary files in the host file system. They could be located on a local or remote storage. They are easy to maintain and deliver good performance. Backup could be performed with standard operating system tools, making lengthy OpenVMS backups unnecessary. By copying an HP Alpha system or user disk back in place, the disk is fully restored.

The disk images can easily be compressed and sent to a remote site, facilitating remote maintenance and upgrade of CHARON systems.

It is NOT recommended to define disk images in network shared directories. A disconnect of the network storage will permanently disable access from CHARON to the remote disk image.

2. **Physical SCSI disks** connected to a SCSI host adapter or iSCSI Initiator in the host system. (These SCSI disks must not be allocated by Windows operating system; otherwise the drive will not be available for use in CHARON. With the utility "**Disk Management**" (**Control Panel -> Administrative Tools -> Computer Management -> Disk Management**) you can verify that Windows has not allocated the disks. If it has, use the disk management tool to delete the partition (i.e. destroy the file system) and thus release the disk).

Currently only Windows SCSI devices with a driver that uses SCSIport method could be used with the direct SCSI access. Those include most non disk devices on Windows 7, Windows Server 2008 R2 and Windows Server 2012. If SCSIport is not supported (and Starport method is used instead), direct SCSI access couldn't be used.

Using a host SCSI and iSCSI connections allows the use remote storage systems with high reliability RAID configuration.

3. **SAN attached storage volumes**. These volumes must not contain a file system known to the Windows operating system; otherwise the drive will not be available for use in CHARON. With the utility "**Disk Management**" (**Control Panel -> Administrative Tools -> Computer Management -> Disk Management**) you can verify that Windows has not allocated the disks. If it has, use the disk management tool to delete the partition (i.e. destroy the file system) and thus release the disk.
4. **CD and DVD devices** on the host server can be used by the virtualization layer by specifying the usual Windows device name in the configuration script. For example: "\\.\CdRom0" (the first CDROM or DVD drive on the host).

Note

Disk images and physical SCSI disks offer similar I/O throughput. Disk images can be generated with the **MkDisk** utility.

6.12.1.2. Physical tapes and tape images

Tape handling is implemented in CHARON in the following ways:

A SCSI tape drive can be connected to a SCSI controller in the Windows host system. If the Windows tape driver is installed; the device is referenced in the configuration file with its usual Windows device name or file name. For instance "\\.\Tape0" is the first tape drive connected to the host system, and "D:\vtape\MKC500.vtape" represents a virtual tape connected to a container file. In the case of the tape drives not supported by Windows it is possible to use the raw SCSI access in the form: "\\.\SCSI1:0:5:0". Tape operation speed is essentially limited by the capabilities of the physical tape drive and the throughput of the SCSI connection.

6.12.2. Virtual Acer Labs 1543C IDE/ATAPI controller

The "ide" is an instance name for an integrated virtual Acer Labs 1543C IDE/ATAPI controller. Thus no "load" command is required to use it.

ide parameter	Type	Description
container	Text string	Specifies the name of ATAPI or SATA CD/DVD-ROM drive attached to the host system. The supported values are of the form "\\.\CdRomN", where N is 0, 1, 2... In most cases 0 is the only meaningful value for N, because usually the host system has only one CD/DVD-ROM drive. By default it is left unspecified.

Example 6.38.

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

Note

When running HP OpenVMS/Alpha Operating System on top of CHARON-AXP virtualization layer the specified CD/DVD-ROM drive is available as **DQA0:** device.

Note

Please note that the virtual Acer Labs 1543C IDE/ATAPI can me mapped **ONLY TO PHYSICAL CD-ROM DRIVES**. In case if a CD-ROM container or ISO file should be used it is required to utilize KZPBA controller for that since it offers full support of both physical and virtual mappings to system resources.

6.12.3. Virtual KZPBA PCI SCSI adapter

The KZPBA is a PCI SCSI adapter (DEC-KZPBA, based on the QLogic ISP1040 Fast Wide SCSI adapter chip) for the HP Alpha. In CHARON it supports up to 120 disks and tapes.

The I/O behavior of the virtual KZPBA is as follows:

- Up to 120 connected units (disks or tapes) operate in parallel.
- For systems with more than 16 heavily used units configure several virtual KZPBA PCI SCSI adapters and distribute the heavily loaded units evenly between the adapters.

6.12.3.1. Attaching virtual KZPBA PCI SCSI Adapter to virtual system

To create an instance of virtual KZPBA PCI SCSI Adapter use "load" command in configuration file as follows:

```
load kzbpa <instance-name>
```

Note that **<instance-name>** is not visible outside configuration file. Operating systems running on top of virtual system use different naming policy and name assigned to virtual KZPBA PCI SCSI Adapter by those operating systems has nothing to do with **<instance-name>** assigned in configuration files.

Example 6.39.

load kzpba SCSI_A

In this example, **SCSI_A** is instance name of virtual KZPBA PCI SCSI Adapter. But HP OpenVMS operating system uses names PKA,PKB,PKC,... to identify instances of virtual KZPBA PCI SCSI Adapters

6.12.3.2. Configuring virtual KZPBA PCI SCSI Adapter

Virtual KZPBA PCI SCSI Adapter offers several configuration parameters controlling its behavior in virtual hardware and its appearance to software running on it (e.g. HP OpenVMS Alpha and HP Tru64 UNIX operating systems).

6.12.3.2.1. KZPBA general parameters

kzpba parameters	Type	Description
media_type	Text string	<p>When specified, the media_type configuration parameter instructs the CHARON software to use the supplied value as PRODUCT field in SCSI INQUIRY data returned to software running on virtual HP Alpha system in response to SCSI INQUIRY command.</p> <p>If the media_type configuration parameter is not specified, the CHARON software attempts to guess SCSI INQUIRY data based on virtual SCSI device type and underlying container (which is specified in the corresponding container configuration parameter).</p> <p>Example 6.40.</p> <pre>set SCSI_B media_type[0]="HSZ70" set SCSI_B media_type[600]="RRD43"</pre>
removable	Boolean	<p>When set to TRUE, the removable configuration parameter instructs the CHARON software to report the corresponding virtual SCSI device as removable.</p> <p>By default the removable configuration parameter is set to <i>FALSE</i>.</p> <p>Note that virtual SCSI tape and cdrom devices are always reported as removable regardless of the removable configuration parameter.</p> <p>Example 6.41.</p> <pre>set SCSI_A removable[400]=true</pre>
use_io_file_buffering	Boolean	<p>When set to <i>TRUE</i>, the use_io_file_buffering configuration parameter instructs the CHARON software to</p>



kzpba parameters	Type	Description
		<p>enable host operating system I/O cache when reading/writing the corresponding container (specified by the corresponding container configuration parameter).</p> <p>When enabled, the host operating system I/O cache may significantly improve I/O performance of the virtual system. At the same time maintaining I/O cache requires additional host resources (CPU and memory) which may negatively affect overall performance of the virtual system.</p> <p>By default the use_io_file_buffering configuration parameter is set to <i>FALSE</i>.</p> <p>Example 6.42.</p> <pre>set SCSI_A use_io_file_buffering[0]=true</pre>
geometry	Text string	<p>The geometry parameter tells the emulator about a specific geometry of the connected media. The parameters above can be omitted.</p> <p>Syntax:</p> <pre>geometry[unit-number]= "<n_of_sectors>/<n_of_tracks>/<n_of_cylinders>"</pre> <p>Example 6.43.</p> <pre>set SCSI_A geometry[0] = "255/255"</pre>
bus	Text string	<p>When specified, the bus configuration parameter tells the CHARON software the virtual PCI bus to which Virtual HP Alpha system shall connect the virtual KZPBA PCI SCSI Adapter.</p> <p>By default the bus configuration parameter is not specified.</p> <p>If the bus configuration parameter is not specified, the CHARON software connects the virtual KZPBA PCI SCSI Adapter to the first available virtual PCI bus.</p> <p>Example 6.44.</p> <pre>load KZPBA SCSI_A bus=pci_1 device=1 function=0</pre>
device	Numeric	<p>When specified, the device configuration parameter specifies position of the virtual KZPBA PCI SCSI Adapter on virtual PCI bus.</p> <p>By default the device configuration parameter is not specified.</p> <p>If the device configuration parameter is not specified, the CHARON software connects the virtual KZPBA PCI</p>

kzpbpa parameters	Type	Description
		<p>SCSI Adapter at the first available position of the virtual PCI bus.</p> <p>Example 6.45.</p> <p>load KZPBA SCSI_A bus=pci_1 device=1 function=0</p>
function	Numeric	<p>When specified, the function configuration parameter specifies position of the virtual KZPBA PCI SCSI Adapter on virtual PCI bus.</p> <p>By default the function configuration parameter is not specified.</p> <p>If the function configuration parameter is not specified, the CHARON software connects the virtual KZPBA PCI SCSI Adapter at the first available position of the virtual PCI bus.</p> <p>Example 6.46.</p> <p>load KZPBA SCSI_A bus=pci_1 device=1 function=0</p>
irq_bus	Text string	<p>When specified, the irq_bus configuration parameter specifies virtual bus routing interrupt requests from virtual KZPBA PCI SCSI Adapter to virtual Alpha CPUs in Virtual HP Alpha system.</p> <p>By default the irq_bus configuration parameter is not specified.</p> <p>The irq_bus configuration parameter must be set to "ISA" for virtual KZPBA SCSI Adapter in virtual AlphaServer 400. For virtual HP Alpha systems other than AlphaServer 400 the irq_bus configuration parameter must be left as is (i.e. not specified).</p> <p>Example 6.47.</p> <p>load KZPBA SCSI_B irq_bus=isa</p>
irq	Numeric	<p>When specified, the irq configuration parameter assigns interrupt request to the virtual KZPBA PCI SCSI Adapter in Virtual HP Alpha system.</p> <p>By default the irq configuration parameter is not specified.</p> <p>If the irq configuration parameter is not specified, the CHARON software uses the correct values depending on the selected PCI position of virtual KZPBA PCI SCSI Adapter in the virtual system.</p> <p>Example 6.48.</p> <p>load KZPBA SCSI_A bus=pci_1 device=1 function=0 irq=24</p>

kzpba parameters	Type	Description
scsi_id	Numeric	<p>The scsi_id configuration parameter specifies self SCSI ID (Initiator SCSI ID) of the virtual KZPBA PCI SCSI Adapter. The same SCSI ID is also used by virtual KZPBA PCI SCSI Adapter when it is configured as virtual SCSI target in virtual SCSI cluster configuration.</p> <p>By default the scsi_id configuration parameter is set to 7.</p> <p>Example 6.49.</p> <p>set SCSI_B scsi_id=6</p>
port	Text string	<p>When specified, the port configuration parameter specifies local end-point (TCP/IP port on local host) of virtual SCSI connection between the virtual KZPBA PCI SCSI Adapter and a virtual KZPBA PCI SCSI Adapter on remote host in virtual SCSI cluster configuration.</p> <p>By default the port configuration option is not specified.</p> <p>Syntax:</p> <p>host[connection-number]="host-name{:tcpip-port-no}"</p> <p>where</p> <p><i>connection_number = remote_scsi_id * 100 + lun_id</i></p> <p>Example 6.50.</p> <p>set SCSI_B port[600]=17060 host[600]="local-host:16070"</p>
host	Text string	<p>When specified, the host configuration parameter specifies remote end-point (remote host name and, optionally, TCP/IP port on remote host) of virtual SCSI connection between the virtual KZPBA PCI SCSI Adapter and a virtual KZPBA PCI SCSI Adapter on remote host in virtual SCSI cluster configuration.</p> <p>By default the host configuration option is not specified.</p> <p>Syntax:</p> <p>host[connection-number]="host-name{:tcpip-port-no}"</p> <p>where</p> <p><i>connection_number = remote_scsi_id * 100 + lun_id</i></p> <p>Example 6.51.</p> <p>set SCSI_B port[600]=17060 host[600]="local-host:16070"</p>

Example 6.52.

```
load KZPBA SCSI_B
set SCSI_B container[0]="dkb0.vdisk"
set SCSI_B media_type[0]="HSZ70"
set SCSI_B media_type[600]="RRD43"
```

6.12.3.2.2. KZPBA mapping to system resources

kzpba parameters	Type	Description
container	Text string	<p>When specified this configuration parameter instructs the CHARON software to create virtual SCSI device and connect to the virtual system through the virtual KZPBA SCSI Adapter. Type of the virtual SCSI device depends on value of the configuration parameter</p> <p><i>Syntax:</i></p> <p>container[unit-number]="{file-path}\file-name.vdisk"</p> <p>container[unit-number]="{file-path}\file-name.vtape"</p> <p>container[unit-number]="{file-path}\file-name.iso"</p> <p>container[unit-number]="\\.\PhysicalDriveN"</p> <p>container[unit-number]="\\.\PhysicalDrive(DevID = 6008-05F3-0005-2950-BF8E-0B86-A0C7-0001)"</p> <p>container[unit-number]="\\.\PhysicalDrive(iScsiTarget = iqn.2008-04:iscsi.charon-target-test1, LUN = 1)"</p> <p>container[unit-number]="\\.\TapeN"</p> <p>container[unit-number]="\\.\CdRomN"</p> <p>container[unit-number]="\\.\ScsiN:B:S:L"</p> <p>where</p> <ul style="list-style-type: none"> unit-number = scsi_id * 100 + lun_id is number of virtual storage element attached to the virtual KZPBA PCI SCSI Adapter. In this formula scsi_id is from 0 through 15 and lun_id is from 0 through 7. This gives the following valid unit numbers: 0, 1, ..., 7, 100, 101, ..., 107, 200, ..., 1507. Note that storage unit number assigned by HP OpenVMS Alpha operating system running on virtual HP Alpha system (appears on device name) is the same as unit number given by the above formula. N is logical number assigned by host operating system (Microsoft Windows) to logical or host's physical storage resource such as physical disk drive (\\.\PhysicalDrive notation), physical tape drive

kzpbpa parameters	Type	Description
		<p>(\\.\Tape notation), physical CD/DVD drive (\\.\CdRom notation) or physical SCSI HBA (\\.\Scsi notation).</p> <ul style="list-style-type: none"> • DevID addresses the target physical disk by its WWID (a hexadecimal 128-bit identifier assigned to the disk drive by its manufacturer/originator) • iScsiTarget addresses the disk by its iSCSI target name • LUN specifies LUN on connected iSCSI disk • B is internal SCSI bus number (usually 0) on host's physical SCSI HBA (\\.\Scsi notation). • S is SCSI ID of physical SCSI target device attached to host's physical SCSI HBA (\\.\Scsi notation) • L is LOGICAL UNIT NUMBER inside physical SCSI target device attached to host's physical SCSI HBA (\\.\Scsi notation) <p>Example 6.53.</p> <pre>set SCSI_A container[0]="d:\disks\dka0.vdisk" set SCSI_A container[600]="d:\disks\mka600.vtape" set SCSI_B container[600]="\\.\CdRom0" set SCSI_B container[500]="\\.\Tape0" set SCSI_B container[501]="\\.\Scsi5:0:3:1"</pre> <p>Types of container parameters:</p> <ul style="list-style-type: none"> • The <i>.vdisk</i> file represents container of virtual disk. When path to <i>.vdisk</i> file is assigned to container configuration parameter the CHARON-AXP software creates virtual SCSI disk device. The CHARON-AXP software supports also <i>.dsk</i> files for backward compatibility, although use of <i>.dsk</i> extension is not recommended. • The <i>.vtape</i> file represents container of virtual tape. When path to <i>.vtape</i> file is assigned to container configuration parameter the CHARON-AXP software creates virtual SCSI tape device. The CHARON-AXP software supports also <i>.mtd</i> files for backward compatibility, although use of <i>.mtd</i> extension is not recommended. • The <i>.iso</i> file represents container of virtual cdrom. When path to <i>.iso</i> file is assigned to container configuration parameter the CHARON-AXP software creates virtual SCSI cdrom device. It is possible to switch from one <i>.iso</i> file to other one (having the same name) w/o stopping CHARON-AXP. But note that

kzpba parameters	Type	Description
		<p>the CD-ROM device must be dismantled first on the CHARON operating system level.</p> <ul style="list-style-type: none"> • The \\.\PhysicalDrive object represents logical or physical disk attached to the host or iSCSI target. When certain \\.\PhysicalDrive is assigned to container configuration parameter the CHARON-AXP software creates virtual SCSI disk device. • The \\.\Tape object represents physical tape drive attached to the host. When certain \\.\Tape is assigned to container configuration parameter the CHARON-AXP software creates virtual SCSI tape device. • The \\.\CdRom object represents logical or physical optical drive attached to the host. When certain \\.\CdRom is assigned to container configuration parameter the CHARON-AXP software creates virtual SCSI cdrom device. • The \\.\Scsi object represents direct (SCSI PASS THROUGH) connection to (presumably) unusual physical SCSI device attached to the host through physical SCSI HBA and for which host operating system (Microsoft Windows) does not have specific device drivers. When certain \\.\Scsi object name is assigned to container configuration parameter the CHARON-AXP software creates virtual SCSI device directly (i.e. using SCSI PASS THROUGH interface) connected to the physical SCSI device.

Example 6.54.

```
load KZPBA SCSI_B
set SCSI_B container[0]="dkb0.vdisk"
set SCSI_B container[600]="\\.\CdRom0"
```

6.12.4. Virtual DEC-KGPSA-CA (EMULEX LP8000) PCI Fibre Channel adapter

CHARON supports emulation of DEC-KGPSA-CA (EMULEX LP8000) PCI FC adapter by loading instances of KGPSA.

Every instance of KGPSA works in one of two modes:

- Fabric virtualization mode (creating virtual fabric in combination with virtual FC-3 Storage Controller). This is default mode.
- CHARON PCI Pass Through mode (using CHARON PCI Pass Through driver)



6.12.4.1. Attaching virtual KGPSA PCI Fibre Channel Adapter to virtual system

To create an instance of virtual KGPSA PCI FC Adapter use “**load**” command in configuration file as follows:

```
load kgpsa <instance-name>
```

Note that **<instance-name>** is not visible outside configuration file. Operating systems running on top of virtual system use different naming policy and name assigned to virtual KGPSA PCI FC Adapter by those operating systems has nothing to do with **<instance-name>** assigned in configuration files.

Example 6.55.

```
load kgpsa FC_A
```

In this example, **FC_A** is instance name of virtual KGPSA PCI FC Adapter. But HP OpenVMS operating system uses names FGA,FGB,FGC,... to identify instances of virtual KGPSA PCI FC Adapters

6.12.4.2. Configuring virtual KGPSA PCI Fibre Channel Adapter in Fabric virtualization mode

Virtual KGPSA PCI FC Adapter offers several configuration parameters controlling its behavior in virtual hardware and its appearance to software running on it (e.g. HP OpenVMS Alpha and HP Tru64 UNIX operating systems).

6.12.4.2.1. KGPSA general parameters

kgpsa parameters	Type	Description
media_type	Text string	<p>When specified, the media_type configuration parameter instructs the CHARON software to use the supplied value as PRODUCT field in FC INQUIRY data returned to software running on virtual HP Alpha system in response to FC INQUIRY command.</p> <p>If the media_type configuration parameter is not specified, the CHARON software attempts to guess FC INQUIRY data based on virtual FC device type and underlying container (which is specified in the corresponding container configuration parameter).</p> <p><i>Syntax:</i></p> <pre>media_type[unit-number]="<vendor>,<product>,<revision>"</pre> <p>Example 6.56.</p> <pre>set FC_B media_type[0]="DEC,HSG80,V89F" set FC_B media_type[1]="HP,MSA1000,V100"</pre>
wwid	Text string	This parameter sets WWID for emulated KGPSA adapter unit.

kgpsa parameters	Type	Description
		<p><i>Syntax:</i></p> <p>wwid[unit-number]="XXXX-XXXX-XXXX-XXXX-XXXX-XXXX-XXXX-XXXX"</p> <p>Example 6.57.</p> <p>set FC_A wwid[2]="6008-05F3-0005-2950-BF8E-0B86-A0C7-0001"</p>
removable	Boolean	<p>When set to TRUE, the removable configuration parameter instructs the CHARON software to report the corresponding virtual FC device as removable.</p> <p>By default the removable configuration parameter is set to <i>FALSE</i>.</p> <p>Note that virtual FC tape and cdrom devices are always reported as removable regardless of the removable configuration parameter.</p> <p>Example 6.58.</p> <p>set FC_A removable[400]=true</p>
use_io_file_buffering	Boolean	<p>When set to <i>TRUE</i>, the use_io_file_buffering configuration parameter instructs the CHARON software to enable host operating system I/O cache when reading/writing the corresponding container (specified by the corresponding container configuration parameter).</p> <p>When enabled, the host operating system I/O cache may significantly improve I/O performance of the virtual system. At the same time maintaining I/O cache requires additional host resources (CPU and memory) which may negatively affect overall performance of the virtual system.</p> <p>By default the use_io_file_buffering configuration parameter is set to <i>FALSE</i>.</p> <p>Example 6.59.</p> <p>set FC_A use_io_file_buffering[0]=true</p>
geometry	Text string	<p>The geometry parameter tells the emulator about a specific geometry of the connected media. The parameters above can be omitted.</p> <p><i>Syntax:</i></p> <p>geometry[unit-number]= "<n_of_sectors>/<n_of_tracks>/<n_of_cylinders>"</p> <p>Example 6.60.</p> <p>set FC_A geometry[0] = "255/255"</p>

kgpsa parameters	Type	Description
host_bus_location	Text string	<p>The parameter triggers CHARON PCI Pass Through mode on and connects the instance of emulated DEC-KGPSA-CA PCI FC adapter to a physical EMULEX LightPulse PCI/PCI-X/PCIe FC adapter plugged into host's PCI/PCI-X/PCIe slot.</p> <p>Example 6.61. Example for Windows</p> <p>load KGPSA FC_A host_bus_location="PCI bus 3,device 1,function 0"</p> <p>Example 6.62. Example for Linux</p> <p>load KGPSA FC_A host_bus_location="/dev/kgpsa0"</p>
bus	Text string	<p>When specified, the bus configuration parameter tells the CHARON software the virtual PCI bus to which Virtual HP Alpha system shall connect the virtual KGPSA PCI FC Adapter.</p> <p>By default the bus configuration parameter is not specified.</p> <p>If the bus configuration parameter is not specified, the CHARON software connects the virtual KGPSA PCI FC Adapter to the first available virtual PCI bus.</p> <p>Example 6.63.</p> <p>load KGPSA FC_A bus=pci_1 device=1 function=0</p>
device	Numeric	<p>When specified, the device configuration parameter specifies position of the virtual KGPSA PCI FC Adapter on virtual PCI bus.</p> <p>By default the device configuration parameter is not specified.</p> <p>If the device configuration parameter is not specified, the CHARON software connects the virtual KGPSA PCI FC Adapter at the first available position of the virtual PCI bus.</p> <p>Example 6.64.</p> <p>load KGPSA FC_A bus=pci_1 device=1 function=0</p>
function	Numeric	<p>When specified, the function configuration parameter specifies position of the virtual KGPSA PCI FC Adapter on virtual PCI bus.</p> <p>By default the function configuration parameter is not specified.</p> <p>If the function configuration parameter is not specified, the CHARON software connects the virtual KGPSA PCI</p>

kgpsa parameters	Type	Description
		<p>FC Adapter at the first available position of the virtual PCI bus.</p> <p>Example 6.65.</p> <p>load KGPSA FC_A bus=pci_1 device=1 function=0</p>
irq_bus	Text string	<p>When specified, the irq_bus configuration parameter specifies virtual bus routing interrupt requests from virtual KGPSA PCI FC Adapter to virtual Alpha CPUs in Virtual HP Alpha system.</p> <p>By default the irq_bus configuration parameter is not specified.</p> <p>The irq_bus configuration parameter must be set to "ISA" for virtual KGPSA FC Adapter in virtual AlphaServer 400. For virtual HP Alpha systems other than AlphaServer 400 the irq_bus configuration parameter must be left as is (i.e. not specified).</p> <p>Example 6.66.</p> <p>load KGPSA FC_B irq_bus=isa</p>
irq	Numeric	<p>When specified, the irq configuration parameter assigns interrupt request to the virtual KGPSA PCI FC Adapter in Virtual HP Alpha system.</p> <p>By default the irq configuration parameter is not specified.</p> <p>If the irq configuration parameter is not specified, the CHARON software uses the correct value depending on the selected PCI position of virtual KGPSA PCI FC Adapter in the virtual system.</p> <p>Example 6.67.</p> <p>load KGPSA SCSI_A bus=pci_1 device=1 function=0 irq=24</p>

Example 6.68.

```
load KGPSA FC_A bus=pci_1 device=1 function=0 irq=24
set FC_A media_type[100]="DEC,HSG80,V89F"
set FC_A removable[100]=true
set FC_A use_io_file_buffering[100]=true
```

6.12.4.2.2. KGPSA mapping to system resources

kgpsa parameters	Type	Description
container	Text string	<p>When specified this configuration parameter instructs the CHARON software to create virtual FC device and connect to the virtual system through the virtual KGPSA FC Adapter. Type of the virtual FC device depends on value of the configuration parameter</p> <p><i>Syntax:</i></p> <p>container[<i>unit-number</i>]="{file-path}\file-name.vdisk"</p> <p>container[<i>unit-number</i>]="\\.\PhysicalDriveN"</p> <p>container[<i>unit-number</i>]="\\.\PhysicalDrive(DevID = XXX-XXXX-XXXX-XXXX-XXXX-XXXX-XXXX-XXXX)"</p> <p>container[<i>unit-number</i>]="\\.\PhysicalDrive(iScsiTarget = <iSCSI target>, LUN = <LUN number>)"</p> <p>where</p> <ul style="list-style-type: none"> • unit-number is number of virtual storage element attached to the virtual DEC-KGPSA-CA PCI FC adapter. The unit-number is 0 through 99998. The unit-number 99999 is reserved. • N is logical number assigned by host operating system (Microsoft Windows) to logical or host's physical disk drive. • DevID addresses the target physical disk by its WWID (hexadecimal 128-bit identifier assigned to the disk drive by its manufacturer/originator) • iScsiTarget addresses the disk by its iSCSI target name • LUN specifies LUN on connected iSCSI disk <p>Example 6.69.</p> <pre>set FC_A container[100]="d:\disks\dka0.vdisk" set FC_B container[128]="\\.\PhysicalDrive1" set FC_B container[132]="\\.\PhysicalDrive(DevID = 6008-05F3-0005-2950-BF8E-0B86-A0C7-0001)" set FC_B container[148]="\\.\PhysicalDrive(iScsiTarget = iqn.2008-04:iscsi.charon-target-test1, LUN = 1)"</pre> <p><i>Description:</i></p> <ul style="list-style-type: none"> • The <i>.vdisk</i> file represents container of virtual disk. When path to <i>.vdisk</i> file is assigned to container configuration parameter the CHARON-AXP software

kgpsa parameters	Type	Description
		<p>creates virtual SCSI disk device. The CHARON-AXP software supports also .dsk files for backward compatibility, although use of .dsk extension is not recommended.</p> <ul style="list-style-type: none"> The \\.\PhysicalDrive object represents logical or physical disk attached to the host or iSCSI target. When certain \\.\PhysicalDrive is assigned to container configuration parameter the CHARON-AXP software creates virtual SCSI disk device. <p>If the container configuration parameter is not specified, the CHARON-AXP software does not create virtual SCSI device for the corresponding unit number.</p>

Example 6.70.

```
load KGPSA FC_A
```

```
set FC_A container[100]="d:\disks\dka0.vdisk"
```

6.12.4.3. Configuring virtual KGPSA PCI Fibre Channel Adapter for CHARON PCI Pass Through

The CHARON PCI Pass Through mode allows connection between virtual DEC-KGPSA-CA PCI FC adapter and physical EMULEX LightPulse PCI/PCI-X/PCIe FC adapter plugged into host's PCI/PCI-X/PCIe slot.

Note

CHARON PCI Pass Through driver for EMULEX LightPulse PCI/PCI-X/PCIe FC adapter must be installed, up, and running.

The connection between virtual DEC-KGPSA-CA PCI FC adapter and physical EMULEX LightPulse PCI/PCI-X/PCIe FC adapter is setup through "**host_bus_location**" parameter as follows.

Syntax:

```
host_bus_location="PCI bus X, device Y, function Z"
```

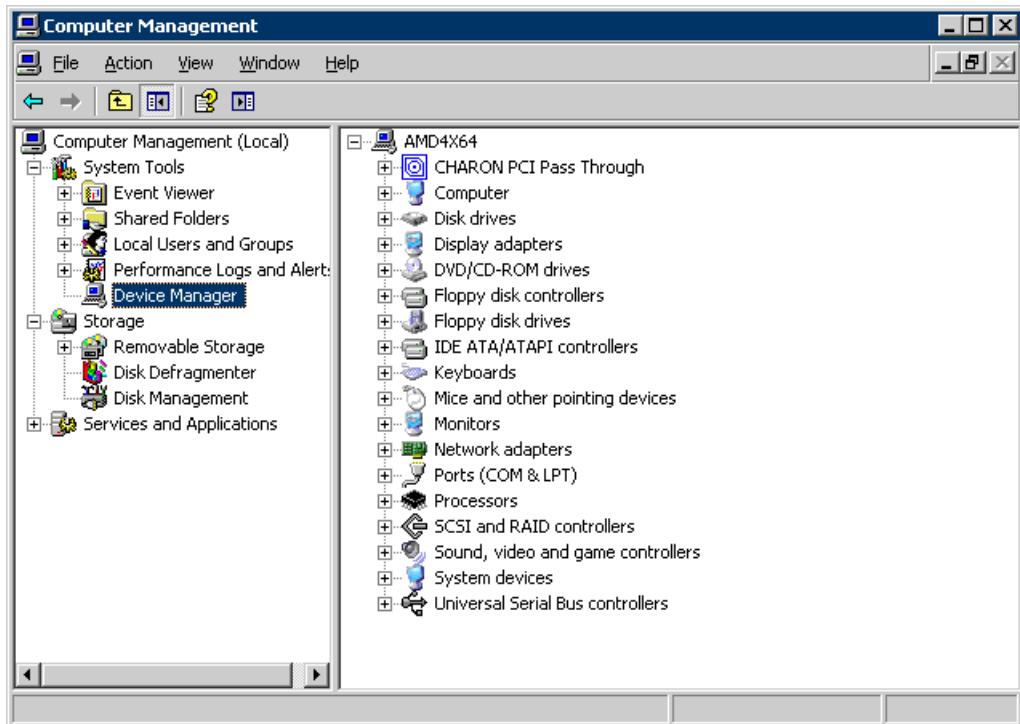
where **X**, **Y**, and **Z** describe location of physical EMULEX LightPulse PCI/PCI-X/PCIe FC adapter in the host computer.

Example 6.71.

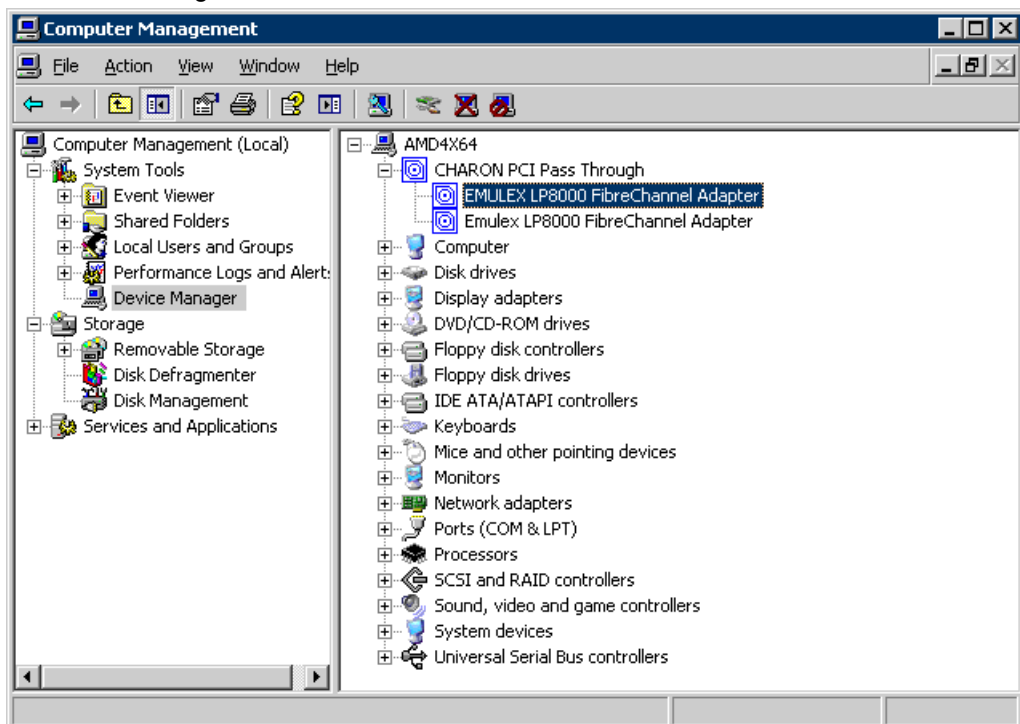
```
load KGPSA FC_A host_bus_location="PCI bus 3,device 1,function 0"
```

6.12.4.3.1. Defining parameters for the "host_bus_location"

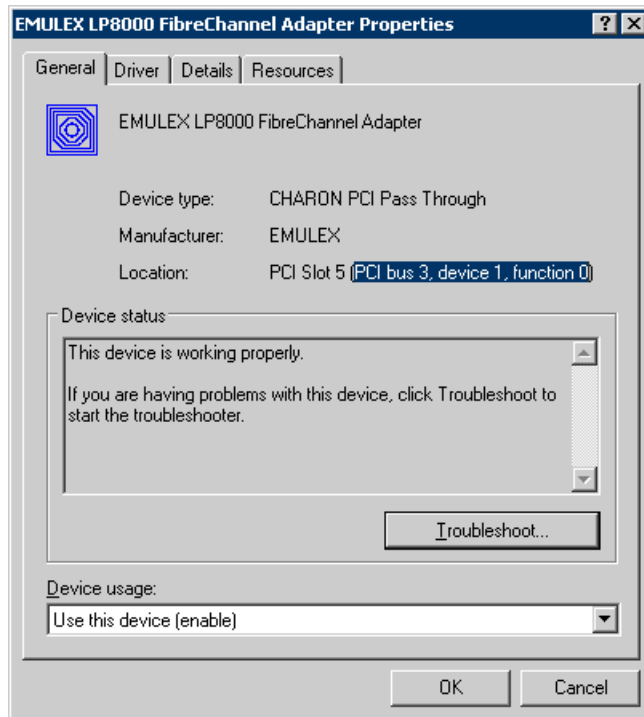
Open "**Computer Management**" application and select "Device Manager":



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



And open its property sheet by double-click on selected adapter:



The “*Location:*” on the above picture gives **X**, **Y**, and **Z** for the **host_bus_location** parameter.

Note

Non-US-EN installations of Windows may present “*Location:*” string in local language, but **host_bus_location** parameter requires English notation, that is words “*PCI*”, “*bus*”, “*device*”, and “*function*” must be specified in English.

6.12.4.3.2. Supported physical EMULEX LightPulse PCI/PCI-X/PCIe FC adapters

The following is the list of EMULEX LightPulse PCI/PCI-X/PCIe FC adapters supported by CHARON PCI Pass Through driver and suitable for emulation of DEC-KGPSA-CA PCI FC adapter in CHARON PCI Pass Through mode:

- LP8000
- LP9000
- LP9002
- LP9802
- LP10000
- LP10000DC
- LP10000-S
- LPX1000
- LP11002
- LPe11002 (FC2242SR, A8003A)
- LPe1105

Not supported:

- LPe1150 (FC2142SR, A8002A)

Not tested:

- LPe11000

6.13. Virtual PCI Ethernet controllers

CHARON-AXP implements the following virtual PCI Network controllers:

- DE435
- DE450
- DE500AA
- DE500BA

Each of them is a PCI Ethernet adapter (based on the DEC21040 (DE435, DE450, DE500AA and DE500BA) PCI Ethernet adapter chips) for the HP Alpha. CHARON-AXP maps the virtual adapter to a dedicated Ethernet adapter in the Windows host system.

The Ethernet adapter in the Windows host system must support dynamic changes of its MAC address (i.e. no reboot of the host system is required to change the MAC address), which is the case with nearly all modern Ethernet adapters.

The virtual Ethernet controller requires installation of the NDIS5/NDIS6 CHARON Packet driver, which provides a direct link between the virtual network interface and dedicated Ethernet adapter in the Windows host system. The driver installation is described in the Chapter 4. The proper sequences is to first load an instance of virtual Ethernet controller, then load an instance of virtual network interface connected to the NDIS5/NDIS6 packet driver, and then finally link the two virtual entities.

Example 6.72. DEC21040 chip adapters (DE435, DE450, DE500AA and DE500BA)

```
load DE500BA/dec21x4x IFC
```

```
load packet_port/chnetwrk IFC0 interface="connection:CHARON"
```

```
set IFC interface=IFC0
```

6.13.1. Virtual DE435, DE450, DE500AA and DE500BA network adapters

6.13.1.1. Attaching virtual DE435, DE450, DE500AA and DE500BA to virtual system

To create instances of virtual DExx series network adapters use “**load**” command in configuration file as follows:

```
load DE435/dec21x4x <instance-name>
```

```
load DE450/dec21x4x <instance-name>
```

```
load DE500AA/dec21x4x <instance-name>
```

load DE500BA/dec21x4x <instance-name>

Note that <instance-name> is not visible outside configuration file. Operating systems running on top of virtual system use different naming policy and name assigned to virtual DExx series network adapters by those operating systems has nothing to do with <instance-name> assigned in configuration files.

Example 6.73. DEC21040 chip adapters (DE435, DE450, DE500AA and DE500BA)

load DE500AA/dec21x4x NI_A

In this example, NI_A is instance name of virtual DE500AA series network adapter. But HP OpenVMS operating system uses names EWA,EWB,EWC,... to identify instances of virtual DE435, DE450, DE500AA or DE500BA network adapters

6.13.1.2. Configuring virtual DE435, DE450, DE500AA and DE500BA network adapters

Virtual DExx series network adapters offer several configuration parameters controlling its behavior in virtual hardware and its appearance to software running on it (e.g. HP OpenVMS Alpha and HP Tru64 UNIX operating systems).

DExxx paramet-ers	Type	Description
interface	Text string	This parameter connects the logical name representing a DExxx instance to the logical name of a host network port ("IFC0" in the following example), after the host network port has been loaded. Example 6.74. set NI_A interface=IFC0
station_address	Text string	station_address provides an 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. <i>Format:</i> XX-XX-XX-XX-XX-XX or XX:XX:XX:XX:XX:XX Example 6.75. set NI_A station_address="E2:F4:00:A6:07:D2" See Appendix for more detailed description.
rx_fifo_size	Numeric	This parameter 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.

DExxx parameters	Type	Description
		<p>Typically, you do not need to specify the rx_fifo_size parameter. It is available mostly for extended tuning.</p> <p>Example 6.76.</p> <p>set NI_A rx_fifo_size=256</p>
bus	Text string	<p>When specified, the bus configuration parameter tells the CHARON software the virtual PCI bus to which the virtual system shall connect the virtual DExx series network adapters.</p> <p>By default the bus configuration parameter is not specified.</p> <p>If the bus configuration parameter is not specified, the CHARON software connects the virtual DExx series network adapters to the first available virtual PCI bus.</p> <p>Example 6.77. DEC21040 chip adapters (DE435, DE450, DE500AA and DE500BA)</p> <p>load DE500AA/dec21x4x NI_A bus=pci_1 device=1 function=0</p>
device	Numeric	<p>When specified, the device configuration parameter specifies position of the virtual DExx series network adapters on virtual PCI bus.</p> <p>By default the device configuration parameter is not specified.</p> <p>If the device configuration parameter is not specified, the CHARON software connects the virtual DExx series network adapters at the first available position of the virtual PCI bus.</p> <p>Example 6.78. DEC21040 chip adapters (DE435, DE450, DE500AA and DE500BA)</p> <p>load DE500AA/dec21x4x NI_A bus=pci_1 device=1 function=0</p>
function	Numeric	<p>When specified, the function configuration parameter specifies position of the virtual DExx series network adapters on virtual PCI bus.</p> <p>By default the function configuration parameter is not specified.</p> <p>If the function configuration parameter is not specified, the CHARON software connects the virtual DExx series network adapters at the first available position of the virtual PCI bus.</p>

DExxx parameters	Type	Description
		<p>Example 6.79. DEC21040 chip adapters (DE435, DE450, DE500AA and DE500BA)</p> <p>load DE500AA/dec21x4x NI_A bus=pci_1 device=1 function=0</p>
irq_bus	Text string	<p>When specified, the irq_bus configuration parameter specifies virtual bus routing interrupt requests from virtual DExx series network adapters to virtual Alpha CPUs in Virtual HP Alpha system.</p> <p>By default the irq_bus configuration parameter is not specified.</p> <p>The irq_bus configuration parameter must be set to "ISA" for virtual DExx series network adapters in virtual AlphaServer 400. For virtual HP Alpha systems other than AlphaServer 400 the irq_bus configuration parameter must be left as is (i.e. not specified).</p> <p>Example 6.80. DEC21040 chip adapters (DE435, DE450, DE500AA and DE500BA)</p> <p>load DE500AA/dec21x4x NI_A irq_bus=isa</p>
irq	Numeric	<p>When specified, the irq configuration parameter assigns interrupt request to the virtual DExx series network adapters in Virtual HP Alpha system.</p> <p>By default the irq configuration parameter is not specified.</p> <p>If the irq configuration parameter is not specified, the CHARON software uses the correct value depending on the selected PCI position of virtual DExx series network adapters in the virtual system.</p> <p>Example 6.81. DEC21040 chip adapters (DE435, DE450, DE500AA and DE500BA)</p> <p>load DE500AA/dec21x4x NI_A bus=pci_1 device=1 function=0 irq=24</p>

Example 6.82.

```
load DE435/dec21x4x EWA interface=EWA0
set EWA station_address="E2:F4:00:A6:07:D2"
set EWA bus=pci_1 device=1 function=0 irq=24
set EWA irq_bus=isa
```

This example assumes that the network packet port is defined as **EWA0**

It is recommended to review the sample configuration files to see the correct structure of the Ethernet configuration commands.

If your OpenVMS/Alpha system disk is configured for automatic TCP/IP startup and you use UCX, not loading an Ethernet adapter in the CHARON-AXP configuration can cause OpenVMS to crash. The problem appears only if UCX is enabled while the networking device is missing. DECnet works correctly.

For the extended set of the adapter configuration parameters, tuning and troubleshooting please refer to the 'Charon networking Guide'.

6.13.2. Virtual DEFPA PCI FDDI adapter

CHARON supports emulation of DEFPA PCI FDDI adapter by loading instances of DEFPA.

6.13.2.1. Attaching virtual DEFPA PCI FDDI Adapter to virtual system

CHARON virtualization layer supports virtual DEFPA PCI FDDI adapter using Pass Through mode. CHARON PCI Pass Through mode enables connection between the virtual DEFPA PCI FDDI adapter and the physical DEFPA PCI FDDI adapter plugged into a hosting server PCI bus

To create an instance of virtual DEFPA PCI FDDI Adapter use “load” command in configuration file as follows:

load defpa <instance-name>

Note that <instance-name> is not visible outside configuration file. Operating systems running on top of virtual system use different naming policy and name assigned to virtual DEFPA PCI FDDI Adapter by those operating systems has nothing to do with <instance-name> assigned in configuration files.

Example 6.83.

load defpa FDDI_A

In this example, **FDDI_A** is instance name of virtual DEFPA PCI FDDI Adapter. But HP OpenVMS operating system uses names FWA,FWB,FWC,... to identify instances of virtual DEFPA PCI FDDI Adapters

6.13.2.2. Configuring virtual DEFPA PCI FDDI Adapter

Virtual DEFPA PCI FDDI Adapter offers several configuration parameters controlling its behavior in virtual hardware and its appearance to software running on it (e.g. HP OpenVMS Alpha and HP Tru64 UNIX operating systems).

defpa parameters	Type	Description
host_bus_location	Text string	Enables usage of CHARON PCI Pass Through and specifies physical location of physical DEFPA PCI FDDI adapter. Example 6.84. load defpa FDDI_A host_bus_location="PCI bus 3,device 1,function 0"

defpa parameters	Type	Description
bus	Text string	<p>When specified, the bus configuration parameter tells the CHARON software the virtual PCI bus to which the virtual system shall connect the virtual DEFPA PCI FDDI Adapter.</p> <p>By default the bus configuration parameter is not specified.</p> <p>If the bus configuration parameter is not specified, the CHARON software connects the virtual DEFPA PCI FDDI Adapter to the first available virtual PCI bus.</p> <p>Example 6.85.</p> <p>load defpa FDDI_A bus=pci_1 device=1 function=0</p>
device	Numeric	<p>When specified, the device configuration parameter specifies position of the virtual DEFPA PCI FDDI Adapter on virtual PCI bus.</p> <p>By default the device configuration parameter is not specified.</p> <p>If the device configuration parameter is not specified, the CHARON software connects the virtual DEFPA PCI FDDI Adapter at the first available position of the virtual PCI bus.</p> <p>Example 6.86.</p> <p>load defpa FDDI_A bus=pci_1 device=1 function=0</p>
function	Numeric	<p>When specified, the function configuration parameter specifies position of the virtual DEFPA PCI FDDI Adapter on virtual PCI bus.</p> <p>By default the function configuration parameter is not specified.</p> <p>If the function configuration parameter is not specified, the CHARON software connects the virtual DEFPA PCI FDDI Adapter at the first available position of the virtual PCI bus.</p> <p>Example 6.87.</p> <p>load defpa FDDI_A bus=pci_1 device=1 function=0</p>
irq_bus	Text string	<p>When specified, the irq_bus configuration parameter specifies virtual bus routing interrupt requests from virtual DEFPA PCI FDDI Adapter to virtual Alpha CPUs in Virtual HP Alpha system.</p> <p>By default the irq_bus configuration parameter is not specified.</p>

defpa parameters	Type	Description
		<p>The <code>irq_bus</code> configuration parameter must be set to <code>ISA</code> for virtual DEFPA PCI FDDI Adapter in virtual AlphaServer 400. For virtual HP Alpha systems other than AlphaServer 400 the <code>irq_bus</code> configuration parameter must be left as is (i.e. not specified).</p> <p>Example 6.88.</p> <pre>load defpa FDDI_A irq_bus=isa</pre>
irq	Numeric	<p>When specified, the <code>irq</code> configuration parameter assigns interrupt request to the virtual DEFPA PCI FDDI Adapter in Virtual HP Alpha system.</p> <p>By default the <code>irq</code> configuration parameter is not specified.</p> <p>If the <code>irq</code> configuration parameter is not specified, the CHARON software uses the correct values depending on the selected PCI position of virtual DEFPA PCI FDDI Adapter in the virtual system.</p> <p>Example 6.89.</p> <pre>load defpa FDDI_A bus=pci_1 device=1 function=0 irq=24</pre>

Example 6.90.

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

Note

CHARON PCI Pass Through driver must be installed and running for physical DEFPA PCI FDDI adapter

Installation of the driver:

1. Open "Computer Management"
2. Select "Device Manager"
3. In the right window select the desired physical FDDI adapter connected to the system, right-click the mouse button, and the corresponding menu will appear.
4. From the menu select "Update driver...". Windows will show "Hardware Upgrade Wizard"
5. Select "No, not this time", click "Next"
6. Select "Install from a list or specific location (Advanced)", click "Next"
7. Select "Don't search. I will choose the driver to install", click "Next"

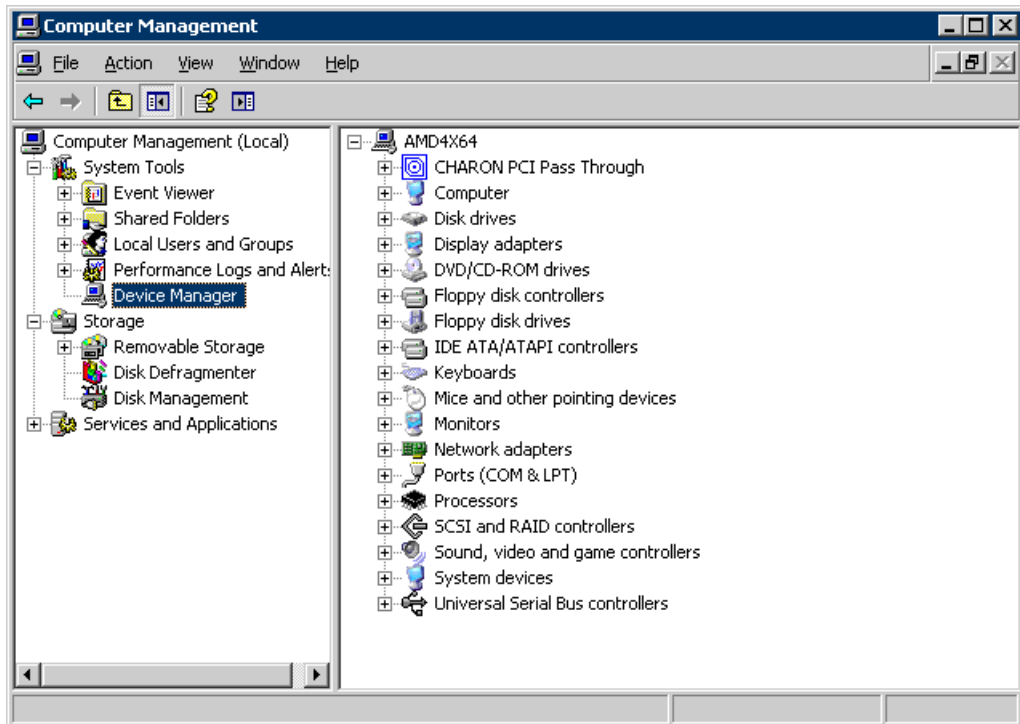


8. Click "Have Disk...". Windows shows dialog "Install From Disk"
9. Instead of "A:\", click "Browse" and select path to the folder in which driver's INF file is located, select "*defpa_ppt_amd64.inf*", and click "Open"
10. The "Hardware Upgrade Wizard" should have "**CHARON DEFPA FDDI adapter**". Select it, and click "Next"
11. There will be one or two more dialogs, but they are usual for device driver installation.

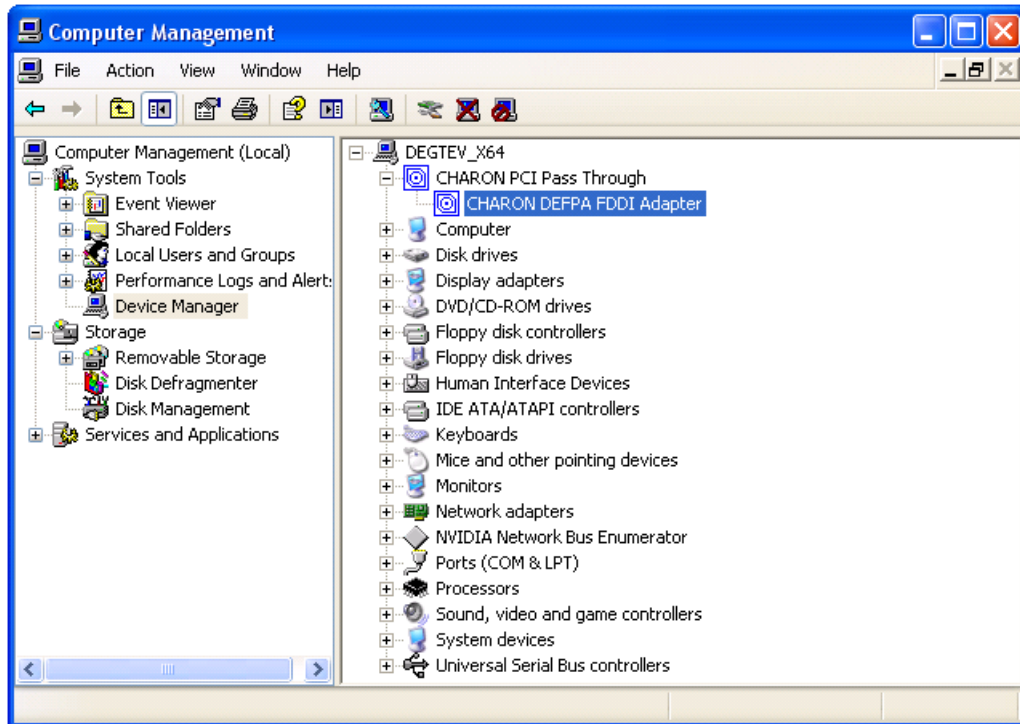
Upon completion, a new device will appear in the device manager with the CHARON logo on it.

6.13.2.3. Defining parameters for the "host_bus_location"

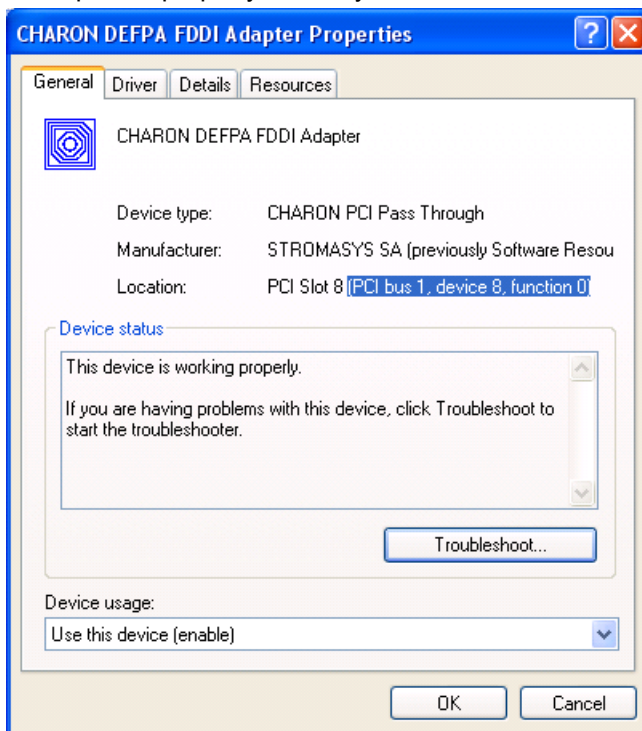
Open "Computer Management" application and select "Device Manager":



On the right panel please select the proper physical DEFPA adapter:



And open its property sheet by double-click on selected adapter:



The "Location:" on the above picture gives X, Y, and Z for the `host_bus_location` parameter.

Note

Non-US-EN installations of Windows may present "Location:" string in local language, but `host_bus_location` parameter requires English notation, that is words "PCI", "bus", "device", and "function" must be specified in English.

6.13.3. NDIS5/NDIS6 CHARON Packet Port

The CHARON specific NDIS5/NDIS6 packet driver establishes the connection between an Ethernet adapter in the Windows host system and Ethernet adapter in the virtual system. For every virtual adapter instance loaded, one dedicated host Ethernet adapter is required. For each of these Ethernet adapters the NDIS5/NDIS6 packet driver must be enabled, and all Windows drivers and services disabled, since CHARON needs exclusive use of these adapters for optimal performance, and to avoid interference from the host network traffic. The NDIS5/NDIS6 driver is installed automatically by the CHARON installation procedure (if this component is not disabled).

6.13.3.1. Attaching NDIS5/NDIS6 CHARON Packet Port to virtual system

To create instances of NDIS5/NDIS6 CHARON Packet Port use “**load**” command in configuration file as follows:

```
load packet_port/chnetwrk <instance-name>
```

Note that <instance-name> is not visible outside configuration file.

Example 6.91.

```
load packet_port/chnetwrk pp_1
```

In this example, **pp_1** is instance name of NDIS5/NDIS6 CHARON Packet Port. This instance name is used for attaching CHARON virtual Ethernet adapters to it.

6.13.3.2. Configuring NDIS5/NDIS6 CHARON Packet Port

NDIS5/NDIS6 CHARON Packet Port offers several configuration parameters controlling its behavior.

6.13.3.2.1. CHARON Packet Port general parameters

packet_port parameter	Type	Description
port_enable_mac_addr_change	Boolean	If <i>true</i> is specified, CHARON sets the appropriate Ethernet address automatically. If false is specified, set the Ethernet address manually. The default value is <i>true</i> . Example 6.92. set pp_1 port_enable_mac_addr_change=false
port_ignore_on_rx	Numeric	port_ignore_on_rx provides the ability to shutdown the port when the sequential errors "on receive" exceeds the specified number. Typically, errors on receive indicate serious (unrecoverable) errors. By default, the value is set to the value of the port_pending_rx_number parameter. Value of '0' means infinite. Example 6.93. set pp_1 port_ignore_on_rx=16

packet_port parameter	Type	Description
port_retry_on_tx	Numeric	port_retry_on_tx controls the number of times the port attempt to transmit the packet until giving up. By default, the value is 3. Increasing this value might introduce problems in carrier losing logic, because not all NIC drivers support carrier status query. Typically, you do not need to increase the value. Example 6.94. set pp_1 port_retry_on_tx=8
port_pending_rx_number	Numeric	port_pending_rx_number 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 6.95. set pp_1 port_pending_rx_number=128
port_pending_tx_number	Numeric	port_pending_tx_number 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 6.96. set pp_1 port_pending_tx_number=128
suspend_msg_on_mac_change	Boolean	To avoid confusion arising from non critical errors during MAC address change, by default, logging is suppressed (default value is true). To enable tracing during MAC address change set this parameter to false Example 6.97. set pp_1 suspend_msg_on_mac_change=false

6.13.3.2.2. NDIS5/NDIS6 CHARON Packet Port mapping

packet_port parameter	Type	Description
interface	Text string	This parameter Identifies the dedicated Ethernet adapter in the Windows host system. Syntax: set <name> interface="connection:<connection>"

packet_port parameter	Type	Description
		Example 6.98. set pp_1 interface="connection:CHARON"

Example 6.99.

load DE500BA/dec21x4x IFC

load packet_port IFC0 interface="connection:CHARON"

set IFC interface=IFC0

Example 6.100.

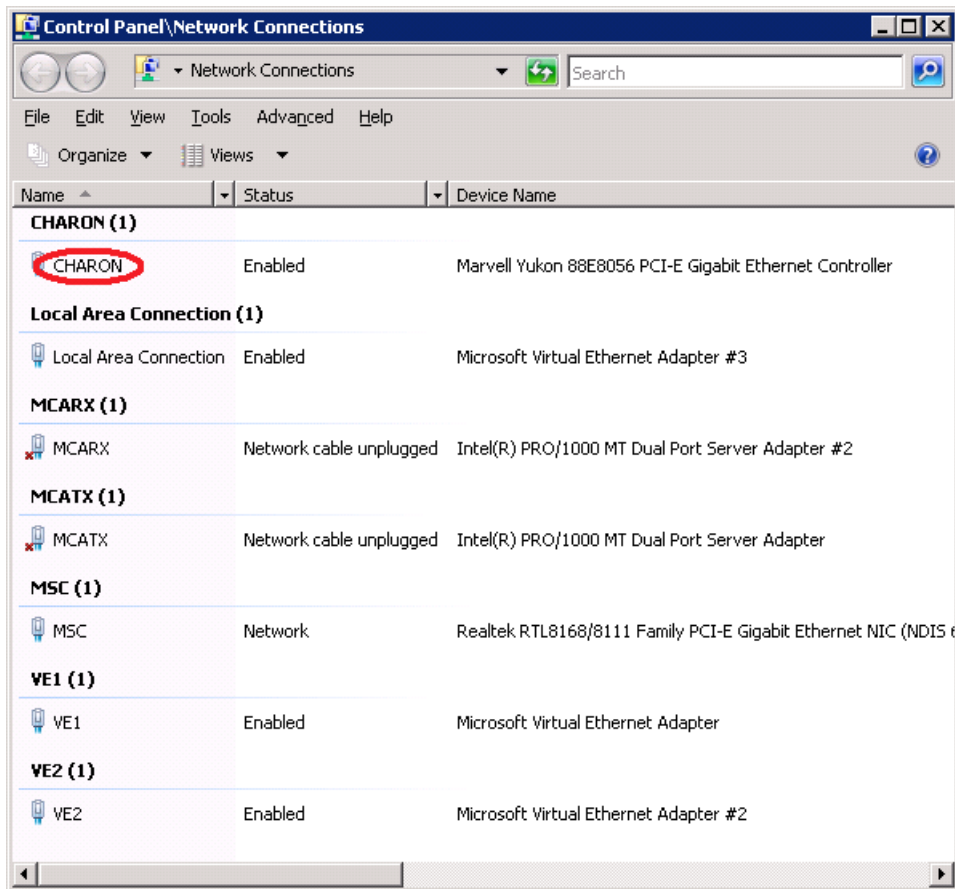
load DEQNA XQA

load packet_port XQA0 interface="connection:CHARON"

set XQA interface=XQA0

The Network connection name to be entered in the configuration file can be obtained from:

- Network Connections on systems prior to the Windows Server 2008 R2/Windows 7/Windows 2012
- Control Panel\Network Connections on Windows 7, Windows Server 2008 R2 and Windows 2012 (activated through Control Panel\Network and Sharing center)

Example 6.101. for Windows 2008 R2

This dialog suggests the following configuration lines:

```
load DE500BA/dec21x4x IFC
```

```
load packet_port IFC0 interface="connection:CHARON"
```

```
set IFC interface=IFC0
```

For the extended set of the port configuration parameters, tuning and troubleshooting please refer to 'CHARON-VAX and CHARON-AXP/4100/DS/ES/GS networking Guide'

6.14. Serial lines support in the virtualization layer

6.14.1. Virtual PBXDA-xx series PCI serial adapters based on DIGI AccelePort adapters

CHARON supports virtual PCI PBXDA-xx series serial adapters based on DIGI AccelePort serial adapters in Pass Through mode.

6.14.1.1. Attaching virtual PBXDA-xx series PCI serial Adapter to virtual system

The CHARON PCI Pass Through mode enables connection between the virtual DEC PBXDA-xx adapter and the physical DEC PBXDA-xx adapter plugged into a host's PCI bus.

To create an instance of virtual PBXDA-xx series PCI serial Adapter use “**load**” command in configuration file as follows:

```
load defpa <instance-name>
```

Note that **<instance-name>** is not visible outside configuration file. Operating systems running on top of virtual system use different naming policy and name assigned to virtual PBXDA-xx series PCI serial Adapter by those operating systems has nothing to do with **<instance-name>** assigned in configuration files.

Example 6.102.

```
load digi SERIAL_A
```

In this example, **SERIAL_A** is instance name of virtual PBXDA-xx series PCI serial Adapter. But HP OpenVMS operating system uses names TXA, TXB, TXC, ... to identify instances of virtual PBXDA-xx series PCI serial Adapters

6.14.1.2. Configuring virtual DEFPA PCI FDDI Adapter

Virtual PBXDA-xx series PCI serial Adapter offers several configuration parameters controlling its behavior in virtual hardware and its appearance to software running on it (e.g. HP OpenVMS Alpha and HP Tru64 UNIX operating systems).

digi parameters	Type	Description
host_bus_location	Text string	Enables usage of CHARON PCI Pass Through and specifies physical location of physical PBXDA-xx series PCI serial adapter. Example 6.103. load digi SERIAL_A host_bus_location="PCI bus 3,device 1,function 0"
bus	Text string	When specified, the bus configuration parameter tells the CHARON software the virtual PCI bus to which the virtual system shall connect the virtual PBXDA-xx series PCI serial Adapter. By default the bus configuration parameter is not specified. If the bus configuration parameter is not specified, the CHARON software connects the virtual PBXDA-xx series PCI serial Adapter to the first available virtual PCI bus. Example 6.104. load digi SERIAL_A bus=pci_1 device=1 function=0

digi parameters	Type	Description
device	Numeric	<p>When specified, the device configuration parameter specifies position of the virtual PBXDA-xx series PCI serial Adapter on virtual PCI bus.</p> <p>By default the device configuration parameter is not specified.</p> <p>If the device configuration parameter is not specified, the CHARON software connects the virtual PBXDA-xx series PCI serial Adapter at the first available position of the virtual PCI bus.</p> <p>Example 6.105.</p> <p>load digi SERIAL_A bus=pci_1 device=1 function=0</p>
function	Numeric	<p>When specified, the function configuration parameter specifies position of the virtual PBXDA-xx series PCI serial Adapter on virtual PCI bus.</p> <p>By default the function configuration parameter is not specified.</p> <p>If the function configuration parameter is not specified, the CHARON software connects the virtual PBXDA-xx series PCI serial Adapter at the first available position of the virtual PCI bus.</p> <p>Example 6.106.</p> <p>load digi SERIAL_A bus=pci_1 device=1 function=0</p>
irq_bus	Text string	<p>When specified, the irq_bus configuration parameter specifies virtual bus routing interrupt requests from virtual PBXDA-xx series PCI serial Adapter to virtual Alpha CPUs in Virtual HP Alpha system.</p> <p>By default the irq_bus configuration parameter is not specified.</p> <p>The irq_bus configuration parameter must be set to "ISA" for virtual PBXDA-xx series PCI serial Adapter in virtual AlphaServer 400. For virtual HP Alpha systems other than AlphaServer 400 the irq_bus configuration parameter must be left as is (i.e. not specified).</p> <p>Example 6.107.</p> <p>load digi SERIAL_A irq_bus=isa</p>
irq	Numeric	<p>When specified, the irq configuration parameter assigns interrupt request to the virtual PBXDA-xx series PCI serial Adapter in Virtual HP Alpha system.</p> <p>By default the irq configuration parameter is not specified.</p>

digi parameters	Type	Description
		<p>If the irq configuration parameter is not specified, the CHARON software uses the correct value depending on the selected PCI position of virtual PBXDA-xx series PCI serial Adapter in the virtual system.</p> <p>Example 6.108.</p> <p>load digi SERIAL_A bus=pci_1 device=1 function=0 irq=24</p>

Example 6.109.

load digi SERIAL_A host_bus_location="PCI bus 3,device 1,function 0"

set SERIAL_A bus=pci_1 device=1 function=0 irq=24

set SERIAL_A irq_bus=isa

Note

CHARON PCI Pass Through driver must be installed and running for physical DIGI adapter

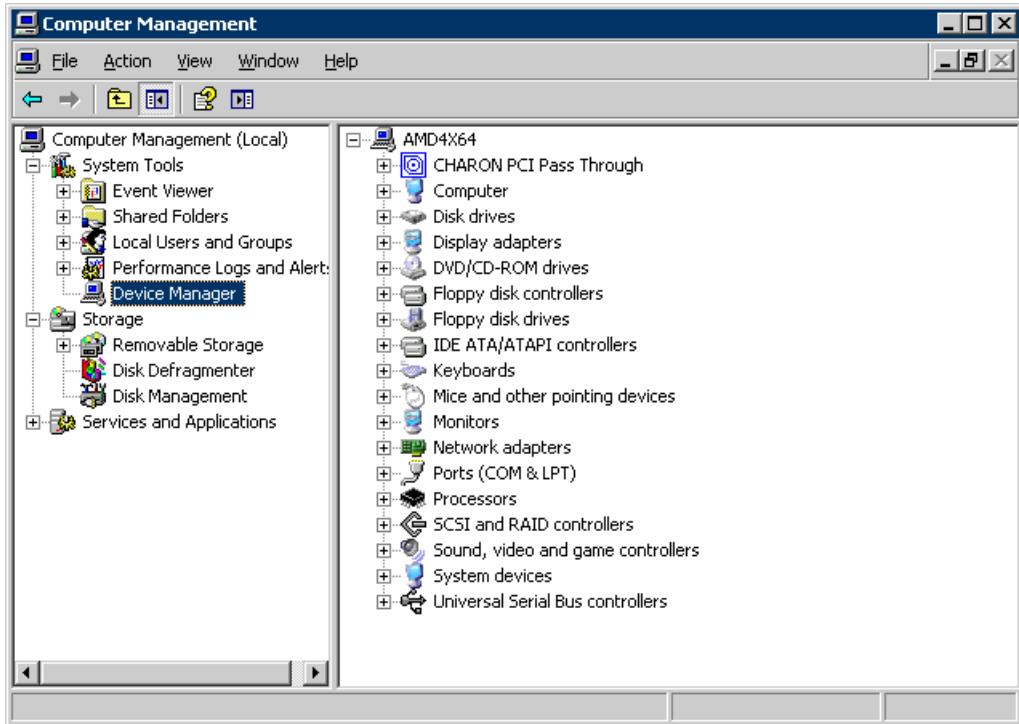
Installation of the driver:

1. Open "Computer Management"
2. Select "Device Manager"
3. In the right window select the desired physical FDDI adapter connected to the system, right-click the mouse button, and the corresponding menu will appear.
4. From the menu select "Update driver...". Windows will show "Hardware Upgrade Wizard"
5. Select "No, not this time", click "Next"
6. Select "Install from a list or specific location (Advanced)", click "Next"
7. Select "Don't search. I will choose the driver to install", click "Next"
8. Click "Have Disk...". Windows shows dialog "Install From Disk"
9. Instead of "A:\", click "Browse" and select path to the folder in which driver's INF file is located, select "*defpa_ppt_amd64.inf*", and click "Open"
10. The "Hardware Upgrade Wizard" should have "**CHARON DEFPA FDDI adapter**". Select it, and click "Next"
11. There will be one or two more dialogs, but they are usual for device driver installation.

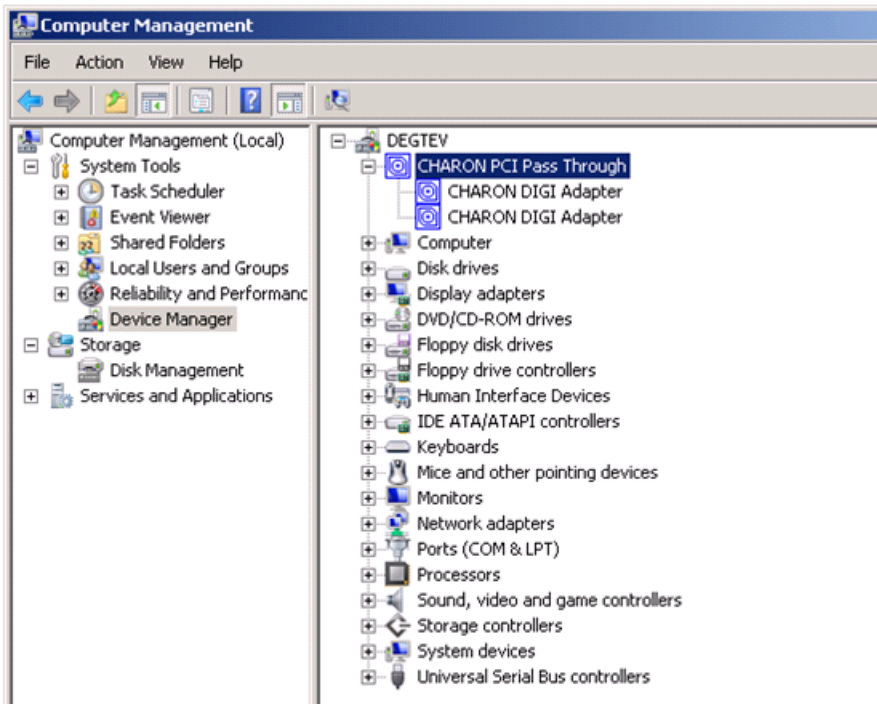
Upon completion, a new device will appear in the device manager with the CHARON logo on it.

6.14.1.3. Defining parameters for the "host_bus_location"

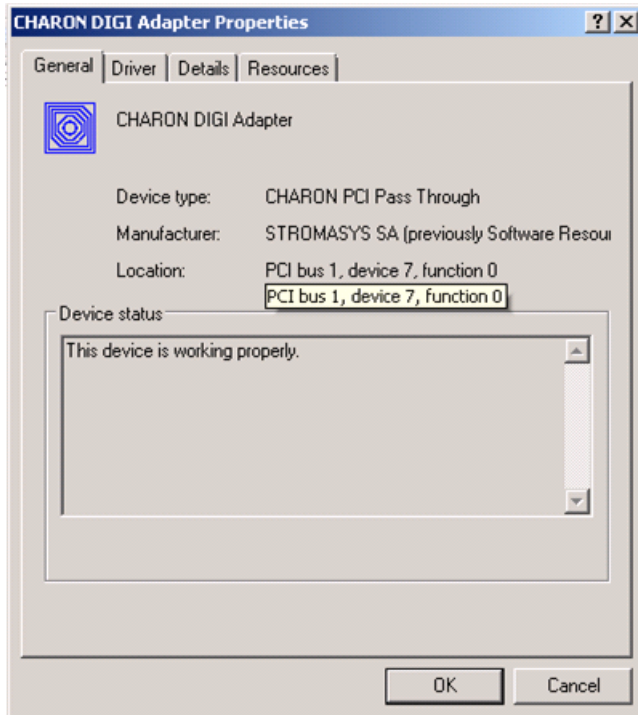
Open "**Computer Management**" application and select "Device Manager":



On the right panel please select the proper physical DIGI adapter:



And open its property sheet by double-click on selected adapter:



The “*Location:*” on the above picture gives **X**, **Y**, and **Z** for the **host_bus_location** parameter.

Note

Non-US-EN installations of Windows may present “*Location:*” string in local language, but **host_bus_location** parameter requires English notation, that is words “*PCI*”, “*bus*”, “*device*”, and “*function*” must be specified in English.

6.14.1.4. Compatibility

The following physical DEC PBXDA-xx adapters are supported by virtual DEC PBXDA-xx adapter in CHARON PCI Pass Through mode:

DEC PBXDA-xx adapters	Name	Controller	Vendor ID	Device ID
PBXDA-BA	AccelePort 4r 920	ASIC PCI	114Fh	0026h
PBXDA-BB	AccelePort 8r 920	ASIC PCI	114Fh	0027h
PBXDA-AC	AccelePort Xem	ASIC PCI	114Fh	0004h
PBXDA-AC	AccelePort Xem	ASIC PCI	114Fh	0008h

Chapter 7. Operating CHARON

When CHARON starts, license checking takes a few seconds. If you remove the license key while CHARON is running, a warning message is given after a few minutes, and you have a maximum of 1 hour to save your files and shut down your virtual system.

If CHARON cannot start, you might have a license key enabled for a product version different than the installed version. Since CHARON has not yet read the configuration file with the log file definition, its log file cannot be updated. In this case the reason why CHARON does not start is logged in the Windows Application event log.

CHARON could be started manually as a Windows application, or can be defined as a Windows service with manual or automatic startup. The procedure to create and manage CHARON services is described below.

Note

It is possible to install several CHARON services at the same time (multi-instance mode) using the Launcher utility and then use the CHARON Service Manager to set "automatic" mode of execution for each CHARON service. In this case all CHARON instances could be scheduled for an automatic startup.

Note

CHARON Launcher, CHARON Service Manager and CHARON Network Center utilities require running them with all the administrative privileges to provide entire functionality.

Note, that even one is logged as "Administrator" on Windows 7, Windows Server 2008 R2, Windows Server 2012 or Windows 8/8.1 not all the administrative privileges are gained since with the User Account Control (UAC) turned on programs run without Administrator privileges by default.

So it is strongly recommended to start those utilities in the "Run as administrator" mode. To do that open up the folder containing the following executables:

- *Launcher.exe*
- *CharonManager.exe*
- *NetDiag.exe*

(By default they can be found in the "Utilities_1.0.XXX\x86" and the "Utilities_1.0.XXX\x64" subfolders of the CHARON installation folder)

Once the executables are found, click on them with a right button of mouse and choose "Run as administrator" option.

It is also possible to disable the "User Account Control: Run all administrators in Admin Approval Mode" feature completely. This feature is controlled by the "EnableLUA" entry in the following registry key: `HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Policies\System`

Chapter 8. CHARON-AXP Utilities

8.1. Overview

CHARON-AXP provides the following set of utilities:

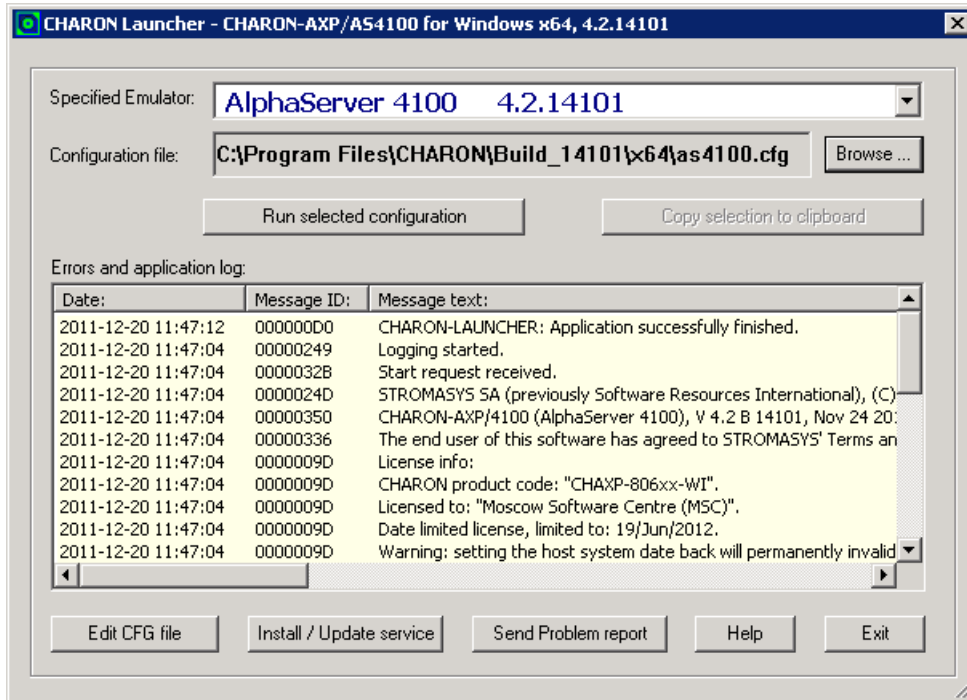
Utility	Description
CHARON Launcher	An GUI helping to run CHARON, see its constantly updated log file and install CHARON as service
CHARON Service Manager	An utility for managing installed CHARON services
CHARON Network Control Center	An utility for various operation for CHARON networking including CHARON network driver installation, dedication of chosen adapter to CHARON, monitoring CHARON networking, fixing problems on CHARON network adapter etc
MkDisk	An utility for creation of CHARON virtual disk containers of custom or standard types
MkDskCmd	A command line utility for creation of CHARON virtual disk containers of custom or standard types. This utility has an additional functionality to transfer virtual disks of one type to virtual disks of other type.
License Update Service	An utility for CHARON license management. It helps to collect host system fingerprint and information about existing license and apply license updates. It is also capable of transferring software licenses from one host to another one.
HASP View	An utility for viewing CHARON licenses
Host Device Check	An utility designed to locate the correct CHARON names for physical disks, tapes, CD-ROM drives, floppies and other devices found on host system
MTD	A command line utility for creating CHARON tape images from physical tapes and writing tape images back to a physical tapes
HOSTprint	An utility that receives data from CHARON LPV11 line printer and prints them on standard Windows printer
Idle	An utility that significantly reduces CHARON host CPU usage whenever a VMS system running on CHARON is idle

All the described utilities can be found in **Start->Programs->CHARON-><CHARON product>->Utilities** folder and - in case of command line utilities - in CHARON installation directory in subfolder *Utilities_1.XXX\i86* or/and *Utilities_1.XXX\i64*

8.2. CHARON Launcher

8.2.1. Overview

Use the CHARON Launcher to work with CHARON in the application mode. CHARON Launcher switches the interface language based on locale settings (English or Chinese PRC). Default interface language is English. Click on Help or press **F1** for help. Select the CHARON configuration file to run a particular CHARON instance. If more than one version of CHARON is installed, you can select the version to run from the drop down menu above the configuration file. The CHARON Launcher will display the model specified in this configuration file, if valid.



Click "Run selected configuration" and CHARON will start. It displays the CHARON log (including any configuration and run-time errors) in the Launcher Window. While CHARON is running, the Launcher updates the log file contents every 60 seconds. After CHARON stops, the Launcher loads the final application log contents for review.

The log file language depends on the locale settings of your system. Currently English, Chinese, Dutch, Spanish and Swedish versions are available.

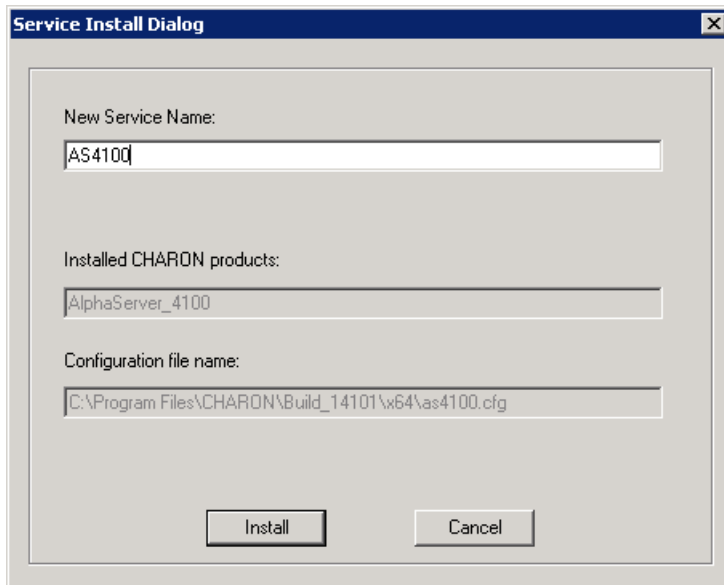
Click on "Edit CFG file" to edit the selected configuration file using the notepad editor. The ability to run a configuration, displaying the log and edit the configuration from a single interface makes the CHARON Launcher useful for debugging new installations/configuration files or examining error messages.

In addition, you can create a problem report by clicking on the "Send Problem report" button. Complete the message template (UNICODE format text file) by filling in the blanks in the draft message generated by CHARON. Send the problem report to your support team.

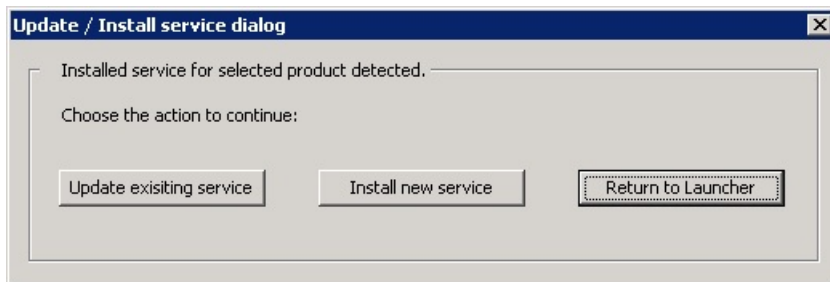
Use the "Copy selection to clipboard" button to copy selected lines of the log to the clipboard.

8.2.2. Installing CHARON as service

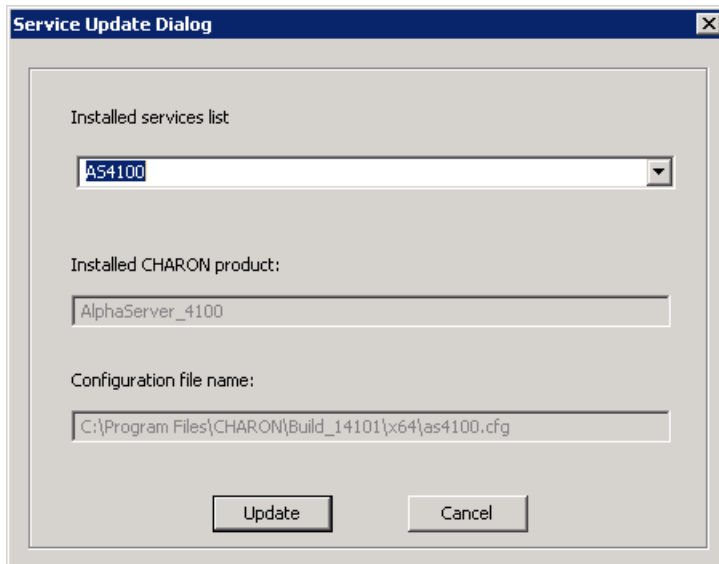
You can configure CHARON as a Windows service by pressing the button "Install / Update service". The following dialog is displayed:



Enter the desired service name in "New Service Name" and press the "Install" button. Note that the name of the service cannot have spaces. If the service already exists the following dialog is displayed:



If the "Install new service" button is pressed the Launcher displays the dialog for creating new service shown above. Otherwise the following dialog is displayed:



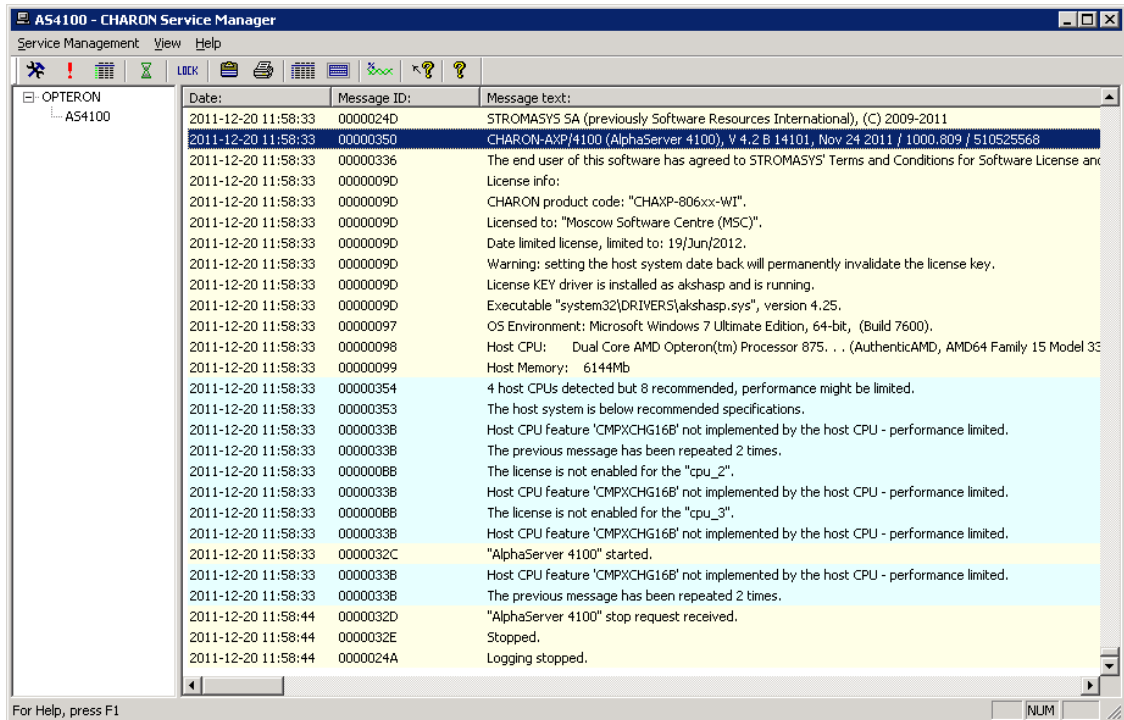
Press the "Update" button. Once the "Service Update Dialog" has disappeared press the "Return to Launcher" button in the "Update / Install Service Dialog" window. The Launcher will update the service and inform you if the service was successfully updated.

8.3. CHARON Service Manager

8.3.1. Overview

The CHARON Service Manager manages CHARON services (i.e. specific models configurations) available on your computer. The utility starts automatically on login to the host system and creates an icon in the system tray. Click on the icon to invoke the CHARON Service Manager main window. If the utility is not running, it is possible to start it directly from the CHARON tray icon with the right click and the "Start" button.

Click Help or press **F1** for help.



In the Service Manager window, the tree structure on the left shows all CHARON services installed on the host system. Initially the right hand panel displays the product license key information. Clicking on a service name shows the most recent event log display for this service in the right hand panel.

8.3.2. Interface description

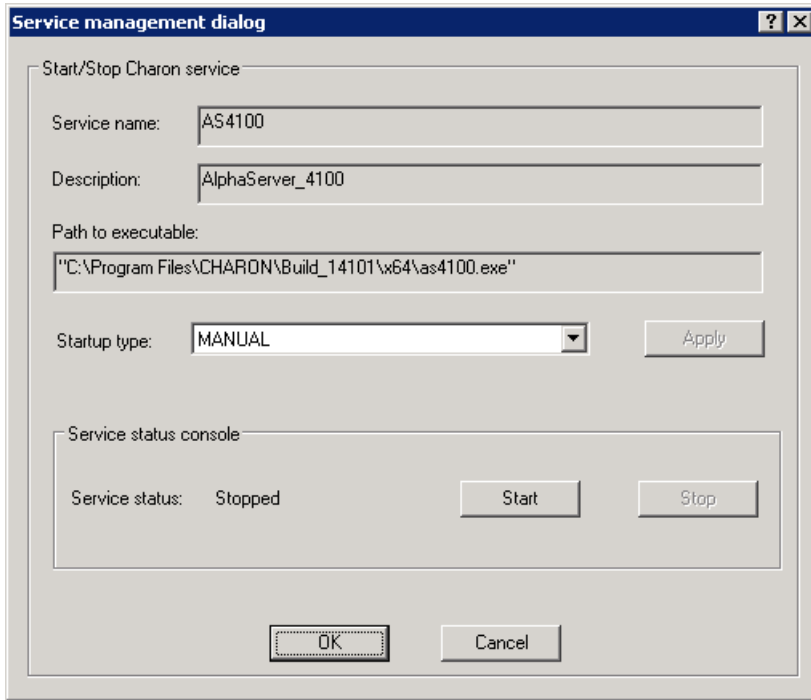
The buttons on the panel have the following meaning (left to right):

- Manage CHARON services opens a panel with the following options:

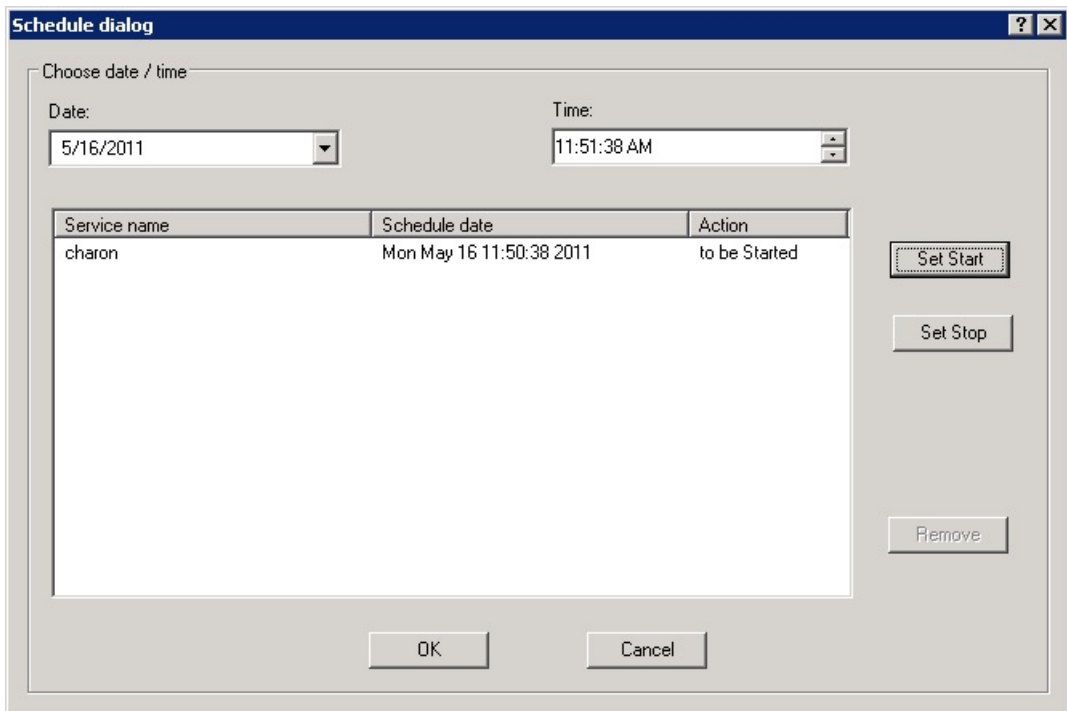
Service Name stands for the name of the chosen service, Description displays the CHARON model, and Path to executable refers to the executable to run.

Startup type can be *"Manual"*, *"Automatic"* or *"Disabled"*. Type or choose the desired option and press Apply to apply the setting.

Service Status provides information about the current status of the service. You can start the service or stop it, depending on its current status.

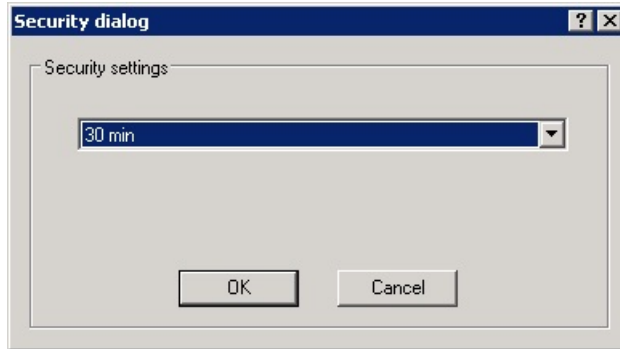


- Remove selected CHARON services removes the service you have currently selected.
- Update the list of installed CHARON services updates the service list shown in the left pane of the application window. This button is useful if you added new services while the CHARON Service manager runs. Added services are invisible until you restart the CHARON Service manager or update this list. This is also available through the corresponding item on the "Service Management" submenu.
- Schedule start service (also available through the corresponding item on the "Service Management" submenu) sets the start and, if necessary, stop time of a selected service. It invokes the following dialogue:



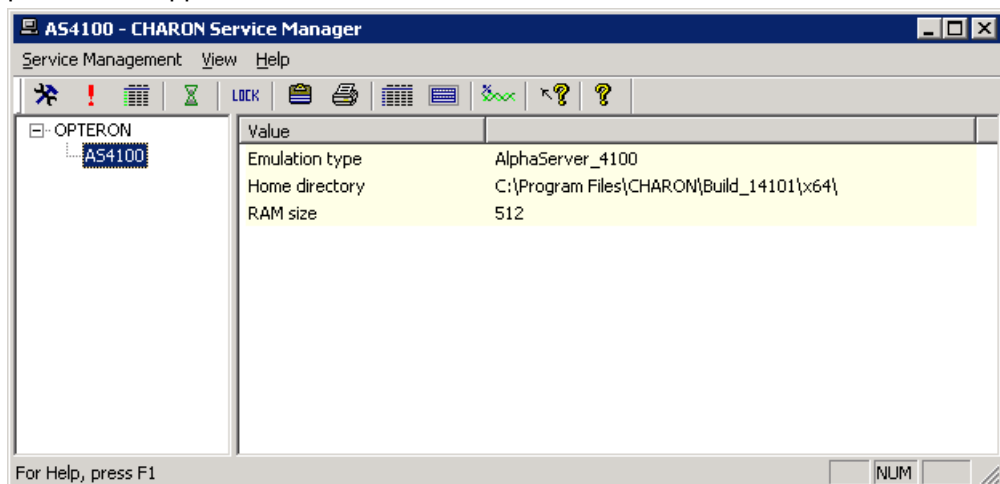
Press the Set start or Set Stop buttons to apply the date and time. To remove any schedule date, select it and press the Remove button. Note that all scheduled tasks run only if the CHARON Service manager is running. No scheduled information is stored in the system registry. When the CHARON service manager restarts, the scheduled task queue will be empty.

- Security setting (also available through the corresponding item on the "Service Management" submenu). This function locks the computer after a specified amount of time.



Enter the Windows User name/password combination in the standard Windows box to regain access. By default this function is disabled.

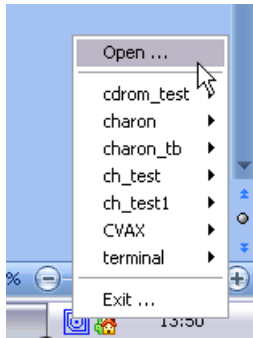
- Mail Problem Report (also available through the mail item on the "Service Management" submenu) automatically creates a problem report template.
- Print the active document opens up the standard printer dialogue to print the right hand panel of the application. Use the submenu "Service Management" to customize printing through "Print Preview" and "Print Setup".
- Display service startup and error log displays the service startup and event log of a chosen service in the right panel of the application. This function is also available on the "View" submenu.
- Display the configuration displays the configuration of the selected service on the right panel of the application. This function is also available on the "View" submenu.



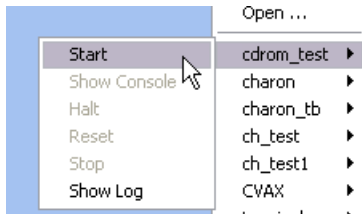
- The System Monitor invokes the Windows system monitor to trace the selected service's activity. This function is also available on the "View" submenu.
- The "question mark" button provides access to the help system.

8.3.3. Controlling CHARON Service Manager via system tray menu

It is possible to use the system tray menu to manage CHARON services w/o opening the main window of the utility. Click at it with the right button of the mouse and the following pop-up menu will appear:



The tray menu lists all the installed CHARON services. For each service the full range of essential operations is available:



So it is possible to start the service (“Start” option), stop it (“Stop” option), halt (“Halt” option) and reset (“Reset” option) it.

It is also possible to view the service log (“Show Log” option).

Option “Show Console” is very important if the CHARON service has console configured for terminal emulator (like the default “PuTTY”). If the service is set to “*Automatic*” mode the console will not appear on system reboot despite the fact that CHARON service will be up and running normally. To access the console use the “Show Console” option.

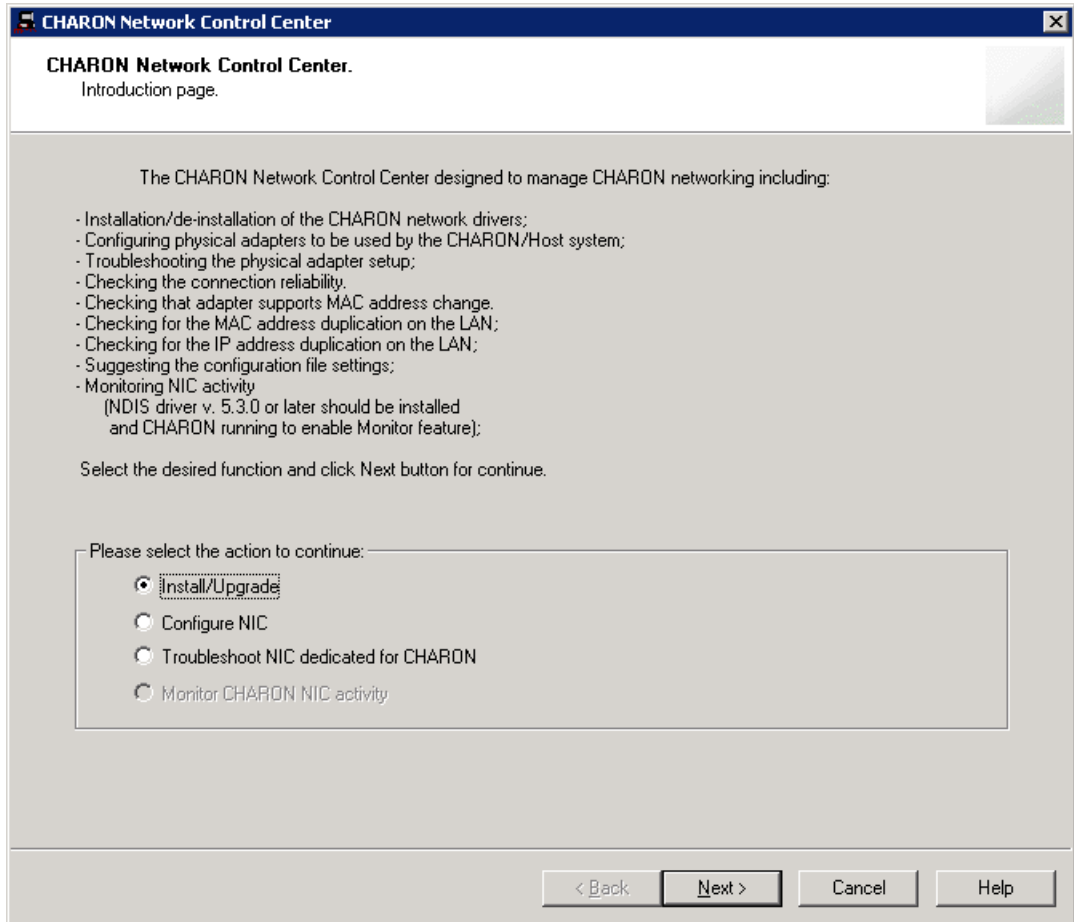
8.4. CHARON Network Control Center

8.4.1. Overview

The CHARON network control center performs the following operations:

- Install/uninstall/upgrade the NDIS5/NDIS6 Packet Driver
- Configure physical/virtual adapters for CHARON
- Troubleshoot physical/virtual adapters
- Check the connection reliability
- Check for MAC/IP address duplication on LAN
- Suggest configuration file settings
- Monitor network activity

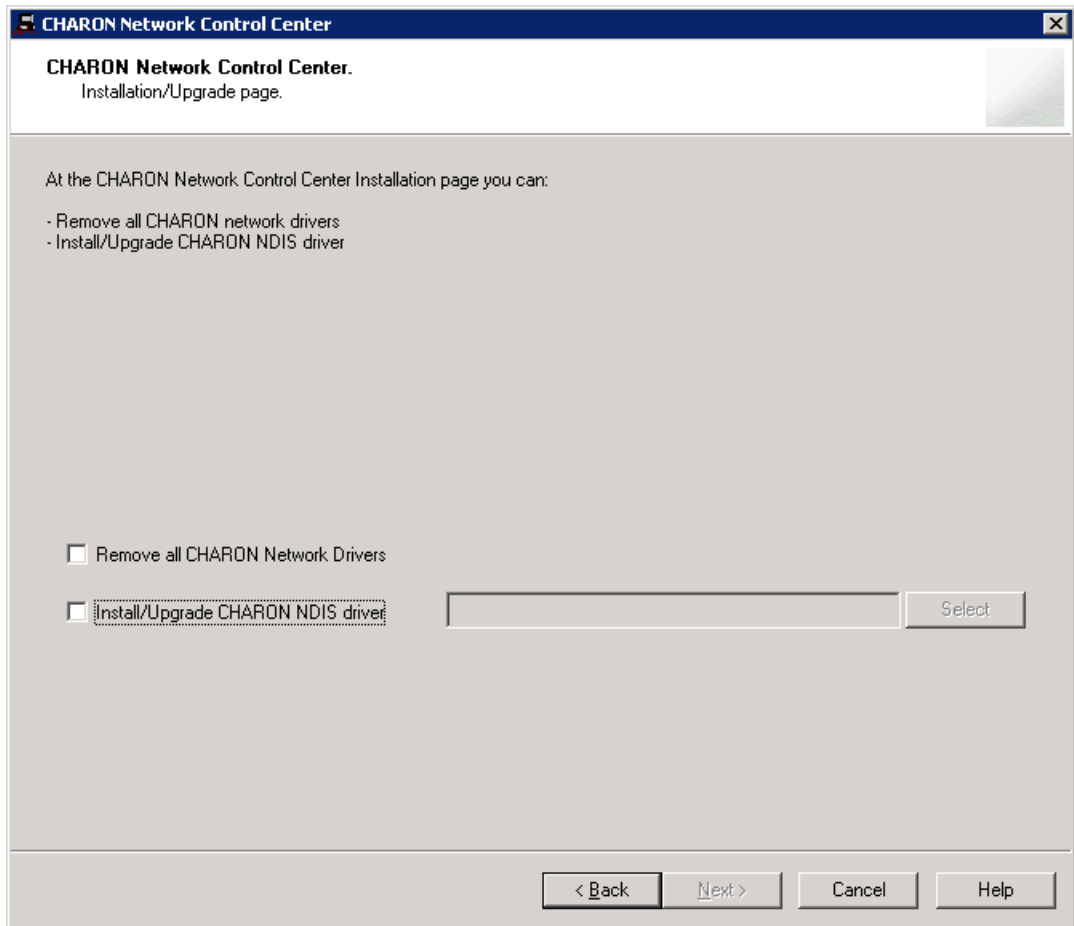
The start dialog of the utility provides a choice of the basic operations that can be performed:



Each dialog of the Network Control Center contains detailed information on available options and actions that could be performed.

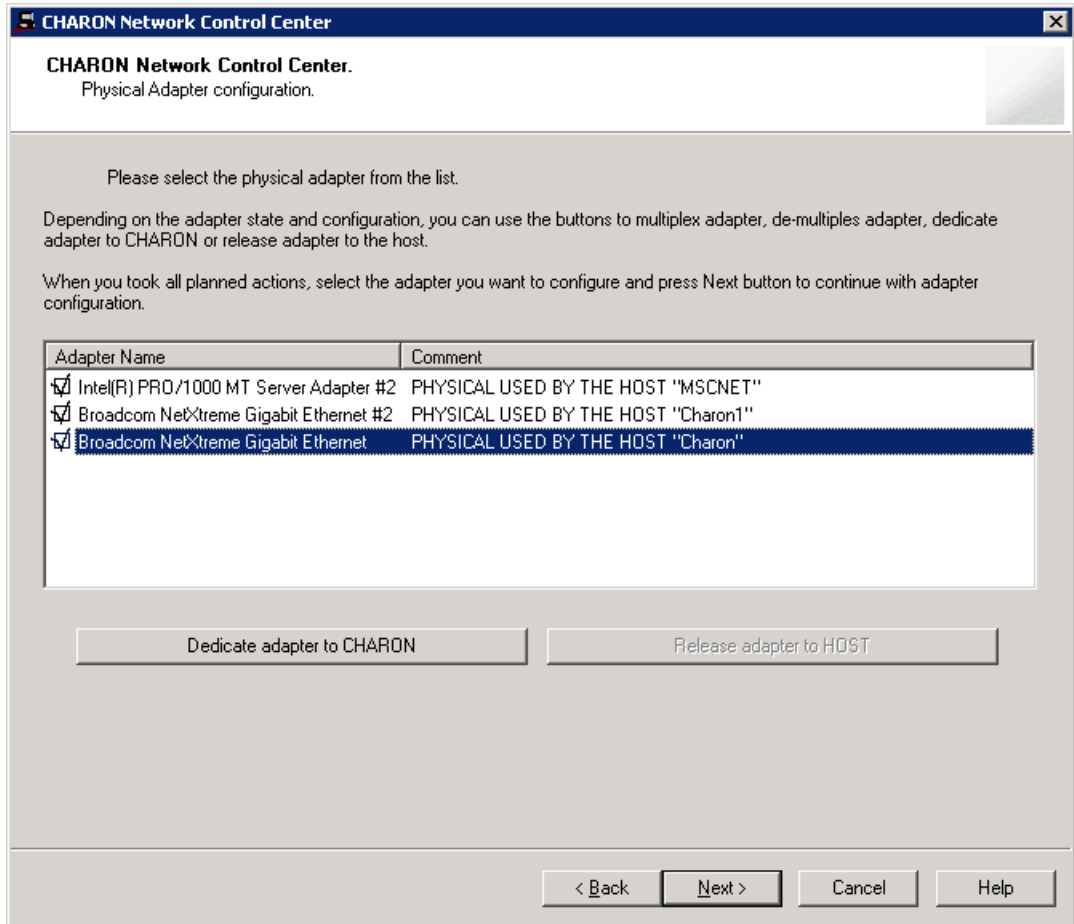
8.4.2. Install and upgrade CHARON network driver

Once "Install/Upgrade" is selected and the Next button is pressed, the following dialog appears for specifying the desired action and the paths to the drivers. Once the option is selected and the drivers are specified (if needed) just press the Next button to proceed.



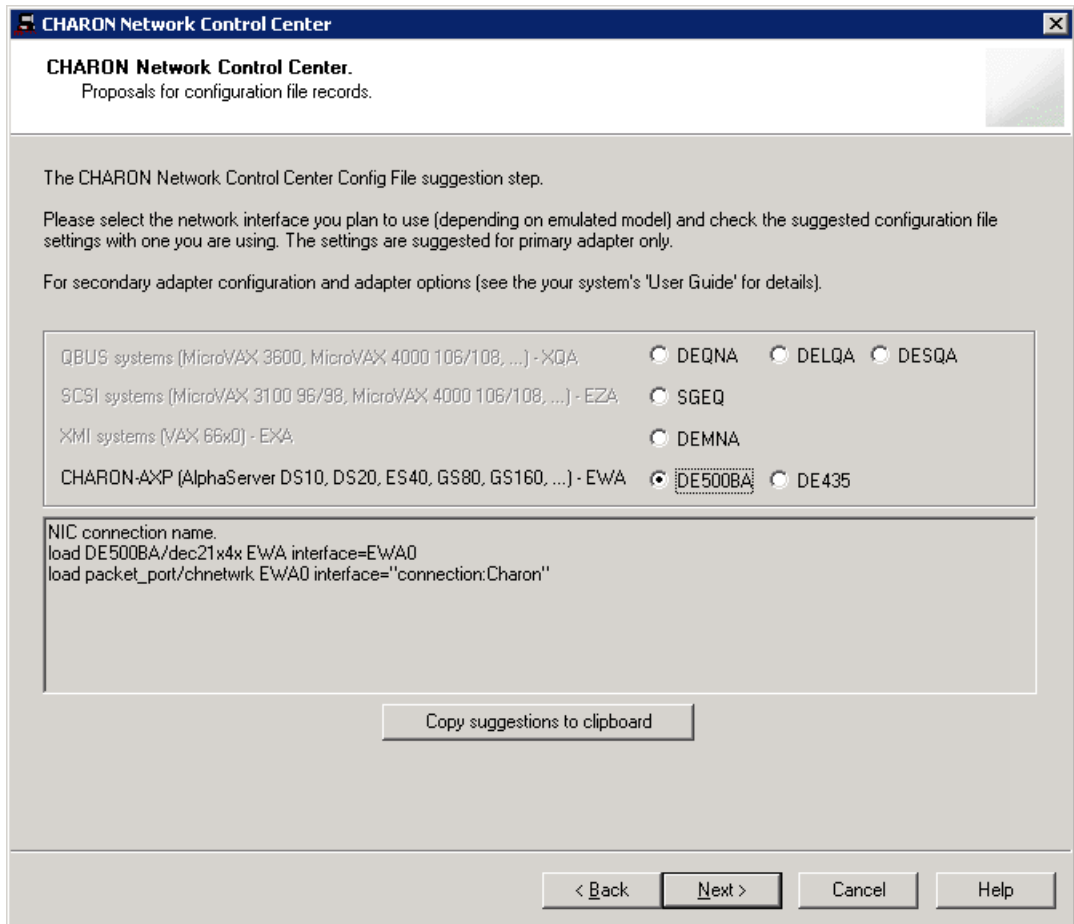
8.4.3. Configure NIC for CHARON

The following dialog is used for configuring a network adapter:



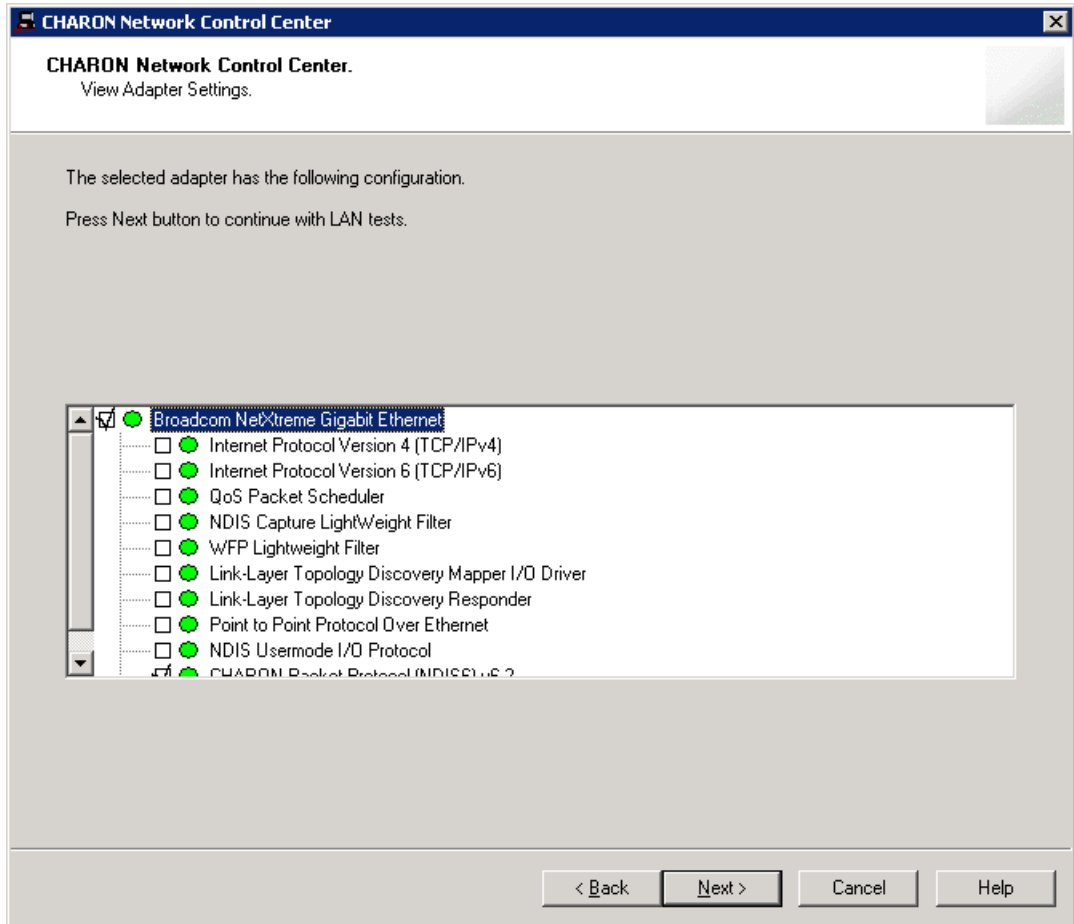
Select the desired adapter(s) to be used for CHARON and press Next to dedicate it to the emulator.

Once an adapter is dedicated to CHARON, the Network Control Center provides information to be inserted into configuration file for the adapter and various emulated adapters:

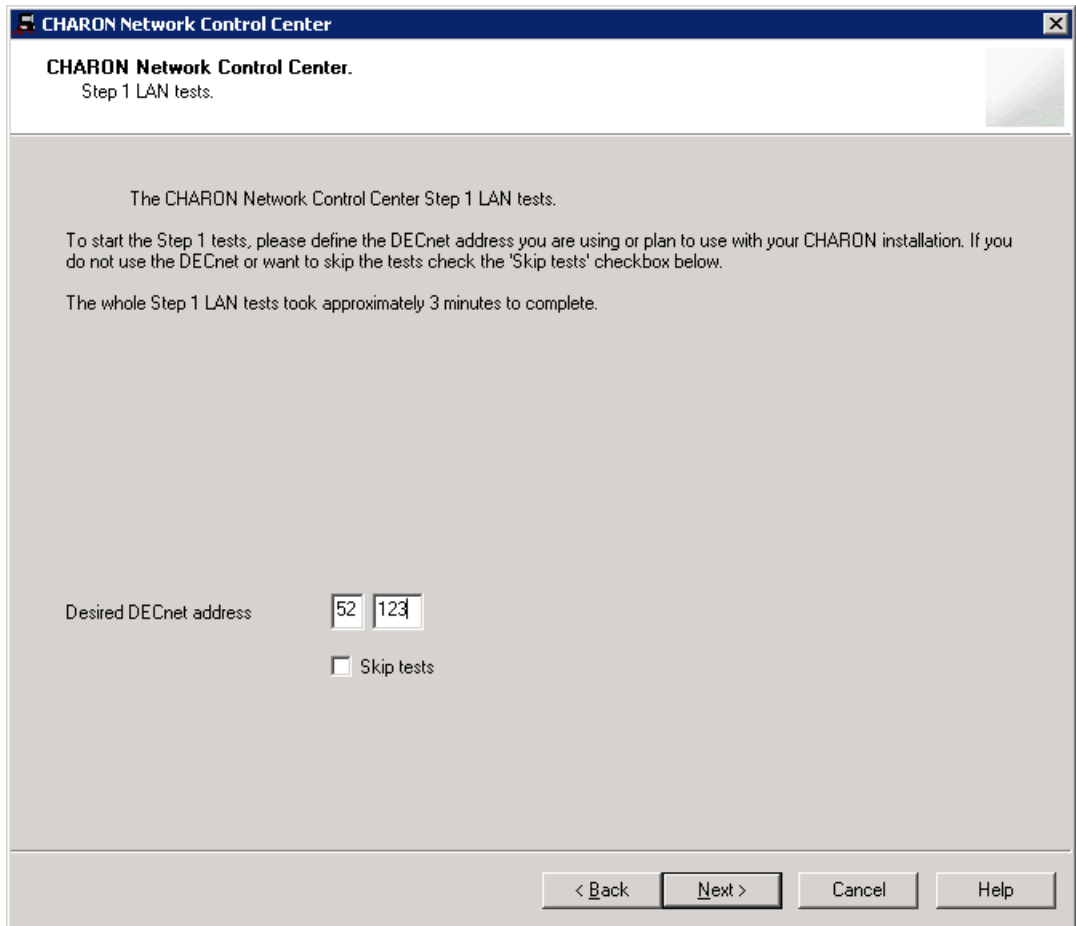


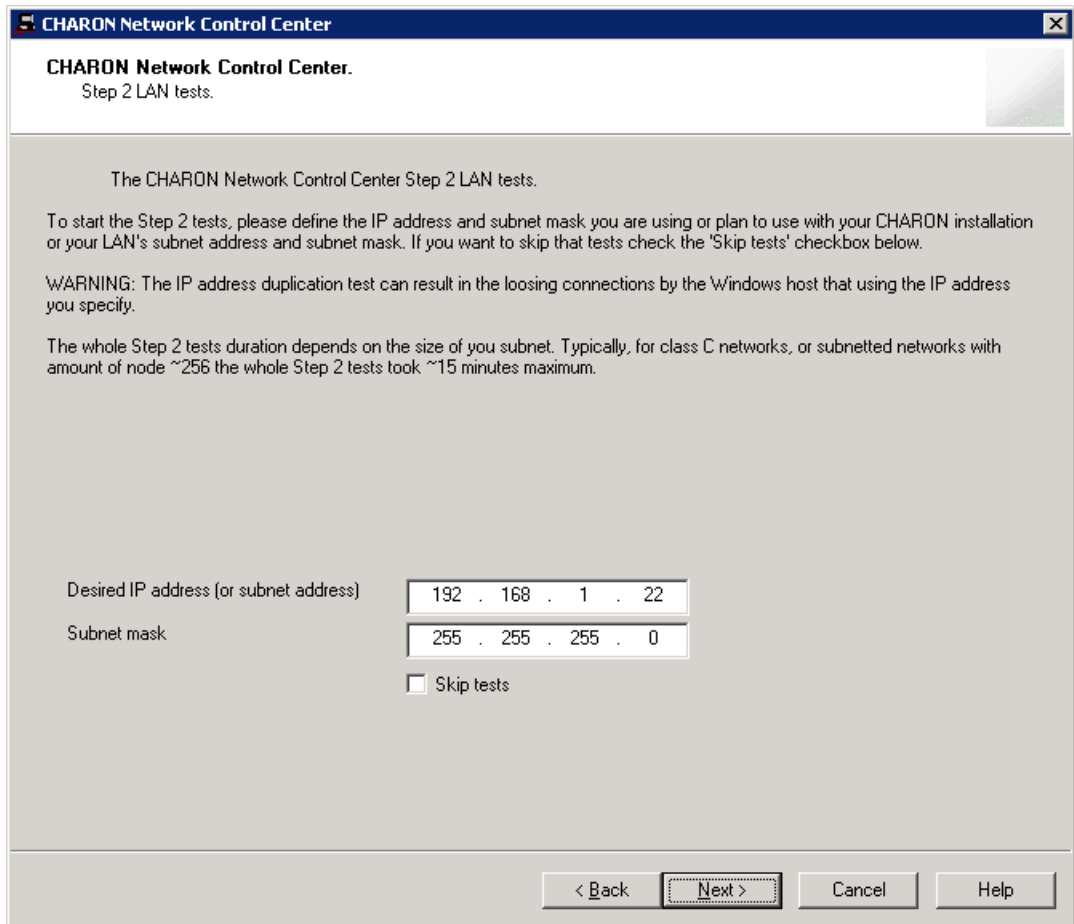
8.4.4. Troubleshoot NIC dedicated to CHARON

It is also possible to check the status of the adapter dedicated to CHARON. Green means that the adapter is ready for using with CHARON, red – that some problems are detected:



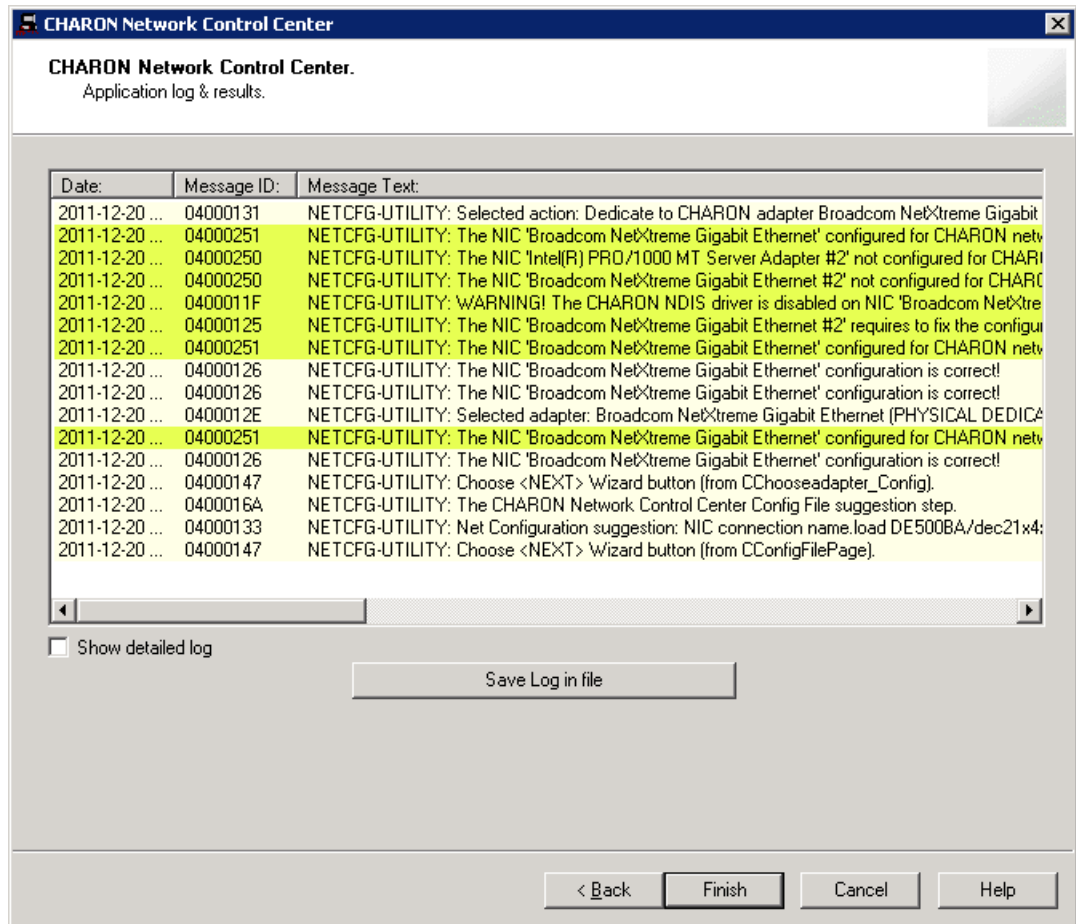
The following two screenshots demonstrate the ability to check whether the selected DECnet and TCP/IP addresses are available for CHARON:





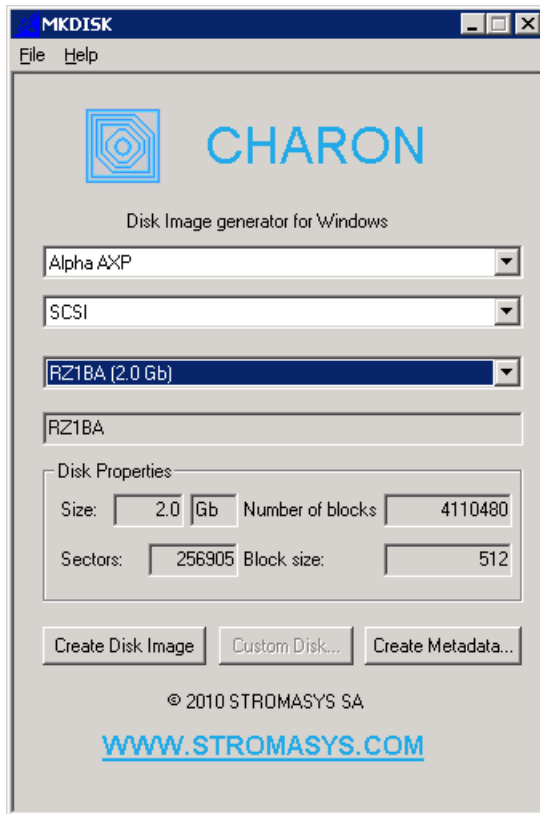
8.4.5. CHARON Network Control Center log

At the end of each operation for a network interface, the following dialog file is displayed, providing a log:



8.5. MKDISK utility

The MKDISK utility creates empty disk images of given standard disk drive types or of custom disk images. It is available in the utilities folder under **Start -> Programs -> CHARON -> <product> -> Utilities**.



Choose the system in the first drop-down box, specify "All controllers" in the second and choose the desired disk in the third one. After that press the Create Disk Image button, select the destination folder, enter the desired disk image name and press Save.

To generate disk images with any size use the "Custom" type and enter the number of blocks and block size. In the disk properties you see the size of the disk to be created.

8.6. "MkDskCmd" utility

The "MkDskCmd" command line utility creates empty disk images of given standard disk types or of custom disk size and can transfer existing disk images of one type to disk images of other type.

The utility is located in CHARON installation directory, in the "Utilities_XXX\1x86" subfolder. Open up "cmd" terminal and "cd" to that directory to use the "MkDskCmd".

The first step is obtaining the name of the disk that should be created:

```
MkDskCmd -list
```

This command results in getting a list of all supported disk types. Choose a desired disk (for example RZ22) and command the "MkDskCmd" to create a virtual disk image:

```
MkDskCmd -disk rz22 -output rz22.vdisk
```

The disk container "rz22.vdisk" will be created in the current directory.

Note

A file *rz22.avdisk* will be created in addition. This file helps CHARON to recognize a specific disk image type more accurately. So it is recommended to put the *.avdisk* file beside the created disk image.

It is also possible to create custom disk image using switches "*-blcount*" (blocks count) and "*-blsize*" (blocks size).

To get all the available parameters please use the switch "*-help*":

```
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

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

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

-transfer      - please see the '--transfer' options description
-t            - please see the '-t' options description

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

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

Note

The parameters "*-avtable*" is added for usage of an alternative disk specification database - or to point to the standard one ("*mkdsk.vtable*") if it is located in some other directory.

8.6.1. Transferring disk images

The "**MkDskCmd**" utility is able to transfer of a disk image of one type to a disk image of other type. This operation is needed for example to obtain more free space on a disk image already having some data.

Note

If a disk image is initially larger than the disk image it will be transferred to, the extra data is lost. If the disk image is initially shorter, it will be extended, and the extended part will be filled up with null bytes ('\0')

The syntax is following:

MkDskCmd -transfer <source disk file name> <source disk parameters>

where:

- *<source disk file name>* - a file name of the disk image to be transferred
- *<source disk parameters>* - the name of the disk from the list of available on "**MkDskCmd -list**" request or the disk geometry specification (see below).

Example 8.1.

```
mkdskcmd -transfer \etc\rz22.vdisk rz25
```

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

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

Example 8.2.

```
mkdskcmd -t \etc\custom.vdisk -z 512 -c 262134
```

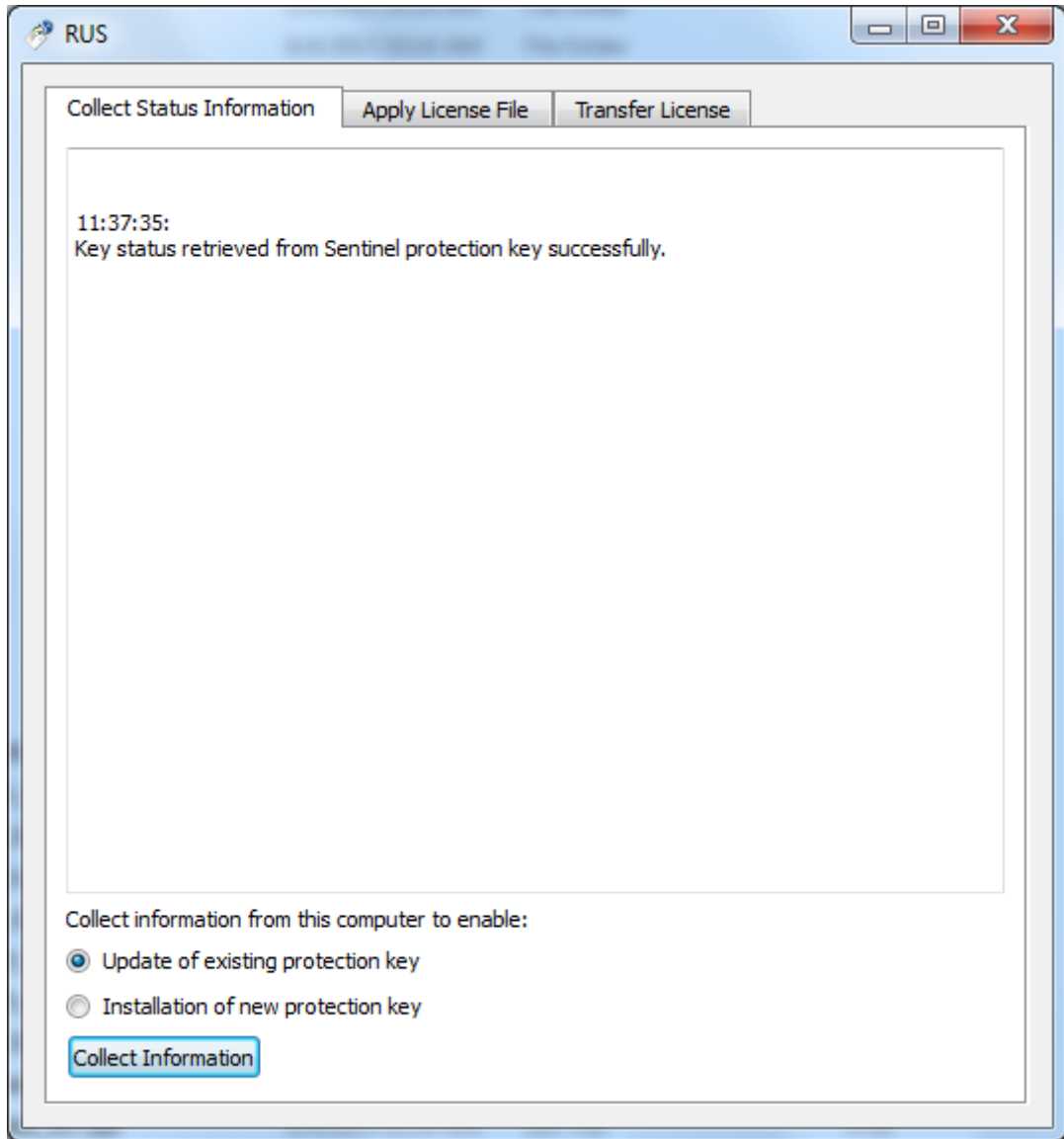
Note

There is a certain delay between a moment when the utility reports that a disk image has been transferred and its actual availability to CHARON. This delay can reach to several minutes in case of very big disks to transfer to. It happens because the host operating systems needs some time for actual allocation of the enlarged file on HDD.

8.7. License Update Service

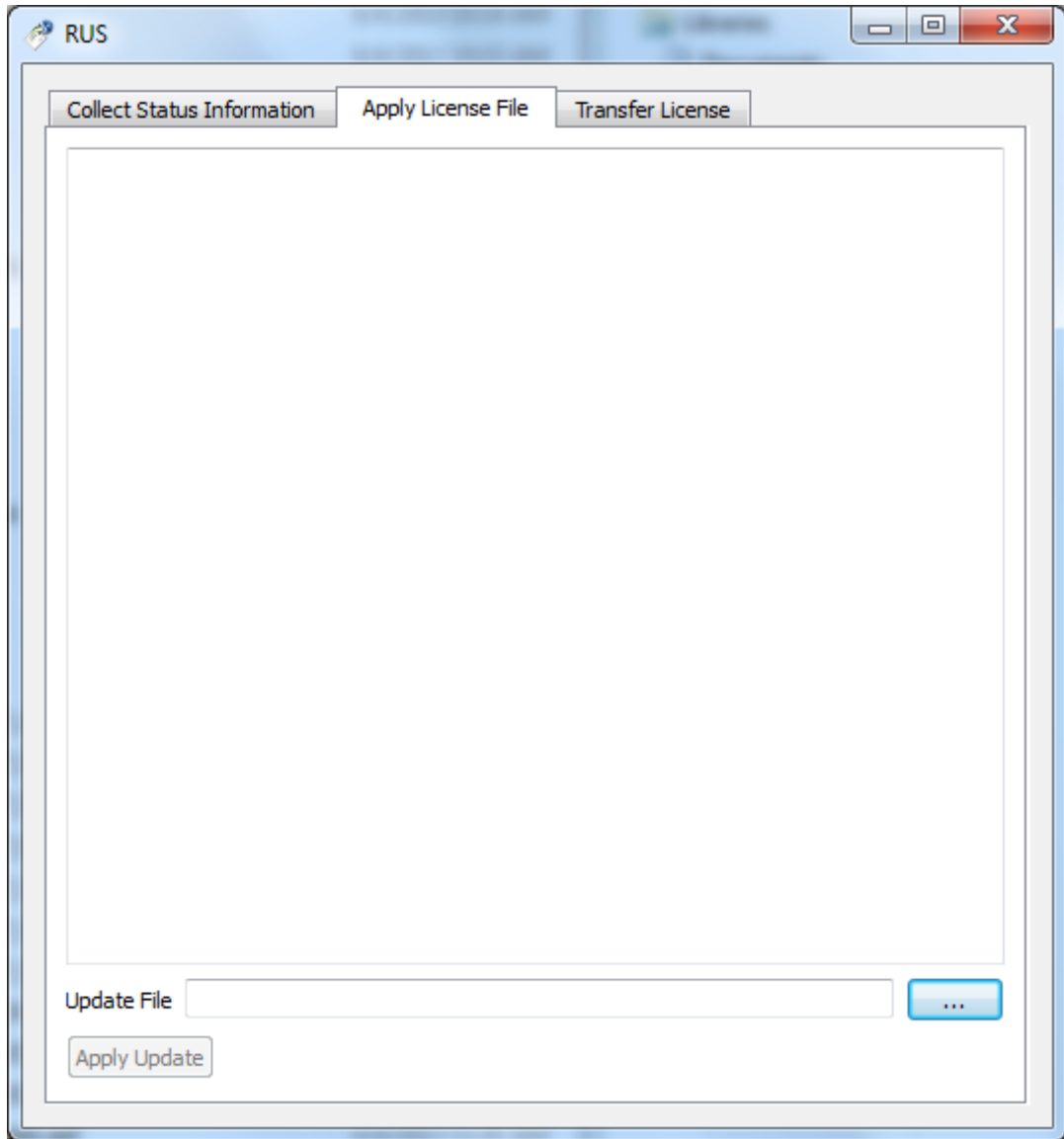
License Update Service allows applying a license update to the CHARON HASP license dongle and software license. It provides the following functionality:

- Collecting Sentinel HASP key license information and creating a special ".c2v" file which should be sent to STROMASYS®.



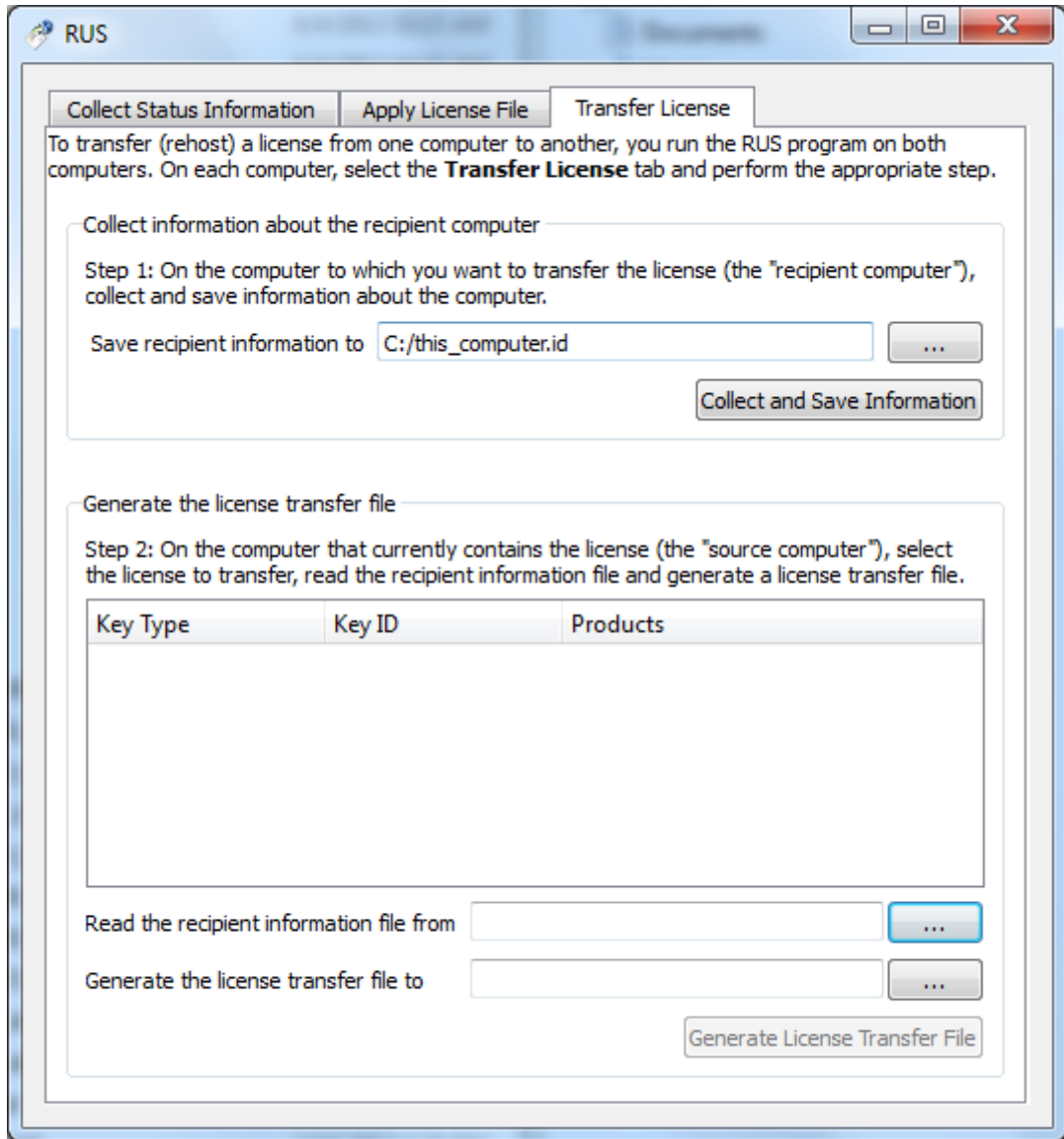
To collect license information select "Collect Status Information", choose whether updating of existing key or installation of new one is required and then press Collect Information button. The utility will ask for name of the ".c2v" file to be created and its location. Once the file is created it should be provided to STROMASYS® for getting update to the license.

- Updating CHARON license with ".v2c" file received from STROMASYS®.



To update the current license select "Apply License File", choose ".v2c" file provided by STROMASYS® with the "..." button at the lower edge of the dialog and press Apply Update. The License Update Service will display information whether the update was successful or not. In case of failure it is required to provide the displayed status to STROMASYS® to trace the updating problem.

- Transfer a software license from one host to another one



To transfer installed software license:

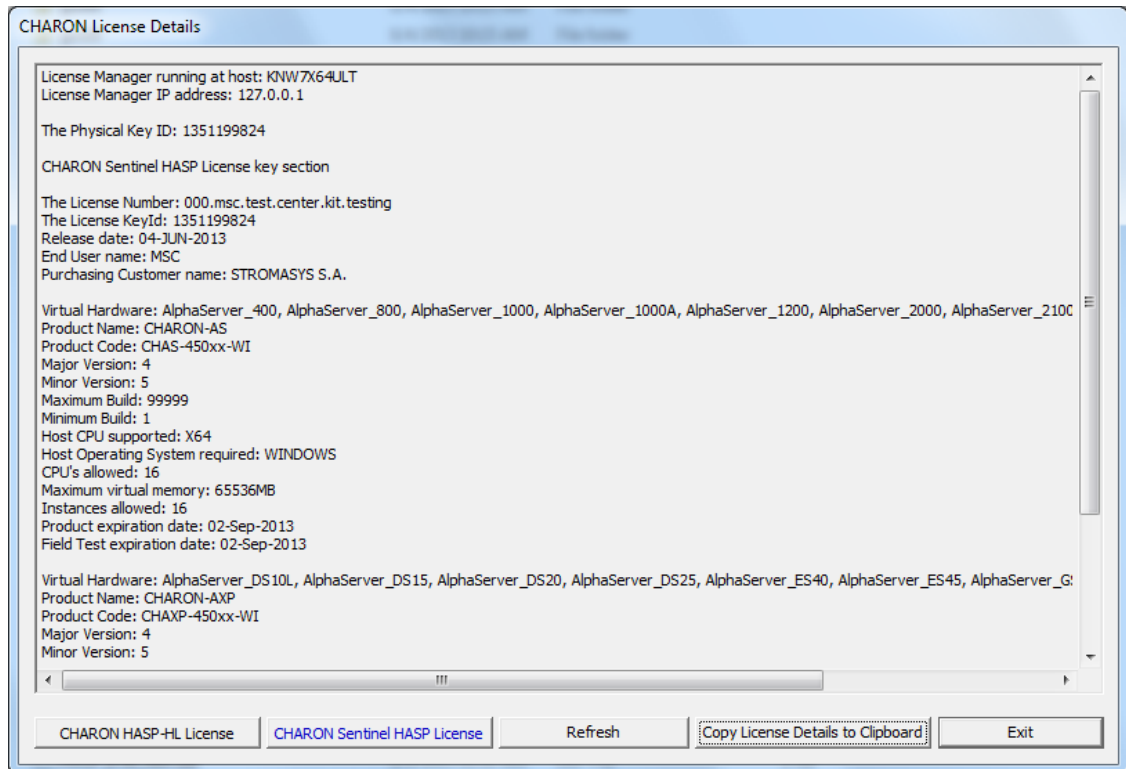
1. Run the **License Update Service** on the target host, select “Transfer License” tab and collect the host information into a specific *.id* file
2. Copy the target host information *.id* file to the source host
3. Run the **License Update Service** on the source host, select “Transfer License” tab, choose the license to transfer, specify the target host information file with the “...” button, choose some filename to store the license transfer file with the lower “...” button and press “Generate License Transfer File” button
4. Copy the license *v2c* file back to the target host and apply it as described before.

It is possible to update CHARON license “on fly”, while CHARON is running. But it is still recommended to stop the emulator, update the license and run the emulator again.

8.8. HASP View Utility

“HASP View” utility allows reading the content of CHARON license. The Utility reads connected dongles with HASP-HL or Sentinel HASP licenses, Network and Software licenses and

provides content of the license in its main window. The license text can be scrolled up and down and copied to clipboard. Use Refresh button to reread the license dongle.



HASP View Utility always reports Id and IP address of the hosts where active licenses are found. It helps in situation of multiple licenses.

8.8.1. Resolving problems with multiple licenses installed

In case of any problems relevant to multiple licenses (local/network/Sentinel SL) available for CHARON the only way to resolve them is to determine which license is connected to CHARON emulator.

To do that run the "HASP View" ("*hasp_srm_view*" in case of Linux) application. "HASP View" utility is based on the same license processing mechanism as the one implemented in CHARON emulators, but it provides detailed information about the license, the physical key connected or the software license available.

The utility is able to show the physical host ID (or its IP address) and the license key connected to it (or the Sentinel SL License installed). This functionality is very useful in case if any network license is available.

To disable all unnecessary licenses it is needed to make sure that the only one license key is connected to the local host or the only one Sentinel HASP Net key is available on network segment at the moment.

In case if some unnecessary networking licenses are present (detected by the "HASP View" application) it is needed to avoid local usage of the networking licenses. To do that use the Sentinel Admin Control Center application in the following way:

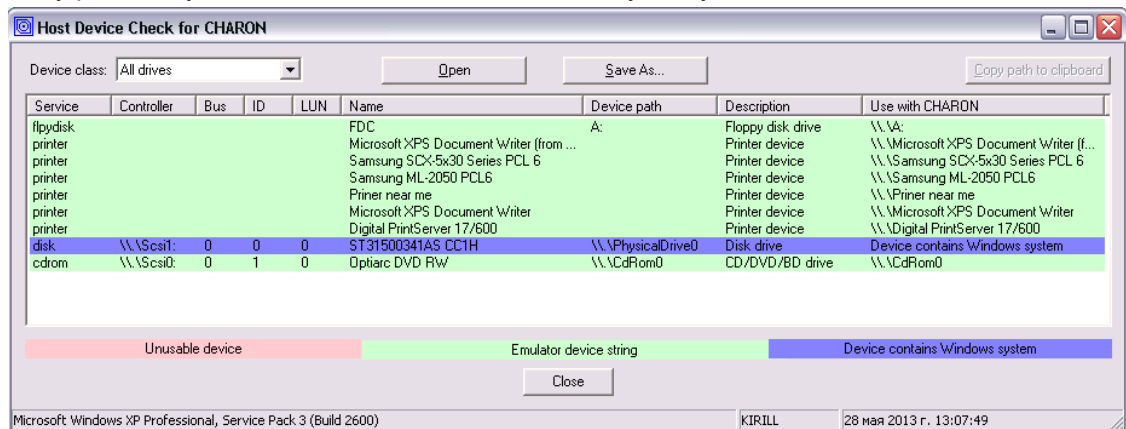
- Start internet browser and go to http://localhost:1947/_int_/config.html
- Choose the 'Access to Remote License Managers' menu item
- Disable (un-select) the 'Allow access to Remote Licenses' check box item

- Disable (un-select) the 'Broadcast Search for Remote Licenses' check box item
- Press 'Submit' button to save the new settings.

In case if a network license must be used, but more than one network key is connected just run the "HASP View" application sequentially, detect and disconnect unnecessary network keys until the only one network license key is left on the given network segment

8.9. Host Device Check utility

The Host Device Check utility is designed to locate the correct CHARON names for physical disks, tapes, CD-ROM drives, floppies and other devices found on the system. At startup the utility presents you with a list of disk drives found on your system:



The green color indicates the devices, which are not locked by the host operating system, thus can be used in the CHARON environment

It is possible to choose a specific device class from the following list: "Disk drives", "Tape drives", "CD-ROM drives", "Floppy disk drives", "Printers", "Unknown devices". You can select "All drives" to inspect your host system.

Note

The current version of Host Device Check utility does not correctly report iSCSI devices.

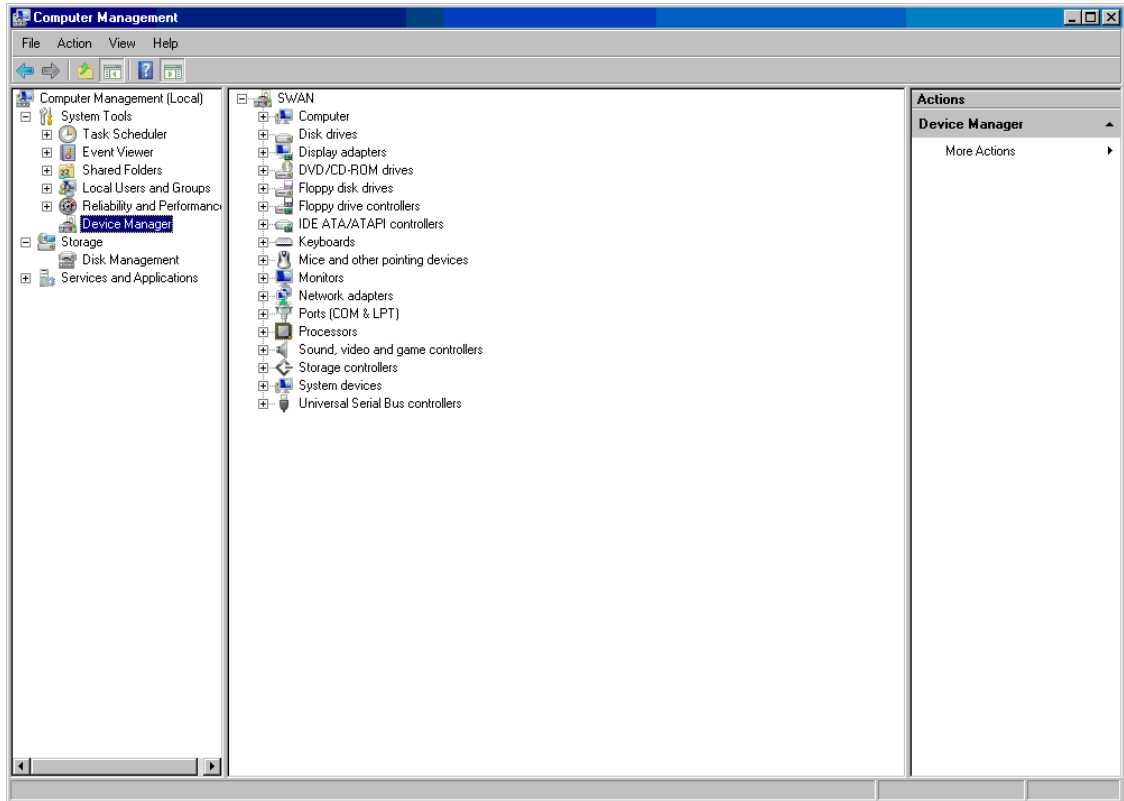
You can store the configuration as found with Host Device Check using the Save As... button and re-display this configuration by pressing the Open button.

Selecting any specific device from the list makes the Copy path to clipboard button available. Copying to clipboard helps to avoid a typing error when entering the device name into a CHARON configuration file.

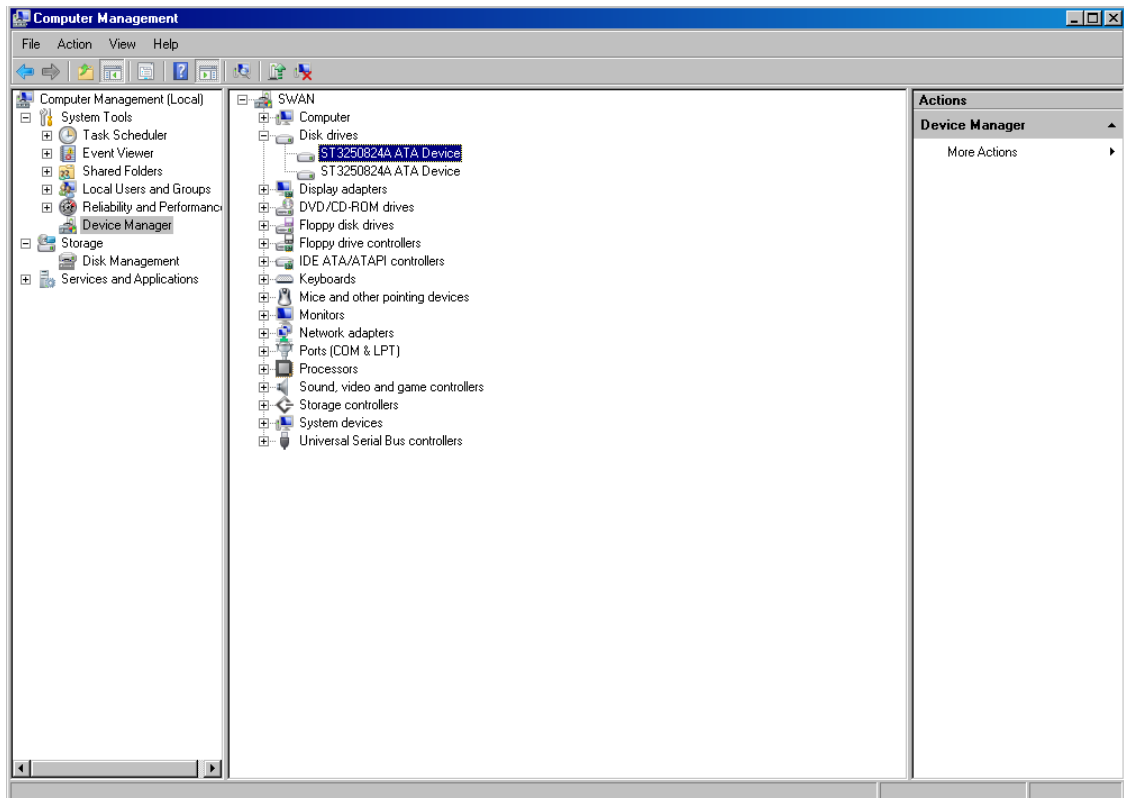
8.9.1. An alternative way to find out SCSI configuration details

If Host Device Check utility cannot be used for some reason (for example, for some kind of NAS / SAN network storage devices), the alternative way to find out SCSI configuration details it to use Windows build-in tools.

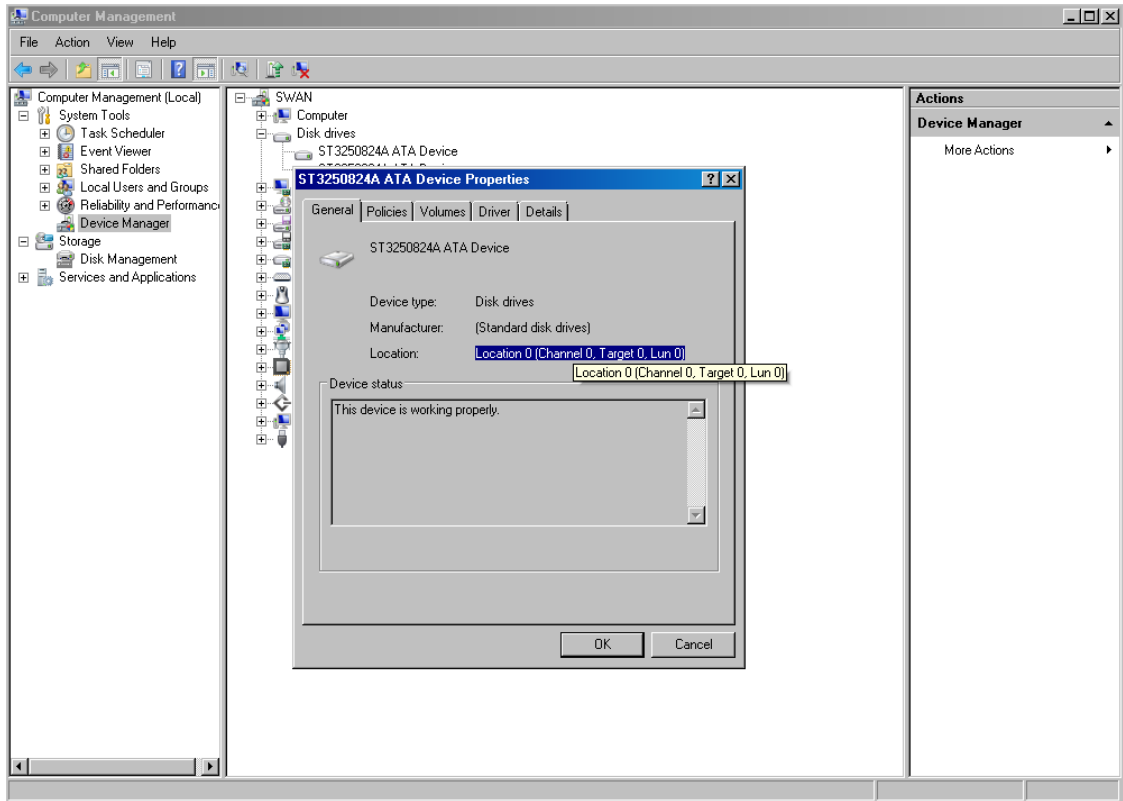
To find proper SCSI device information (to be used as a CHARON emulator configuration file parameter), open "Computer Management" application and select "Device Manager":



On the right panel please select the proper physical device:



Click right mouse button and choose the "Property" item at the pop-up menu appears.



The information from "Location" field to be used for selected device configuration.

8.10. MTD Utility

The "MTD" utility allows creating CHARON tape image from a physical tape and writing tape image to a physical tape. It is a command line utility. Usage is the following:

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

where the **option** are:

Parameter	Description
<code>/log = <file name></code>	Creates the execution log in the file " <i>file name</i> ".
<code>/reads = <number></code>	Specifies a number of attempts to read a damaged data block
<code>/ignore</code>	Directs to ignore bad blocks and continue processing w/o interruption

Example 8.3.

```
mtd \\.\Tape0 C:\TapeImages\tape1.vtape /log="tape1.txt" /reads=10 /ignore
```

Using the following syntax it is possible to write a content of a tape container to a physical tape:

mtd <tape container name> <tape device name>

Example 8.4.

```
mtd C:\TapeImages\tape1.vtape \\.Tape0
```

8.11. HOSTprint utility

HOSTprint is a Windows application that receives data from an emulated LPV11 on CHARON via a TCP/IP socket and prints the data received on the default Windows printer (if no printer specified at the utility command line) of the host computer.

There are two operation modes supported by HOSTprint application:

1. 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, you should properly configure the OpenVMS printing queue and use OpenVMS **"PRINT"** command.

2. Working in old version of the utility compatible printing mode

Access popup menu and set Flush buffer delay time. Recommended value is 5 seconds. Under OpenVMS type: **"COPY MY_PRINTED_FILE LPA0:"**

Both modes supports print preview of the last page in the printing buffer.

Command line parameters	Description
<code>-host=<hostname></code>	Name of the host - source of printing data
<code>-port=<connection port number></code>	Port on the host to get the information to print from
<code>-delay=<delay for automatically buffer flush in seconds></code>	Flushing delay, 0 - wait infinite, 5..10800 - timeout for flush
<code>-printer=[PrinterDeviceName]</code>	Host name for the printer used. Example 8.5. <code>-printer=[\\.\Microsoft Office Document Image Writer]</code> <code>-printer=[\print_server\MSCCLPS]</code>
<code>-font=<default font face></code>	Default font
<code>-fontsize=<default font size></code>	Default font size

2 last parameters are only for compatibility with old version of the utility (HOSTprint allow to change font settings from popup menu).

It is strongly recommended to used fixed-size fonts (by default the "Courier" font is used) to avoid any problems relevant to proper calculation of the printing line length.

Example 8.6. Configuration file examples

```
load chapi lpv1
```

```
set lpv1 dll=lpv11.dll
```



```
set lpv1 port=10015 application="hostprint.exe -port=10015 -printer=[\\print_serv-  
er\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_serv-  
er\MSCCLPS] -font=\Courier New\ -fontsize=12"
```

After initialization, HOSTprint creates the icon at the Windows toolbar. There are two colors of the icon:

- **GREEN** indicates Idle (or Ready) state
- **YELLOW** indicates 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 you can preview the last page content, change utility mode via the flush buffer delay and change the default font settings.

8.12. "idle" utility

The "Idle" utility significantly reduces the CHARON-AXP host CPU usage whenever a VMS/Alpha system running on CHARON-AXP is idle. "Idle" utility stalls the emulated CPU (note that at the moment it supports the models emulating just 1 CPU only, namely: AlphaStation 400, 800, 1000, 1000A and DS10L) when it detects an OpenVMS idle condition. While the "Idle" utility is running the emulated CPU consumes, on average, less host system CPU time. However it is not recommended to employ "Idle" utility in real-time process control environments.

The supported OpenVMS versions are from V6.2-1H3 up to V8.4. The provided PCSI distributive is used for all the versions of OpenVMS.

Note

On Linux this utility can be used only with models with single CPU emulation

The "Idle" utility is provided in form of a virtual disk image named "*idle_vms_pkg.vdisk*". Mount this disk with the "*over=id*" qualifier under the emulated VMS/Alpha operating system and go to the "*[000000.AXP]*" directory.

The following files are resided there:

README.TXT

SRI-AXPVMS-IDLE-V0102--1.PCSI

VMS62TO71U2_PCSI-V0200.PCSI-DCX_AXPEXE

VMS62TO71U2_PCSI-V0200.TXT

At the first step it is needed to apply a specific PCSI patch "*VMS62TO71U2_PCSI*" if the target VMS/Alpha operating system version is below V7.2. Copy the "*VMS62TO71U2_PCSI-V0200.PCSI-DCX_AXPEXE*" file to some directory on any spare disk and run this file from there:

```
$ RUN VMS62TO71U2_PCSI-V0200.PCSI-DCX_AXPEXE
```

then proceed with the patch installation:

\$ PRODUCT INSTALL VMS62TO71U2_PCSI /SOURCE=<directory containing the VMS62TO71U2_PCSI kit>

Once the installation is over please return to the "[000000.AXP]" directory of the "idle_vms_pkg.vdisk" and proceed with installation of the "Idle" utility itself:

\$ PRODUCT INSTALL IDLE /SOURCE=<directory containing the IDLE kit>

Once the "Idle" utility is installed it starts to take effect immediately, reducing the host system CPU usage if VMS/Alpha system running on CHARON-AXP is idle. No reboot is required. The utility is loaded automatically on reboot, no additional configuring or startup sequence is needed.

Deinstallation of the "Idle" utility:

\$ PRODUCT REMOVE IDLE

The utility stops working after the system reboot.

Please also refer to the supplied documents "README.TXT" and "VMS62TO71U2_PCSI-V0200.TXT" for more details.

Appendix A. Installing and transferring an original host software to CHARON

There are several ways to transfer data from an original system to CHARON:

A.1. Using Local Area Network

First, perform a standard installation of your host Operating System from the manufacturer's original media using CD-ROM drive. Then configure a network (DECnet and/or TCP/IP) to your CHARON for your existing Network with a unique address, and use DECnet or TCP/IP to copy your applications and data to your CHARON system. If for any reason installing a host Operating System from scratch is a problem, call your CHARON sales contact for help. Once you have CHARON connected to your network, you may use standard utilities to transfer the required data. Before copying the data you will have to configure CHARON with adequate free space on disks, or on disk images which can be created with the **MkDisk** (Windows) or "**mkdiskcmd**" (Linux) utilities.

A.2. Using a physical disk drive

You can remove a SCSI disk from your original system and reconnect it to a SCSI adapter on CHARON host operating system. Assign the SCSI disk within the CHARON configuration file to a disk controller, and it becomes a disk drive in the CHARON. If the SCSI disk is a bootable disk you can boot CHARON from it.

A.3. Using a tape

CHARON supports the connection of a SCSI tape drive to a SCSI adapter in your CHARON host system. Assign the tape drive in the CHARON configuration file to access the tape drive by the operating system running on CHARON. This way you can boot from standalone tape to restore your system backup.

Appendix B. Configuration file examples

B.1. Virtual HP AlphaServer ES40 configuration template. (e.g. *es40.cfg*)

This file contains the basic information to set the parameters for the devices used by CHARON-AXP AlphaServer ES40. Make a copy and edit it to set up the connections to your disks, disks images, tape drives, network adapters, etc.

Note

In the CHARON-AXP installation directory you can find the *as400.cfg*, *as800.cfg*, *as1000.cfg*, *as1000a.cfg*, *as1200.cfg*, *as2000.cfg*, *as2100.cfg*, *as4000.cfg*, *as4100.cfg*, *ds10l.cfg*, *ds15.cfg*, *ds20.cfg*, *ds25.cfg*, *es40.cfg*, *es45.cfg*, *gs80.cfg*, *gs160.cfg*, and *gs320.cfg* files for the particular model installed.

Note

The most common solution for a console is using PuTTY terminal emulator. Note that PuTTY is installed by default by the CHARON-AXP installation procedure.

```

#
# Copyright (C) 1999-2012 STROMASYS
# All rights reserved.
#
# The software contained on this media is proprietary to and embodies
# the confidential technology of STROMASYS. Possession, use, duplication,
# or dissemination of the software and media is authorized only pursuant
# to a valid written license from STROMASYS.
#
#=====
#
# Sample configuration file for AlphaServer ES40 machines.
#-----

set session hw_model = AlphaServer_ES40

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

#set session configuration_name = AlphaServer_ES40

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

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

#=====
#
# Overrides system assigned process's CPU affinity. The session changes
# the process's CPU affinity to the one specified.
#-----

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

#=====
#
# The 'n_of_io_cpus' option overrides number of host CPU cores reserved for
# I/O processing. If omitted the session reserves 33% of available host CPU
# cores for I/O processing. Note that total amount of available host CPU
# cores is determined based on process's CPU affinity.
#-----

#set session n_of_io_cpus=1
#set session n_of_io_cpus=2
#set session n_of_io_cpus=...

#=====
#
# AlphaServer ES40 6/500
#-----

#set ace cpu_architecture = EV6

```

```

#set rom dsrdb[0] = 1816 system_name = "AlphaServer ES40 6/500"
#set rom version[1] = 1.98-4 version[2] = 1.92-5

#=====
#
# AlphaServer ES40 6/667
#
#-----

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

#=====
#
# The 'n_of_cpus' option reduces number of emulated Alpha CPUs in the
# configuration.
#
#-----

#set session n_of_cpus=1
#set session n_of_cpus=2
#set session n_of_cpus=3

#=====
#
# Specify size of RAM from 256MB up to 32768MB (32GB) in 256MB extents.
#
#-----

#set ram size=256
#set ram size=512
#set ram size=1024
#set ram size=4096
#set ram size=32768

#=====
#
# Uncomment to allow the SRM console environment be preserved across
# emulator restarts.
#
#-----

#set rom container="clipper.bin"

#=====
#
# Uncomment to allow saving CMOS NVRAM content, so that to preserve
# Time & Date information.
#
#-----

#set toy container="clipper.dat"

#=====
#
# Select connection for the console serial line OPA0.
#
#-----

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

#=====
#

```

```

# Improve granularity of emulated AXP timer.
#
#-----

#set isa clock_period=1000

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

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

#=====
#
# Load optional DE500BA PCI Ethernet Adapter (EWA).
#
#-----

#load DE500BA/dec21x4x EWA interface=EWA0
#load packet_port/chnetwrk EWA0 interface="(disabled)"
#load packet_port/chnetwrk EWA0 interface="connection:<connection-name>"

#=====
#
# Load another optional DE500BA PCI Ethernet Adapter (EWB).
#
#-----

#load DE500BA/dec21x4x EWB interface=EWB0
#load packet_port/chnetwrk EWB0 interface="(disabled)"
#load packet_port/chnetwrk EWB0 interface="connection:<connection-name>"

#=====
#
# Load another optional DE500BA PCI Ethernet Adapter (EWC).
#
#-----

#load DE500BA/dec21x4x EWC interface=EWC0
#load packet_port/chnetwrk EWC0 interface="(disabled)"
#load packet_port/chnetwrk EWC0 interface="connection:<connection-name>"

#=====
#
# Uncomment to enable emulation of DEC-KZPBA SCSI controller.
#
#-----

#load KZPBA PKA scsi_id = 7

#=====
#
# Uncomment to connect the emulator's DKA0 to the disk image.
#
#-----

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

#=====
#
# Uncomment to connect the emulator's DKA100 to host's disk drive.
#
#-----

#set PKA container[100]= "\\.\PhysicalDrive0"

```

```

#set PKA container[100]="\\.\PhysicalDrive<N>"

#=====
#
# Uncomment to connect the emulator's GKA200 to an unknown SCSI device.
#
#-----

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

#=====
#
# Uncomment to connect the emulator's DKA300 to host's CD/DVD-ROM drive.
#
#-----

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

#=====
#
# Uncomment to connect the emulator's DKA400 to .ISO file (CD/DVD-ROM image).
#
#-----

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

#=====
#
# Uncomment to connect the emulator's MKA500 to host's SCSI tape drive.
#
#-----

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

#=====
#
# Uncomment to connect the emulator's MKA600 to .VTAPE file (tape image).
#
#-----

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

#=====
#
# Uncomment to connect the emulator's DKA600 to host's 3.5" FDD. Uncomment
# the next line for the FDD to appear as DEC RX23 (otherwise it appears as
# DEC RX26).
#
# Either "A:" or "B:" may be used in container specification.
#
#-----

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

#=====
#
# Uncomment to enable emulation of DEC-KGPSA-CA PCI FC Adapter.
#
#-----

#load KGPSA FGA

#=====
#

```



```
# Uncomment to connect the emulator's $1$DGA0 to the disk image.
#
#-----

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

#=====
#
# Uncomment to connect the emulator's $1$DGA100 to host's disk drive.
#
#-----

#set FGA container[100] = "\\.\PhysicalDrive<N>"

#=====
#
# Uncomment to enable emulation of secondary DEC-KGPSA-CA PCI FC Adapter.
#
#-----

#load KGPSA FGB

#=====
#
# Uncomment to enable PCI Pass Through access to physical EMULEX LP FC HBA,
# use two adapters to provide multipath with failover.
#
#-----

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

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

Appendix C. Specification of "system_name" parameter

It is important to have the "system_name", "hw_model", "cpu_architecture" and "dsrdb[0]" (DSRB - Dynamic System Recognition Data Block) parameters in sync. The following table illustrates how to synchronize those values:

system_name (rom)	cpu_architecture (ace)	dsrdb[0] (rom)
set session hw_model = AlphaServer_400		
AlphaStation 200 4/100	EV4	1156
AlphaStation 200 4/133	EV4	1088
AlphaStation 205 4/133	EV4	1250
AlphaStation 255 4/133	EV4	1257
AlphaStation 200 4/166	EV4	1087
AlphaStation 205 4/166	EV4	1251
AlphaStation 255 4/166	EV4	1258
AlphaStation 400 4/166	EV4	1086
AlphaStation 205 4/200	EV4	1252
AlphaStation 255 4/200	EV4	1259
AlphaStation 200 4/233	EV45	1151
AlphaStation 205 4/233	EV45	1253
AlphaStation 255 4/233	EV45	1260
AlphaStation 400 4/233	EV45	1152
AlphaStation 205 4/266	EV45	1254
AlphaStation 255 4/266	EV45	1261
AlphaServer 300 4/266	EV45	1593
AlphaStation 400 4/266	EV45	1153
AlphaStation 400 4/266	EV45	1154
AlphaStation 200 4/300	EV45	1157
AlphaStation 205 4/300	EV45	1255
AlphaStation 255 4/300	EV45	1262
AlphaStation 400 4/300	EV45	1160
AlphaStation 205 4/333	EV45	1256
AlphaStation 255 4/333	EV45	1263
set session hw_model = AlphaServer_800		
AlphaServer 600 5/333	EV56	1310
AlphaServer 800 5/333	EV56	1310
AlphaServer 800 5/400	EV56	1584
AlphaStation 600A 5/500	EV56	1590
AlphaServer 800 5/500	EV56	1585
set session hw_model = AlphaServer_1000		
AlphaServer 1000 4/200	EV4	1090

Specification of "system_name"
parameter

system_name (rom)	cpu_architecture (ace)	dsrdb[0] (rom)
AlphaServer 1000 4/233	EV45	1091
AlphaServer 1000 4/266	EV45	1264
AlphaServer 1000 5/300	EV5	1269
AlphaServer 1000 5/333	EV5*	1559
AlphaServer 1000 5/400	EV56*	1312
AlphaServer 1000 5/500	EV56*	1582
AlphaServer 1000 5/500	EV56*	1583
set session hw_model = AlphaServer_1000A		
AlphaServer 1000A 4/266	EV45	1265
AlphaServer 1000A 5/300	EV5	1270
AlphaServer 1000A 5/333	EV5	1558
AlphaServer 1000A 5/400	EV56	1311
AlphaServer 1000A 5/500	EV56	1580
AlphaServer 1000A 5/500	EV56	1581
set session hw_model = AlphaServer_1200		
AlphaServer 1200 5/300	EV5	1722
AlphaServer 1200 5/300	EV5	1724
AlphaServer 1200 5/400	EV56	1726
AlphaServer 1200 5/400	EV56	1728
AlphaStation 1200 5/400	EV56	1758
AlphaStation 1200 5/400	EV56	1760
AlphaServer 1200 5/466	EV56	1730
AlphaServer 1200 5/466	EV56	1732
AlphaStation 1200 5/466	EV56	1762
AlphaStation 1200 5/466	EV56	1764
AlphaServer 1200 5/533	EV56	1734
AlphaServer 1200 5/533	EV56	1736
AlphaServer 1200 5/533	EV56	1746
AlphaServer 1200 5/533	EV56	1748
AlphaStation 1200 5/533	EV56	1766
AlphaStation 1200 5/533	EV56	1768
AlphaStation 1200 5/533	EV56	1778
AlphaStation 1200 5/533	EV56	1780
AlphaServer 1200 5/600	EV56	1738
AlphaServer 1200 5/600	EV56	1740
AlphaServer 1200 5/600	EV56	1750
AlphaServer 1200 5/600	EV56	1752
AlphaStation 1200 5/600	EV56	1770
AlphaStation 1200 5/600	EV56	1772
AlphaStation 1200 5/600	EV56	1782
AlphaStation 1200 5/600	EV56	1784



Specification of "system_name"
parameter

system_name (rom)	cpu_architecture (ace)	dsrdb[0] (rom)
AlphaServer 1200 5/666	EV56	1742
AlphaServer 1200 5/666	EV56	1744
AlphaServer 1200 5/666	EV56	1754
AlphaServer 1200 5/666	EV56	1756
AlphaStation 1200 5/666	EV56	1774
AlphaStation 1200 5/666	EV56	1776
AlphaStation 1200 5/666	EV56	1786
AlphaStation 1200 5/666	EV56	1788
set session hw_model = AlphaServer_2000		
AlphaServer 2000 4/200	EV4	1123
AlphaServer 2000 4/233	EV45	1171
AlphaServer 2000 4/275	EV45	1127
AlphaServer 2000 5/250	EV5	1131
AlphaServer 2000 5/300	EV5	1175
AlphaServer 2000 5/375	EV56	1505
AlphaServer 2000 5/400	EV56	1517
set session hw_model = AlphaServer_2100		
AlphaServer 2100 4/200	EV4	1059
AlphaServer 2100 4/200	EV4	1135
AlphaServer 2100 4/233	EV45	1179
AlphaServer 2100 4/233	EV45	1187
AlphaServer 2100 4/275	EV45	1115
AlphaServer 2100 4/275	EV45	1139
AlphaServer 2100 5/250	EV5	1119
AlphaServer 2100 5/250	EV5	1143
AlphaServer 2100 5/300	EV5	1183
AlphaServer 2100 5/300	EV5	1191
AlphaServer 2100 5/375	EV56	1509
AlphaServer 2100 5/375	EV56	1513
AlphaServer 2100 5/400	EV56	1521
AlphaServer 2100 5/400	EV56	1525
set session hw_model = AlphaServer_4000		
AlphaServer 4000 5/266	EV5	1409
AlphaServer 4000 5/266	EV5	1411
AlphaServer 4000 5/266	EV5	1421
AlphaServer 4000 5/266	EV5	1423
AlphaServer 4000 5/266	EV5	1433
AlphaServer 4000 5/266	EV5	1435
AlphaServer 4000 5/266	EV5	1445
AlphaServer 4000 5/266	EV5	1447
AlphaServer 4000 5/300	EV5	1413

Specification of "system_name"
parameter

system_name (rom)	cpu_architecture (ace)	dsrdb[0] (rom)
AlphaServer 4000 5/300	EV5	1415
AlphaServer 4000 5/300	EV5	1425
AlphaServer 4000 5/300	EV5	1427
AlphaServer 4000 5/300	EV5	1437
AlphaServer 4000 5/300	EV5	1439
AlphaServer 4000 5/300	EV5	1449
AlphaServer 4000 5/300	EV5	1451
AlphaServer 4000 5/400	EV56	1417
AlphaServer 4000 5/400	EV56	1419
AlphaServer 4000 5/400	EV56	1429
AlphaServer 4000 5/400	EV56	1431
AlphaServer 4000 5/400	EV56	1441
AlphaServer 4000 5/400	EV56	1443
AlphaServer 4000 5/400	EV56	1453
AlphaServer 4000 5/400	EV56	1455
AlphaServer 4000 5/466	EV56	1634
AlphaServer 4000 5/466	EV56	1636
AlphaServer 4000 5/466	EV56	1654
AlphaServer 4000 5/466	EV56	1656
AlphaServer 4000 5/533	EV56	1638
AlphaServer 4000 5/533	EV56	1640
AlphaServer 4000 5/533	EV56	1642
AlphaServer 4000 5/533	EV56	1644
AlphaServer 4000 5/533	EV56	1658
AlphaServer 4000 5/533	EV56	1660
AlphaServer 4000 5/533	EV56	1662
AlphaServer 4000 5/533	EV56	1664
AlphaServer 4000 5/600	EV56	1646
AlphaServer 4000 5/600	EV56	1648
AlphaServer 4000 5/600	EV56	1666
AlphaServer 4000 5/600	EV56	1668
AlphaServer 4000 5/666	EV56	1650
AlphaServer 4000 5/666	EV56	1652
AlphaServer 4000 5/666	EV56	1670
AlphaServer 4000 5/666	EV56	1672
set session hw_model = AlphaServer_4100		
AlphaServer 4100 5/266	EV5	1313
AlphaServer 4100 5/266	EV5	1317
AlphaServer 4100 5/266	EV5	1337
AlphaServer 4100 5/266	EV5	1341
AlphaServer 4100 5/266	EV5	1361



Specification of "system_name"
parameter

system_name (rom)	cpu_architecture (ace)	dsrdb[0] (rom)
AlphaServer 4100 5/266	EV5	1365
AlphaServer 4100 5/266	EV5	1385
AlphaServer 4100 5/266	EV5	1389
AlphaServer 4100 5/300	EV5	1321
AlphaServer 4100 5/300	EV5	1325
AlphaServer 4100 5/300	EV5	1345
AlphaServer 4100 5/300	EV5	1349
AlphaServer 4100 5/300	EV5	1369
AlphaServer 4100 5/300	EV5	1373
AlphaServer 4100 5/300	EV5	1393
AlphaServer 4100 5/300	EV5	1397
AlphaServer 4100 5/400	EV56	1329
AlphaServer 4100 5/400	EV56	1333
AlphaServer 4100 5/400	EV56	1353
AlphaServer 4100 5/400	EV56	1357
AlphaServer 4100 5/400	EV56	1377
AlphaServer 4100 5/400	EV56	1381
AlphaServer 4100 5/400	EV56	1401
AlphaServer 4100 5/400	EV56	1405
AlphaServer 4100 5/466	EV56	1594
AlphaServer 4100 5/466	EV56	1598
AlphaServer 4100 5/533	EV56	1602
AlphaServer 4100 5/533	EV56	1606
AlphaServer 4100 5/533	EV56	1610
AlphaServer 4100 5/533	EV56	1614
AlphaServer 4100 5/600	EV56	1618
AlphaServer 4100 5/600	EV56	1622
AlphaServer 4100 5/666	EV56	1626
AlphaServer 4100 5/666	EV56	1630
set session hw_model = AlphaServer_DS10L		
AlphaServer DS10 6/466	EV6	1839
AlphaStation DS10 6/466	EV6	1879
AlphaStation XP900 6/466	EV6	1879
AlphaServer DS10L 6/466	EV6	1961
AlphaServer DS10L 67/616	EV67	1962
AlphaStation DS10 67/616	EV67	1962
AlphaServer DS10 67/616	EV67	1970
set session hw_model = AlphaServer_DS15		
AlphaServer DS15 68CB/1000	EV68	2047
AlphaStation DS15 68CB/1000	EV68	2048
AlphaServer TS15 68CB/1000	EV68	2049

Specification of "system_name"
parameter

system_name (rom)	cpu_architecture (ace)	dsrdb[0] (rom)
set session hw_model = AlphaServer_DS20		
AlphaServer DS20 6/500	EV6	1838
AlphaServer DS20E 6/500	EV6	1840
AlphaServer DS20 6/500	EV6	1920
AlphaServer DS20 6/500	EV6	1921
AlphaServer DS20E 67/667	EV67	1939
AlphaStation DS20E 6/500	EV6	1941
AlphaStation DS20E 67/667	EV67	1943
AlphaServer DS20E 68A/833	EV68	1964
AlphaServer DS20E 68A/833	EV68	1982
AlphaServer DS20L 68A/833	EV68	2006
set session hw_model = AlphaServer_DS25		
AlphaServer DS25 68CB/1000	EV68	1994
AlphaStation DS25 68CB/1000	EV68	1995
set session hw_model = AlphaServer_ES40		
AlphaServer ES40 6/500	EV6	1813
AlphaServer ES40 6/500	EV6	1861
AlphaServer ES40 6/500	EV6	1869
AlphaServer ES40 6/500	EV6	1923
AlphaServer ES40 6/500	EV6	1931
AlphaServer ES40 6/667	EV6	1817
AlphaServer ES40 6/667	EV6	1865
AlphaServer ES40 6/667	EV6	1873
AlphaServer ES40 6/667	EV6	1927
AlphaServer ES40 6/667	EV6	1935
AlphaStation ES40 67/667	EV67	1949
AlphaStation ES40 67/667	EV67	1957
AlphaStation ES40 68/833	EV68	1984
AlphaStation ES40 68/833	EV68	1988
set session hw_model = AlphaServer_ES45		
AlphaServer ES45/3B 68CB/1000	EV68	1971
AlphaServer ES45/2 68CB/1000	EV68	1975
AlphaServer ES45/2B 68CB/1000	EV68	1975
AlphaServer ES45/1B 68CB/1000	EV68	2002
AlphaServer ES45/3B 68CB/1250	EV68	2013
AlphaServer ES45/2 68CB/1250	EV68	2017

Specification of "system_name"
parameter

system_name (rom)	cpu_architecture (ace)	dsrdb[0] (rom)
AlphaServer ES45/2B 68CB/1250	EV68	2017
AlphaServer ES45/1B 68CB/1250	EV68	2021
set session hw_model = AlphaServer_GS80		
AlphaServer GS80 67/728	EV67	1967
AlphaServer GS1280	EV67	2038
set session hw_model = AlphaServer_GS160		
AlphaServer GS160 67/728	EV67	1968
AlphaServer GS1280	EV67	2039
set session hw_model = AlphaServer_GS320		
AlphaServer GS320 67/728	EV67	1969
AlphaServer GS1280	EV67	2040

AlphaServer GS1280, **AlphaServer GS1280** and **AlphaServer GS1280** also require the parameters **dsrdb[1]** and **dsrdb[2]** to be set in the following way:

- **AlphaServer GS1280: dsrdb[1]=50 dsrdb[2]=3050**

Appendix D. Required Windows Standard Services for CHARON

Stromasys SA recommends the use of the host operating system that is specified in the Software Product Description of the CHARON product (see www.stromasys.com [http://www.stromasys.com]). Stromasys SA also recommends that users do not change the default settings of the Windows operating system, with the exception of the functions listed in the “**In all cases**” and “**Optionally**” sections below. For these recommendations to be effective it is assumed that CHARON is the only user application running on the host system. Simultaneous use of other applications on the same Windows host system, with the exception of a terminal emulator, is outside the design specification of CHARON products.

Additional Windows services may be disabled (“**May be disabled**” section) beyond those listed in the “**In all cases**” section below. Note that disabling these additional services will in general not lead to a significant performance improvement but can meet a customer’s requirement for stability and risk reduction by disabling unused functionality.

The “**May be disabled optionally**” section describes the services that can be disabled if its functionality is not required in a particular host environment.

In addition Stromasys SA has carried out tests to list services that should **not** be disabled, as they are important or critical to the operation of Windows or CHARON.

The conclusions were drawn from the following test environment.

The system software versions (fresh installations) used in the testing:

- Windows 7 x64 and x86, Service Pack 1
- Windows 2008 R2 x64, Service Pack 2

The Windows installations accepted the recommended configuration options.

CHARON was tested for: installation/de-installation, network driver installation/de-installation, general disk management, working with CDROM, networking using DECnet and TCP/IP.

D.1. Services disabling

D.1.1. In all cases

All CHARON users must review the following tasks and services on CHARON host systems to ensure their use is in line with the users CHARON usage requirements. Stromasys SA recommends disabling these tasks and services:

- **Automatic Updates** – may cause network and processor traffic that could disrupt CHARON
- **Screensaver and Power saving features** – could reduce processing resources required by CHARON.
- **Any virus protection** – can cause disruption by accessing CHARON files. See “*Application Note 29*” for Recommendations Regarding Security of CHARON Host Platforms.
- **Windows Messenger** - may cause network and processor traffic that could disrupt CHARON

- **Indexing Service** – may cause burst mode activity that may destabilize a CHARON CPU. This is especially important in SMP systems where the CPU performance must be consistent across CPU's. Examples of CHARON SMP systems are CHARON-VAX/66x0 for Windows and CHARON-AXP for Windows.

D.1.2. Windows services that may be disabled

Service Name	Description
AxInstSV	ActiveX installer
bthserv	Bluetooth Support Service
CertPropSvc	Certificate Propagation
Fax	Fax Support
KtmRm	KtmRm for Distributed Transaction Coordinator
swprv	Microsoft Software Shadow Copy Provider
MMCSS	Multimedia Class Scheduler
WPCSvc	Parental Controls
wercplsupport	Problem Reports and Solutions Control Panel Support
QWAVE	Quality Windows Audio Video Experience
RpcLocator	Remote Procedure Call (RPC) Locator
TabletInputService	Tablet PC Input Service
TapiSrv	Telephony
AudioEndpointBuilder	Windows Audio Endpoint Builder
WinDefend	Windows Defender
ehRecvr	Windows Media Center Receiver Service
ehSched	Windows Media Center Scheduler Service
WMPNetworkSvc	Windows Media Player Network Sharing Service
wuauerv	Windows Update
AeLookupSvc	Application Experience
AppIDSvc	Application Identity
Appinfo	Application Information
BDESVC	BitLocker Drive Encryption Service
PeerDistSvc	BranchCache
RpcSs	Remote Procedure Call (RPC)
WPDBusEnum	Portable Device Enumerator Service
Schedule	Task Scheduler
WSearch	Windows Search
WbioSrv	Windows Biometric Service
SCardSvr	Smart Card
SCPolicySvc	Smart Card Removal Policy
DPS	Diagnostic Policy Service
TrkWks	Distributed Link Tracking Client
iphlpvc	IP Helper
CscService	Offline Files

Service Name	Description
seclogon	Secondary Logon
lmhosts	TCP/IP NetBIOS Helper
HomeGroupListener	HomeGroup Listener
HomeGroupProvider	HomeGroup Provider

D.1.3. Windows services that may be disabled optionally

Service Name	Description
SensrSvc	Adaptive Brightness
defragsvc	Disk Defragmenter
Dnscache	DNS Client
fdPHost	Function Discovery Provider Host
SharedAccess	Internet Connection Sharing (ICS)
ProtectedStorage	Protected Storage
WerSvc	Windows Error Reporting Service
MpsSvc	Windows Firewall

D.2. Windows services that must NOT be disabled

Service Name	Description
EventSystem	COM+ Event System
hkmsvc	Health Key and Certificate Management
COMSysApp	COM+ System Application
Dhcp	DHCP Client
PolicyAgent	IPsec Policy Agent
NetTcpPortSharing	Net.Tcp Port Sharing Service
Netlogon	Netlogon
RasAuto	Remote Access Auto Connection Manager
TermService	Remote Desktop Services
LanmanServer	Server
BFE	Base Filtering Engine
wbengine	Block Level Backup Engine Service
KeyIso	CNG Key Isolation
VaultSvc	Credential Manager
CryptSvc	Cryptographic Services
DcomLaunch	DCOM Server Process Launcher
UxSms	Desktop Window Manager Session Manager
WdiServiceHost	Diagnostic Service Host
WdiSystemHost	Diagnostic System Host
MSDTC	Distributed Transaction Coordinator
EFS	Encrypting File System (EFS)

Required Windows Standard Services for CHARON

Service Name	Description
EapHost	Extensible Authentication Protocol
FDResPub	Function Discovery Resource Publication
gpsvc	Group Policy Client
IKEEXT	IKE and AuthIP IPsec Keying Modules
UI0Detect	Interactive Services Detection
lltdsvc	Link-Layer Topology Discovery Mapper
MSiSCSI	Microsoft iSCSI Initiator Service
napagent	Network Access Protection Agent
Netman	Network Connections
netprofm	Network List Service
NlaSvc	Network Location Awareness
nsi	Network Store Interface Service
PNRPsvc	Peer Name Resolution Protocol
p2psvc	Peer Networking Grouping
p2pimsvc	Peer Networking Identity Manager
pla	Performance Logs & Alerts
PlugPlay	Plug and Play
IPBusEnum	PnP-X IP Bus Enumerator
PNRPAutoReg	PNRP Machine Name Publication Service
Power	Power
Spooler	Print Spooler
PcaSvc	Program Compatibility Assistant Service
RasMan	Remote Access Connection Manager
SessionEnv	Remote Desktop Configuration
UmRdpService	Remote Desktop Services UserMode Port Redirector
RpcEptMapper	RPC Endpoint Mapper
SstpSvc	Secure Socket Tunneling Protocol Service
SamSs	Security Accounts Manager
wscsvc	Security Center
SNMPTRAP	SNMP Trap
sppsvc	Software Protection
sppuinotify	SPP Notification Service
SysMain	Superfetch
SENS	System Event Notification Service
THREADORDER	Thread Ordering Server
TBS	TPM Base Services
upnphost	UPnP Device Host
uvnc_service	uvnc_service
vds	Virtual Disk
WatAdminSvc	Windows Activation Technologies Service
SDRSVC	Windows Backup

Service Name	Description
idsvc	Windows CardSpace
WcsPlugInService	Windows Color System
wcncsvc	Windows Connect Now - Config Registrar
wudfsvc	Windows Driver Foundation - User-mode Driver Framework
Weccsvc	Windows Event Collector
eventlog	Windows Event Log
FontCache	Windows Font Cache Service
msiserver	Windows Installer
Winmgmt	Windows Management Instrumentation
TrustedInstaller	Windows Modules Installer
FontCache3.0.0.0	Windows Presentation Foundation Font Cache 3.0.0.0
WinRM	Windows Remote Management (WS-Management)
W32Time	Windows Time
WinHttpAutoProxySvc	WinHTTP Web Proxy Auto-Discovery Service
dot3svc	Wired AutoConfig
Wlansvc	WLAN AutoConfig
wmiApSrv	WMI Performance Adapter
LanmanWorkstation	Workstation
WwanSvc	WWAN AutoConfig

D.3. Exceptions

The above lists are valid for the conditions that were tested. A user may require specific features not covered by these tests. For example when PC host networking should not be used at all (rare case) all of the services relevant to networking may be disabled. The same is true in case of RDP, etc.

Note that “**Telephony**” service is needed for Windows default Hyper Term terminal emulator that also may be utilized for CHARON instead of the default “**Putty**” one.

Note

Changing the status of standard services can be dangerous!