

CHARON™-VAX

Utilities and interconnects



© 2003 - 2006 Software Resources International S.A.

Under the copyright laws, this manual and the software described within may not be copied for profit, in whole or in part, without the written consent of Software Resources International. The same proprietary and copyright notices must be affixed to any permitted copies as were affixed to the original. This exception does not allow copies to be made for others, whether or not sold. Under the law, copying includes translating into another language or format.

CHARON-VAX/ Utilities and interconnects

Manual number: SRI-30-16-015

Version: 27 February 2006

Email: charon@softresint.com

Ownership Notice

Software Resources International SA owns all rights, including proprietary rights, copyrights, trademarks, and world-wide distribution rights to a methodology for the execution of VAX applications and system software by emulating VAX hardware components in software, henceforth referred to as CHARON-VAX.

The license for the CHARON-VAX emulator allows the execution of the CHARON-VAX run-time system on a single host system or cluster. The software transfer utilities described in this manual can be run on any system provided that the resulting code is only used on systems with a valid CHARON-VAX license.

The tools and utilities described in this manual are not covered by the standard CHARON-VAX product support agreements. They can be used at your own risk with a licensed CHARON-VAX emulator. Software Resources International explicitly rejects any responsibility or obligation to provide upgrades of, or support for the use of the tools and utilities described in this document.

Trademarks

The CHARON name with logo is a trademark of Software Resources International. VAX, Qbus, VMS and OpenVMS are trademarks of Compaq Computer Corporation, now HP company. Windows is a registered trademark in the United States and other countries, licensed exclusively through Microsoft Corporation, USA and Intel are registered trademarks in the United States and other countries, licensed exclusively through Intel Corporation, USA. All other trademarks and registered trademarks are the property of their respective holders.

Contents

Preface	4
1 Overview	5
1.1 Functionality limitations.....	6
2 Utilities	7
2.1 The disk utility MKIMAGE.....	7
2.2 The UNDUMP utility	13
2.3 The tape utility MTD.....	14
2.4 The File Transfer Utility VAXFT.....	16
2.5 The SCSI device locator SCSI_CHECK.....	20
2.6 The Files-11 Browser.....	21
2.7 The CHARON-VAX terminal emulator.....	23
3 Software Interconnects	25
3.1 Host connectivity via emulated serial lines.....	25
3.2 The CHARON virtual LAN adapter (MUX driver).....	27
Further information	i

Preface

The CHARON-VAX emulators of Software Resources International are used for the replacement of VAX systems in commercial, military, and educational environments. During the development and evolution of these emulator products, specific tools and solutions were created to resolve particular installation and operational issues. This manual documents some of that software, which is provided purely as a convenience.

The utilities and interconnect software described in this manual were originally developed either for specific project needs, or to assist in CHARON-VAX installation. The software is provided on an as-is basis, but we would appreciate feedback concerning improvements and problems in its use.

The software described in this manual is either included in a CHARON-VAX product installation kit or is available on request to holders of a valid CHARON-VAX license. We hope that these utilities can assist you in installing, using and enriching your CHARON-VAX emulator application.

1 Overview

The following software utilities are described in chapter 2:

- The disk utility MKIMAGE, which can capture the image of a VAX system or user disk (including images of CD's or floppy disks) and convert such image into a (bootable) disk container file as used by the CHARON-VAX emulators.
- The UNDUMP utility which restores an original disk image from an ASCII dump file. This ASCII dump can be obtained via a VAX serial console port, in case no other means of software transfer exist.
- The tape utility MTD, which creates tape container files from tapes and convert CHARON-VAX tape container files to physical tapes.
- VAXFT, which is an experimental file transfer utility to transfer data between CHARON-VAX disk container files and the host file system. VAXFT has an FTP-like command line interface.
- SCSI_CHECK, which is a Windows utility to locate SCSI devices (essentially tapes and disk drives) on the host system bus. SCSI_CHECK generates the appropriate device string for inclusion in a CHARON-VAX/XM/XL configuration file.
- The Files-11 Browser, which is an add-in to the Microsoft Explorer for Windows 2000/XP. It extends the Explorer to view the contents of VMS-formatted (Files-11 ODS Level 2) CHARON-VAX disk image files.
- The CHARON-VAX terminal, which is an (obsolete) integrated VT100 terminal emulator available in some versions of CHARON-VAX/Industrial.

Methods to interconnect emulated CHARON-VAX peripherals to software running in the Windows host system are described in chapter 3:

- Host application connectivity via emulated serial lines.
- The CHARON virtual LAN adapter (MUX driver)

In addition to the interconnect methods above, several CHARON-VAX products have an API (CHAPI) that allows to create new emulated Qbus peripherals. These emulated custom peripherals run as DLL's in the host system together with the standard CHARON-VAX product.

The CHAPI QBUS /VAX manual (30-16-013) describes the CHAPI concept, API and programming examples. A C++ library is available to quickly configure Qbus addresses and interrupt vectors according to the VAX Qbus standards.

1.1 Functionality limitations

The CHARON-VAX emulators are intended to prolong the use of legacy VAX operating systems and applications. The emulators provide software models of all VAX components necessary to create a complete 'virtual' VAX system. Note that the emulated functionality does not include diagnostic modes or time delays that would be required for diagnostic software to correctly verify mechanical device behavior.

As a model of VAX hardware, the CHARON-VAX emulators execute in principle any MicroVAX operating system or binary application. It remains emulation, however, and the characteristics of the host system – e.g. CPU clock rate – influence the real-time behavior of the emulated system. This influence also applies to the use of third party hardware options as described in this document. Interrupt handling and DMA operation is influenced by the host system.

The CHARON-VAX interconnect methods described in this manual have originated out of specific projects. Where appropriate, suggestions about potential problem areas and applicable limitations are given. While much care has been given to obtain satisfactory operation for specific applications, no general guarantees for correct operation can be given. Consequently, this software is provided on an 'as is' basis. Software Resources International cannot accept any responsibility for a particular use or suitability of the contents of this manual. As a general rule, testing with the target application is always required to verify the correct functionality.

2 Utilities

2.1 The disk utility MKIMAGE

Overview

The CHARON-VAX/Industrial emulator uses disk image files on the host system to represent the disks of an emulated VAX hardware platform. The MKIMAGE utility creates these disk images from a physical device like a distribution CD or a disk on a VAX system. MKIMAGE has two functions:

- Collecting the disk data on the VAX system (MKIMAGE in *client mode*).
- Generating a file containing the binary disk image (MKIMAGE in *server mode*).

A CHARON-VAX/Industrial disk image file has approximately the same size as the original VAX disk, independent of the amount of data stored on it. This implies that on VAX systems with a single disk the image file cannot be stored locally; such images must be directly transferred to another system, for instance the host system running CHARON-VAX/Industrial. On a VAX system with multiple disks the image file could be stored locally and later transferred via network services or tape. The network transfer time of a disk image depends on the effective transfer rate of the connection and can be several hours depending on the disk size.

To cover the various platforms of CHARON-VAX/Industrial and to provide the facility to transfer the data over TCP/IP, the MKIMAGE utility is provided in four binary executable versions:

- MKIMAGE-VMS-VAX.EXE, for VAX/VMS systems.
- MKIMAGE-VMS-AXP.EXE, for OpenVMS Alpha systems.
- MKIMAGE-W32-I386.EXE, for Intel compatible Windows systems.
- MKIMAGE-LNX-I386, for Intel compatible Linux systems.

MKIMAGE can directly transfer data through a TCP/IP network connection to a remote system (*client* and *server* part are separate) or generate a disk image on the same system or on a system connected via DECnet (in this case the

instance of MKIMAGE is both *client and server*). A disk image can for example be transferred from a VAX to an OpenVMS Alpha or PC Windows system via TCP/IP, or an image of a VAX/VMS CD-ROM can be generated locally on an Intel or Alpha system.

MKIMAGE is called from the command line and has two arguments: an input stream and an output stream. Invoked without arguments the MKIMAGE utility displays the usage information. The same MKIMAGE utility executable can work as *client*, *server* or both, depending on the parameters specified in the command line. The format is:

MKIMAGE if=<input name> of=<output name>

If MKIMAGE runs on the VAX/VMS system from which you want to obtain a disk image, <input name> must be the logical name of the physical disk to copy. The <output name> specifies in this case the name of the file that will contain the disk image (it is recommended to use a .dsk extension for it), either on the same system or on a remote DECnet node. If <output name> specifies a TCP/IP address (specified as tcpip:host:port) MKIMAGE on the VAX/VMS system switches to *client* mode. You must specify a TCP/IP port number greater than 1024 that is not used by any other applications.

When running MKIMAGE on a CHARON-VAX/Industrial host platform, <input name> can be the device that will be the source for a disk image, for instance a VAX CD-ROM in drive D: of a Windows system. If <input name> specifies the TCP/IP address and the port number where a MKIMAGE *client* is sending data, MKIMAGE switches to *server* mode. You should specify the same port number for the client and server transmissions.

When running MKIMAGE on a CHARON-VAX/Industrial host platform, <output name> is the name and location of the resulting disk image.

In the following examples it is assumed that symbols are defined in OpenVMS that invoke MKIMAGE, for example in VAX/VMS:

```
$ MKIMAGE := "$ USER$DISK:[USERS.CHARON-VAX]MKIMAGE-VMS-VAX.EXE"
```

In OpenVMS/Alpha:

```
$ MKIMAGE := "$ USER$DISK:[USERS.CHARON-VAX]MKIMAGE-VMS-AXP.EXE"
```

In order to transfer data from a VAX system you must have one of the following TCP/IP support packages installed on that VAX/VMS system: UCX, TCPware (from PROCESS Software) or Digital TCPIP Services.

Generating a disk image on a local VAX disk

Assuming that the VAX disk of which an image is to be made is called DKA100: and the result is stored on DKA200: we can use the following sequence:

```
$ MKIMAGE IF=DKA100: OF=DKA200:[000000]MY_DISK.DSK
```

```
(C) Copyright 2000, Software Resources International
```

```
USER$DISK:[USERS.CHARON-VAX]MKIMAGE-VMS-VAX.EXE;1, utility to  
copy data from a physical device to a disk image file
```

```
Total transferred: 681574400 bytes
```

```
$
```

Generating a disk image in Windows with TCP/IP

To create a TCP/IP connection with the MKIMAGE utility between VAX/VMS and PC/Windows, *first start the server part on the PC* (MKIMAGE-W32-I386.EXE) with the proper parameters, and then the *client* part on VAX. Assuming that the VAX disk is called DKA100: we can use the following sequence:

On the Windows system:

```
MYNOTE>mkimage-w32-i386 if=tcpip:mynote.mydomain:6000  
of=my_disk.dsk
```

```
(C) Copyright 2000, Software Resources International
```

```
Utility to copy data from a physical device to a disk image file
```

```
Total transferred: 681574400 bytes
```

```
MYNOTE>
```

On the VMS system:

```
$ MKIMAGE IF=DKA100: OF=TCPIP:MYNOTE.MYDOMAIN:6000
```

```
(C) Copyright 2000, Software Resources International
```

```
USER$DISK:[USERS.CHARON-VAX]MKIMAGE-VMS-VAX.EXE;1, utility to  
copy data from a physical device to a disk image file
```

```
Total transferred: 681574400 bytes
```

```
$
```

Generating a disk image on OpenVMS/Alpha & TCP/IP

On the OpenVMS/Alpha system start MKIMAGE in server mode, for example:

```
$ MKIMAGE if=tcpip:192.168.1.2:5000 of=mydisk.dsk
```

NOTE: Since in server mode MKIMAGE is using its local TCP/IP address as part of the input parameter, the host component of the TCP/IP address can be omitted, i.e.

\$ MKIMAGE if=tcpip:5000 of=mydisk.dsk

On the VAX/VMS system start MKIMAGE in client mode, for example:

\$ MKIMAGE if=DKA100: of=tcpip:192.168.1.2:5000

In the above example the *server*, an Alpha system running OpenVMS, creates a CHARON-VAX container file by listening for data on port 5000 of TCP/IP address 192.168.1.2 for a file from the MKIMAGE *client*, which writes the disk DKA100: to the specified port.

Copying to a remote DECnet node

Assuming that the VAX disk is called DKA100: we can use the following sequence:

\$ MKIMAGE IF=DKA100: OF=MYNODE::MY_DISK.DSK

(C) Copyright 2000, Software Resources International

USER\$DISK:[USERS.CHARON-VAX]MKIMAGE-VMS-VAX.EXE;1, utility to copy data from a physical device to a disk image file

Total transferred: 681574400 bytes

\$

Generating a VMS CD-ROM image in Windows

Assuming that the CD-ROM drive is called D: we can use the following sequence:

MYNOTE> mkimage-w32-i386 if=\\.\D: of=cdrom.dsk

(C) Copyright 2000, Software Resources International

mkimage-w32-i386, utility to copy data from a physical device to a disk image file

Total transferred: 681574400 bytes

MYNOTE>

The MKIMAGE utility was tested on:

OpenVMS/VAX 6.2 with UCX 3.3

OpenVMS/VAX 6.2 with UCX 4.1

OpenVMS/VAX 6.2 with TCPWARE 5.4

OpenVMS/VAX 7.2 with UCX 4.1

OpenVMS/VAX 7.2 with TCPWARE 5.4

The MKIMAGE utility was tested on:

OpenVMS/VAX 7.2 with DIGITAL TCPIP SERVICES 5.0
OpenVMS/Alpha 7.1 with DIGITAL TCPIP SERVICES 5.0
OpenVMS/Alpha 7.2 with DIGITAL TCPIP SERVICES 5.0
VAX /VMS 5.5 with UCX 4.1
VAX/VMS 5.5 with TCPWARE 5.4
Windows NT 4.0
Windows 2000
Windows 95
Windows 98
RedHat Linux 6.0 kernel 2.2.5-15

Note: Reading of physical media (Floppy disk or CD-ROM) is not supported on Windows 95 and 98.

MKIMAGE MACRO32 source code and usage details

The source code of MKIMAGE is available, written in MACRO32 assembler (to avoid VMS version dependencies) and is used as follows on the target VMS system:

1. Copy the MKIMAGE MACRO32 build package (mkimage.mar, available on the download page of www.vaxemulator.com) to the VMS system. This can be done for example via the network, via a tape or via a serial line using for instance KERMIT.
2. Generate the MKIMAGE executable from the MKIMAGE sources. On the VMS system issue the commands:

```
set def <directory where you copied mkimage.mar>
```

```
macro mkimage.mar
```

```
link mkimage.obj
```

For example:

```
SET DEF DKA0:[000000.MKIMAGE]
```

```
MACRO MKIMAGE.MAR
```

```
LINK MKIMAGE.OBJ
```

When generating CHARON-VAX disk images from your VAX disks, MKIMAGE can store them on another local disk of your VAX system or on a disk of a

remote VMS node (or any TCP/IP network node in case you have TCP/IP Services on your VAX system).

MKIMAGE has no command line parameters, it will interactively ask for the required parameters. Start MKIMAGE on the VMS system console, for example:

\$ RUN MKIMAGE

INPUT NAME OF SOURCE DEVICE (DEV:): DUA2:

INPUT DESTINATION FILE NAME (DEV:<PATH>): DUA1:[000000]D.DSK

or

\$ RUN MKIMAGE

INPUT NAME OF SOURCE DEVICE (DEV:): DUA2:

INPUT DESTINATION FILE NAME (DEV:<PATH>):

MYVMS::[SPACE]D.DSK

where MYVMS is the DECnet name of your other VMS system and D.DSK is the name of the resulting file. This example applies when on the MYVMS node the FAL service is enabled.

Otherwise, the ***system"login password"::disk:[path]*** notation can be used. Note that the resulting disk image file will have the same size as the disk from which you have created it, so you must have enough free space in the destination location to store the data. The MKIMAGE utility indicates the copying process by showing on the screen the number of blocks that have been processed (in hundreds of blocks).

IMPORTANT: Copying disks over the network from VAX systems is a slow process and might take up to several days for big disks. For example, on a MicroVAX3800 and on a VAX4000-300 the measured copying rates for DSSI disks were about 50 MB/hour and 100 MB/hour respectively.

When MKIMAGE has finished the creation of a disk image file, it must be copied to the host system on which the CHARON-VAX/Industrial emulator is running. The disk image file can then be assigned to a CHARON-VAX/Industrial emulated disk controller in the configuration file:

set DUA file[0]="C:\WinVAX\d.dsk"

2.2 The UNDUMP utility

The UNDUMP utility restores a VAX (system) disk image from an ASCII dump file. To do this, the VAX system is connected to the CHARON-VAX/Industrial host via a serial line (e.g. console port). At the VAX side, run a dump utility to make an ASCII dump of the MicroVAX disk and redirect its output to the serial line connected to the host. At the host side, run a utility that downloads data received from a serial line into a file (e.g. Kermit).

After data transfer is completed (which can take a long time via serial line); the receiving system has a file with the disk ASCII dump. The UNDUMP utility can convert this dump file into a CHARON-VAX/Industrial container file with an exact image of the MicroVAX disk.

When UNDUMP processes a dump file, it converts ASCII lines into sequences of bytes. In case of problems (e.g. a corrupted file), any not-converted line (except blank lines or lines with logical sector information) is written into a report file named "undump.lst".

Usage of this utility is as follows:

undump -d <disk_type> -m <dump_type> source destination

If the destination file already exists, the utility overwrites it.

Note: UNDUMP without parameters prints a list of all supported disk models.

For destination files restored from VMS dump files, UNDUMP overwrites only duplicated sectors (the physical sector number computed from a logical sector number presented inside the dump file). The utility reads the dump file until the number of bytes required for the selected physical drive is converted.

The **UNDUMP** utility is available for Windows, OpenVMS/Alpha, Linux and Compaq Tru64UNIX.

2.3 The tape utility MTD

MTD is an exchange utility between CHARON-VAX/Industrial tape container files and magnetic tapes. It copies the data from a magnetic tape into a file that has a special internal structure to preserve tape record size information and vice versa. This is a way to transfer VAX/VMS backup tapes directly to a CHARON-VAX/Industrial system.

Note: the CHARON-VAX/XM/XL (plus) family of products does not use tape container files.

The utility is used in the following way:

MTD <tape-drive> <container-file>

To prepare a tape container file, or

MTD <container-file> <tape-drive>

To copy a tape container file to a physical tape. <tape-drive> stands for an existing tape device available for read/write operation (i.e. loaded and properly mounted) and <container-file> stands for the name of the file that must be created.

Note that the Windows operating system handles only those tape drives which have been properly installed and with started drivers. All modern tape drives have such drivers, but some old devices (especially SCSI-I tape drives) do not have drivers for the Windows operating system. Since such older tape drives are common on MicroVAX systems, the MTD utility can work without Windows drivers; it implements the SCSI protocol and commands directly the SCSI Port driver.

If the tape device you use has an installed driver, you address it in MTD with its Windows device name:

\\.\tapeN

Where N is a unique number assigned by its loaded driver. If the tape device has no driver installed, MTD can address the SCSI tape drive directly as follows:

\\.\SCSI<scsi-port>:<scsi-bus>:<scsi-id>

Where:

<scsi-port> stands for the number assigned by the Windows operating system to the controller to which the tape drive is connected,

<*scsi-bus*> stands for the SCSI bus path ID to which the tape drive is connected (usually 0),

and <*scsi-id*> stands for the SCSI ID of the tape drive.

In Windows 2000 and Windows XP the <*scsi-port*> number must be obtained from the registry “HKEY_Local_Machine \ Hardware \ Devicemap \ Scsi” key.

The other two parameters can be obtained in Windows 2000 and Windows XP from the Control Panel: “System / Hardware / Device Manager/ Other devices / Scsi Sequential Device”. You should choose the properties of the SCSI device that represent your tape device and note the settings. If you just connected the SCSI tape device, select “Scan for hardware changes” in the Device Manager first.

Below is an example of how the MTD utility can be invoked in Windows when no driver is available for a tape drive:

```
CHARON-VAX> mtd \\.\SCSI2:0:5 mydump.mtd
```

This example creates a tape dump (in *mydump.mtd*) of the tape loaded into the drive with SCSI ID 5 connected to the 0th bus of the third (2) SCSI controller. Often the SCSI port 0 and 1 are occupied by IDE controllers.

If the driver for the tape drive is installed and started, then the appropriate device name must be used:

```
CHARON-VAX> mtd \\.\tape0 mydump.mtd
```

2.4 The File Transfer Utility VAXFT

Overview

The VAXFT utility transfers data between the 'inside' of the CHARON-VAX/Industrial emulator and the host file system, with an FTP-like command line interface.

Files can be transferred either in binary or in text mode. The desired transfer mode should be specified by bin or text (ASCII) commands. The binary mode provides a possibility to transfer raw (binary) files. In this mode, the file is copied 'as is', without any processing with fixed record length in one block.

The text mode is intended for transferring text files. A text file has variable length record size type and the CR/LF carriage control attribute. The format of a text file on the local file system varies from platform to platform and even within one platform, but it is always a standard ASCII file. All file names that are used as command parameters should correspond to the host operating system naming conventions. If it is necessary to transfer a file that does not fit in this category (e.g., has a longer name), you should rename it.

This current version of VAXFT does not set the VMS file attributes correctly when a transfer into a CHARON-VAX disk image is made. The solution is to transfer files in ZIP format as the UNZIP process restores the attributes correctly.

Copying from a PC disk to a VMS volume using VAXFT

- Start CHARON-VAX/Industrial and boot your VMS system disk.
- Create a top-level transfer directory under VMS, such as "\$CREATE/DIRECTORY SYS\$SYSDEVICE:[DEMO]" This operation has to be done only once if you are going to use the same directory for further file transfers.
- Shutdown VMS and stop the CHARON-VAX execution.
- Use ZIP utility on the Windows system to create a zip file that contains the files you want to move to VMS. These files can be in text or binary format.
- Using VAXFT, mount your CHARON-VAX system disk and move the ZIP file into it:

```
E:\Program Files\WinVAX>vaxft
```

```

$ open vms.dsk <<<--- mount the VMS system disk
Volume 'vms.dsk' connected for read/write access.
$ show_def <<<--- see the current directory
DKA100:[000000]
$ cd demo <<<--- switch to the transfer directory
<<<--- [DEMO] has to be pre-created underVMS!
Working directory is [000000.DEMO]
$ bin <<<--- Make sure the data transfer is binary
Type set to I.
$ put test.zip <<<--- moves the file from PC to the VMS disk
Copying 'test.zip' - 193 bytes to DKA100:TEST.ZIP ...
Transfer complete.
$ dir test.zip <<<--- verify the file is there
Directory DKA100:[000000.DEMO]
TEST.ZIP;1
Total of 1 files.
$ exit <<<---- exit VAXFT
Goodbye

```

- Start CHARON-VAX/Industrial and boot your VMS system disk.
- Log into your VMS system running on CHARON-VAX/Industrial.
- Switch to the [DEMO] directory and use a VMS version of the UNZIP utility (taken from the DECUS freeware disk for example) to unpack the zip file:

```

$ set def sys$sysdevice:[demo]
$ dir
Directory SYS$SYSDEVICE:[DEMO]
TEST.ZIP;1 1 <None specified> (RWE,RWE,RE,E)
Total of 1 file, 1 block.

```

Check the content of the test.zip file:

```

$ unzip -v test.zip <<<- This is the freeware UNZIP program
Archive:SYS$SYSDEVICE:[DEMO]TEST.ZIP;1

```

Length	Method	Size	Ratio	Date	Time	CRC-32	Name
87	Defl:N	79	9%	08-21-00	00:13	7e31cb05	TEST.TXT
87		79	9%				1

And now extract the file(s) from test.zip:

\$ unzip test.zip

Archive:SYS\$SYSDEVICE:[DEMO]TEST.ZIP;1

inflating: TEST.TXT

Verify that the data is correct:

\$ type test.txt

This is a simple text file --

it could be a VMS command (.com) file

or anything else.

\$

Copying files from a CHARON-VAX/Industrial system disk to a PC

1. Make sure that CHARON-VAX/INDUSTRIAL is not up. Start VAXFT:

E:\Program Files\WinVAX>vaxft

2. Open a VMS disk you want to work with and list directories on it:

\$ open vms.dsk

Volume 'vms.dsk' connected for read/write access.

\$ dir

Directory DKA100:[000000]

000000.DIR;1

BACKUP.SYS;1

BADBLK.SYS;1

BADLOG.SYS;1

BITMAP.SYS;1

CONTIN.SYS;1

CORIMG.SYS;1

DHRYSTONE.DIR;1

INDEXF.SYS;1

PETE.DIR;1

SECURITY.SYS;1

SYS0.DIR;1

SYSEXE.DIR;1

UTILITIES.DIR;1

VMS\$COMMON.DIR;1

VOLSET.SYS;1

Total of 16 files.

3. The file we are going to copy into a CHARON-VAX/Industrial disk image is placed in the VMS directory [pete].

\$ set def [.PETE]

Working directory is [000000.PETE]

\$ dir

Directory DKA100:[000000.PETE]

VMS.PAK;1

VUP-O-METER.COM;1

Total of 2 files.

4. The file we are going to copy to the Windows host system is an ASCII file (it's the command file vup-o-meter.com).

5. Make sure that the ASCII transfer mode is set:

\$ ASCII

Type set to A.

6. Copy the file:

\$ get vup-o-meter.com

Copying DKA100:VUP-O-METER.COM to 'vup-o-meter.com' ...

Transfer complete.

7. Disconnect the disk image and quit from VAXFT:

\$ close

Volume 'vms.dsk' disconnected.

\$ quit

Goodbye.

8. Now verify that the file is in my host system file system::

E:\Program Files\WinVAX>type vup-o-meter.com

\$!VUPOMETER.COM - super simple VUP meter

\$ time_start = f\$cvtime(,,"time")

\$ set noon

\$ set process/prio=12

\$ n = 0

\$loop:

\$ n = n + 1

\$ if n .lt. 10000 then goto loop

\$ elapsed_time = f\$cvtime("-"time_start",,"time")

\$ mm = f\$element(1, ":", elapsed_time)

\$ hss = f\$element(2, ":", elapsed_time) - "." !hundredths of a second

\$ hss = (mm * 6000) + 'hss'

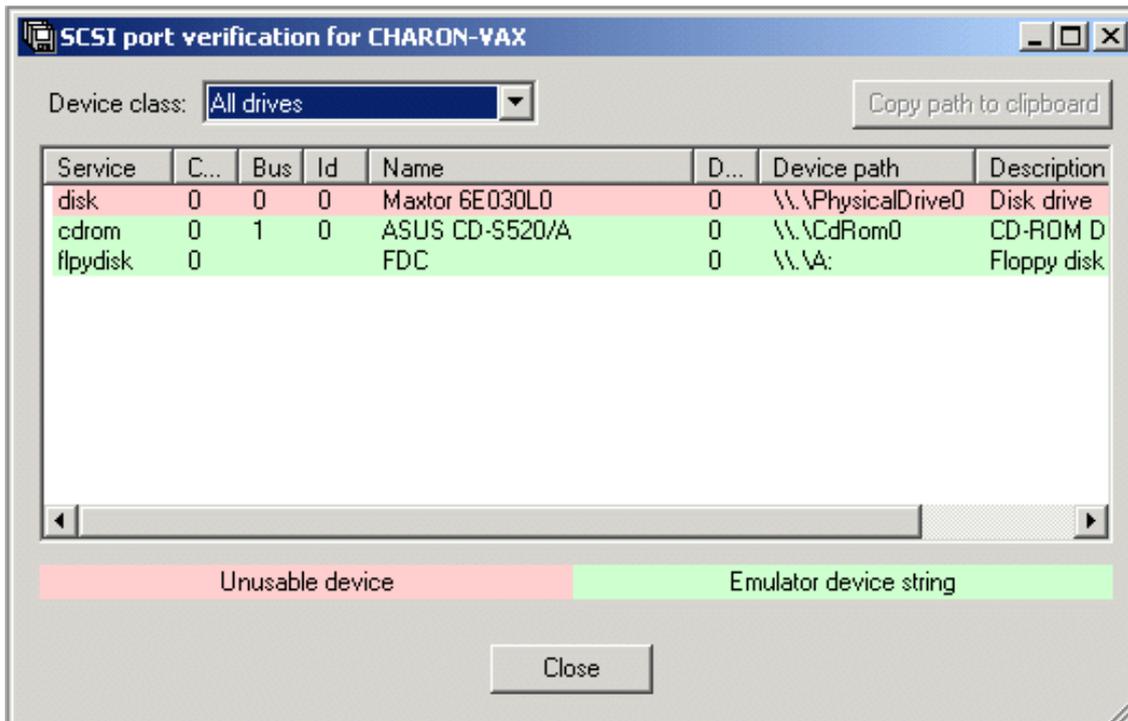
\$ vups = 29300/hss

\$ write sys\$output "VUPs = 'vups' Elapsed time = "elapsed_time"

9. It is, success!

2.5 The SCSI device locator SCSI_CHECK

SCSI_CHECK helps to locate SCSI drives that could be connected to CHARON-VAX. It scans the SCSI controllers in your Windows system and lists the connected devices. The path of the drives found is listed in the format as used in the CHARON-VAX configuration file. Devices that can probably be used by the emulator are displayed with a green background.



The device class select box restricts the display to: Disk drives, Tape drives, CD-Rom drives; Floppy disk drives. Unknown SCSI devices or "All drives".

Floppy drive or CD ROM peripherals can be used by CHARON-VAX. Although they are typically IDE type drives, Windows will list them as a 'SCSI' device on controller 0. Physical disk drives with a Windows file system cannot be used by the emulator and are marked in red.

Tape drives with their Windows drivers installed use the standard "\\.\TapeN" format in CHARON-VAX. When their Windows driver is disabled they can be directly addressed using the "\\.\SCSIx:y:z" format. Older tape drives without a Windows driver (e.g. the TKZ50) are listed by SCSI_check as unknown sequential devices. They are always addressed in the "\\.\SCSIx:y:z" format.

SCSI_CHECK does not scan Logical Unit Numbers (LUNs).

2.6 The Files-11 Browser

The Files-11 Browser is an add-in to the Microsoft Explorer for Windows 2000/XP. It allows to view the contents of VMS-formatted (Files-11 ODS Level 2) CHARON-VAX disk image files.

Files-11 also permits copying of individual files from the disk image to the Windows file system using "Drag and drop" or Copy-Paste operations, even while the emulator is running. Reverse operation (copying from Windows to a CHARON-VAX disk image) is not supported.



After the installation of Files-11 a new icon appears under the "My Computer" node in the Windows Explorer:

To view the contents of a VMS-formatted disk image, it must be opened first. This can be done using the Files-11 context menu (right mouse click on the icon):

Select "Open" and browse for the desired CHARON-VAX disk image. When the file is opened, you can browse the contents of the Files-11 folder just like any other folder in the Windows Explorer.



Transfer files from this folder by using "Drag and drop" or "Copy-Paste" operations. The default copy mode is Ascii, which takes care of the Windows file formatting requirements. Binary mode can be set for special purposes.

The "Open physical drive" option is currently not supported.

2.7 The CHARON-VAX terminal emulator

The internal terminal emulator, available in some versions of CHARON-VAX/Industrial, is a convenience to avoid the need for a hardware MicroVAX VTxxx type console terminal. The functionality approximates VT52 or VT100 emulation with selectable foreground and background colors. The emulator does not pretend to emulate all features of later VT-series terminals and supports a US 101/102 key keyboard layout only.

For every association in the configuration file with a communication line a copy of the terminal emulator is started. Example:

set OPA line="run terminal"

The terminal default setting is VT100/No wrap mode. In addition, the Enter key sends the 'CR' symbol by default. The ASCII and DEC Special Graphics character sets are supported.

If you wish to work with pseudo-graphic symbols on the screen, you must select a fixed-width, non-proportional font, which includes ANSI-compatible pseudo-graphics in the top half of the character map. If you use Double-Height or Double-Width characters on the screen, you must choose a font with the size for which Windows can perform correct rendition of text metrics (i.e. for which double-size equivalents can be found by Windows). The Windows 'Terminal, regular, 11pt' font gives good results in either case.

XON/XOFF sequences are supported. Ctrl+S is supported as a "hold screen" function and Ctrl+Q functions as "release screen". Moreover, the mouse selection also temporarily holds the screen.

When many serial lines are mapped to terminal windows, it can be difficult to determine which line corresponds to which window. To facilitate this, the title of each terminal window can be set in the configuration file with the 'unit' parameter:

set OPA0 line="run terminal" unit="VAX console"

Note: No special characters are allowed in the title string.

The terminal emulator user may set the number of columns and lines. At startup it will set the column width to fit the screen width given the selected font set. The font size, number of columns, rows and buffer size are selectable and saved in the registry.

The terminal window has a menu bar and a terminal screen text area. The menu bar contains the following:

Commands

- *Refresh* resets the terminal to the default state.
- *Exit* terminates the emulator and quits.

Edit

- *Copy* copies the contents of the mouse selection to the clipboard.
- *Paste* translates the contents of the operating system clipboard to the keyboard sequences (pasts the clipboard's contents).

Options

- *Emulation* invokes the Terminal Mode dialog box. Select either VT52 or VT100 emulation mode, choose Line Feed (Carriage Return) expansion mode, enable/disable auto wrap of long lines, and set windows geometry.
- *Colors* set the desired foreground and background colors.
- *Fonts* set the desired font and character size.

Help

- *About* shows the emulator version.

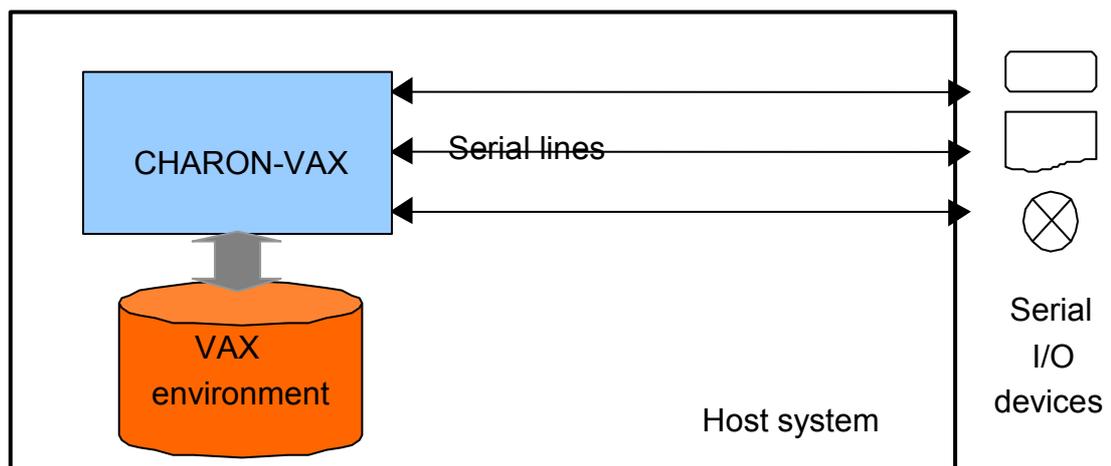
3 Software Interconnects

3.1 Host connectivity via emulated serial lines

The CHARON-VAX/Industrial family of emulators is designed to add or modify application functionality even if it is impossible or undesirable to modify the original VAX code. A flexible and powerful way of achieving this is the use of sockets to represent (emulated) serial line connections. This solution originates out of the requirement to transparently adapt the use of serial line PLCs when replacing industrial VAX systems.

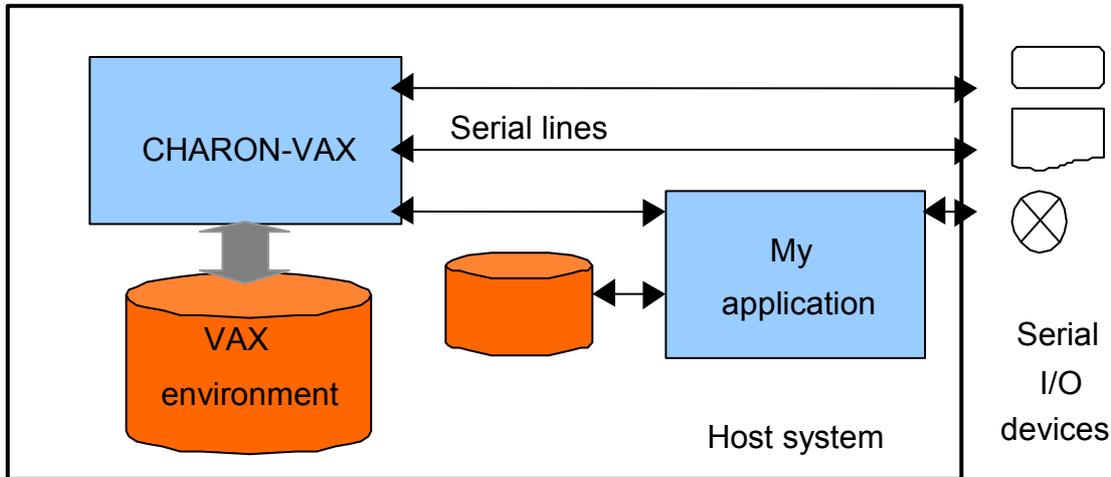
On the Windows platform, an emulated VAX serial line is represented by a socket. In 'standard' use, the emulator connects these sockets to the physical serial I/O ports of the host system, as the following example configuration file entry shows:

```
load DHV11 YVA  
  
set YVA line[1]="COM1:"
```



Instead of a physical serial port, CHARON-VAX/Industrial also accepts a process name in the line assignment, allowing to start a user written task, and to connect this task to the provided socket. The CHARON-VAX/Industrial emulator automatically invokes such task during the scanning of the configuration file:

```
load DHV11 YVA
set YVA line[1]="run c:\my_application.exe"
```



Programming example

The programming example **VAXPRINT** is a print utility to send data from a CHARON-VAX serial line to the default printer (using its default page settings) of a Windows host system by using the following entry in a configuration file:

```
load DHV11 YLA
set YLA line[1]="run vaxprint"
```

The VAXPRINT source code and the executable for Windows can be downloaded as VAXPRINT.ZIP and VAXPRINT.EXE respectively from the "downloads" area of the CHARON-VAX web site at <http://www.CHARON-VAX.com>. The source code is provided as a programming example for other applications.

3.2 The CHARON virtual LAN adapter (MUX driver)

(This option is available as of CHARON-VAX version 3.0)

The CHARON virtual LAN Adapter provides physical Ethernet adapter sharing between a CHARON instance, the Windows host system and any other applications (e.g. an Xterminal emulator) running on the same host system. The CHARON virtual LAN Adapter is implemented as a Microsoft NDIS 4.0 / NDIS 5.1 intermediate driver. It is connected to a physical Ethernet (NIC) adapter in the host system and creates a configurable number (maximum 255) of virtual NIC adapters.

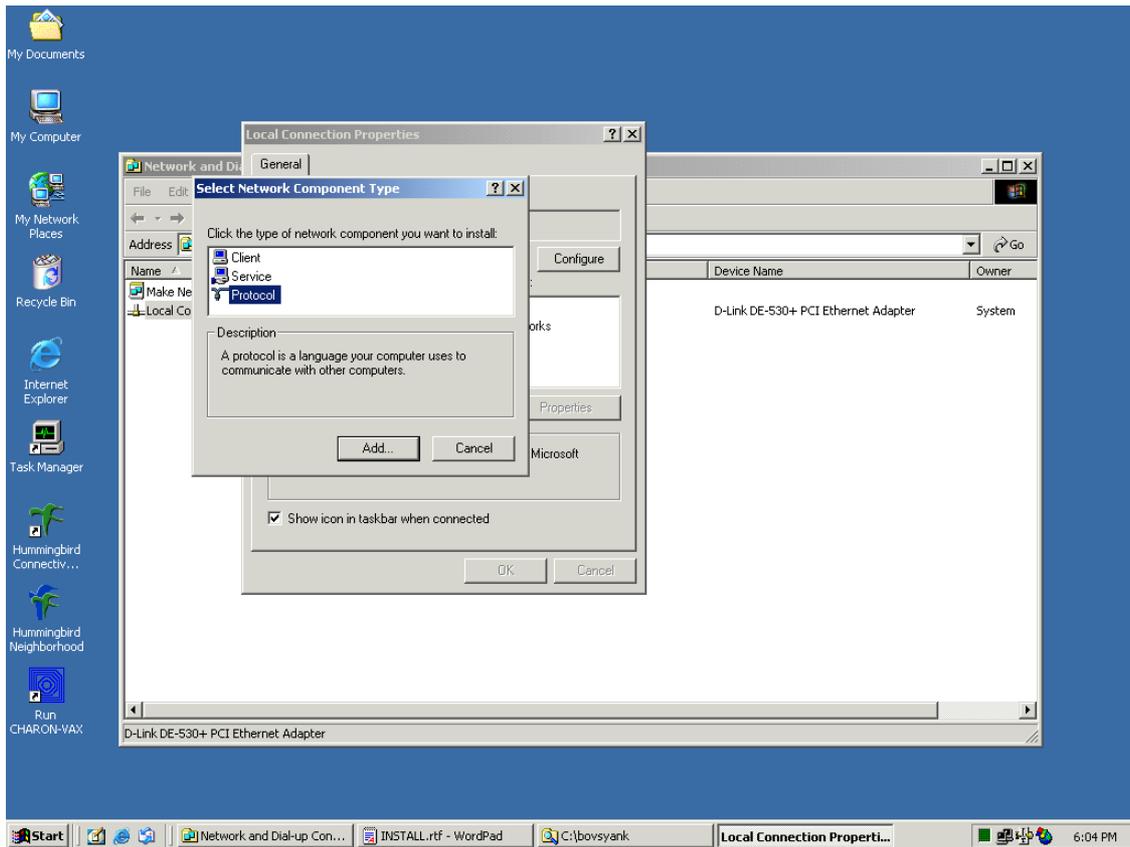
To reduce system overhead, the CHARON virtual LAN Adapter optionally implements per-VLAN (virtual NIC) filtering, which sends the packet stream input from the physical adapter only to specified addresses among the processes that share the same physical adapter.

In VLAN mode, packets sent through a virtual NIC are delivered locally to one or more other virtual NICs, or sent to the physical adapter or both, depending on the filtering. In pass-through mode, packets are delivered to the physical adapter regardless of filtering.

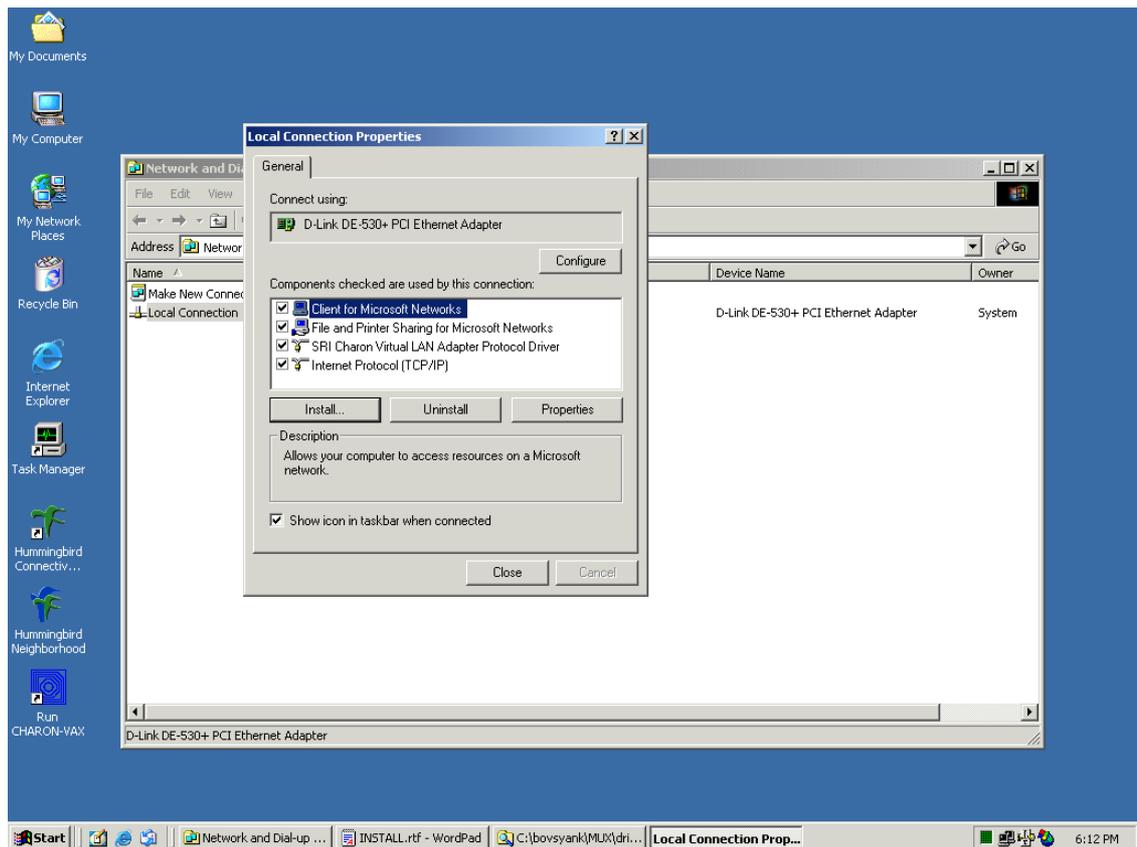
Installation of the CHARON virtual LAN Adapter

This example assumes that the required `srinux.*` files are in the Drivers | Networking subdirectory of the CHARON-VAX installation directory:

- Open the 'Network and Dial-up Connections' dialog.
- Select the connection that you like to use for the CHARON virtual LAN Adapter (either a physical Ethernet adapter or the logical Microsoft Loopback Adapter) and display its properties.
- Click the install button.



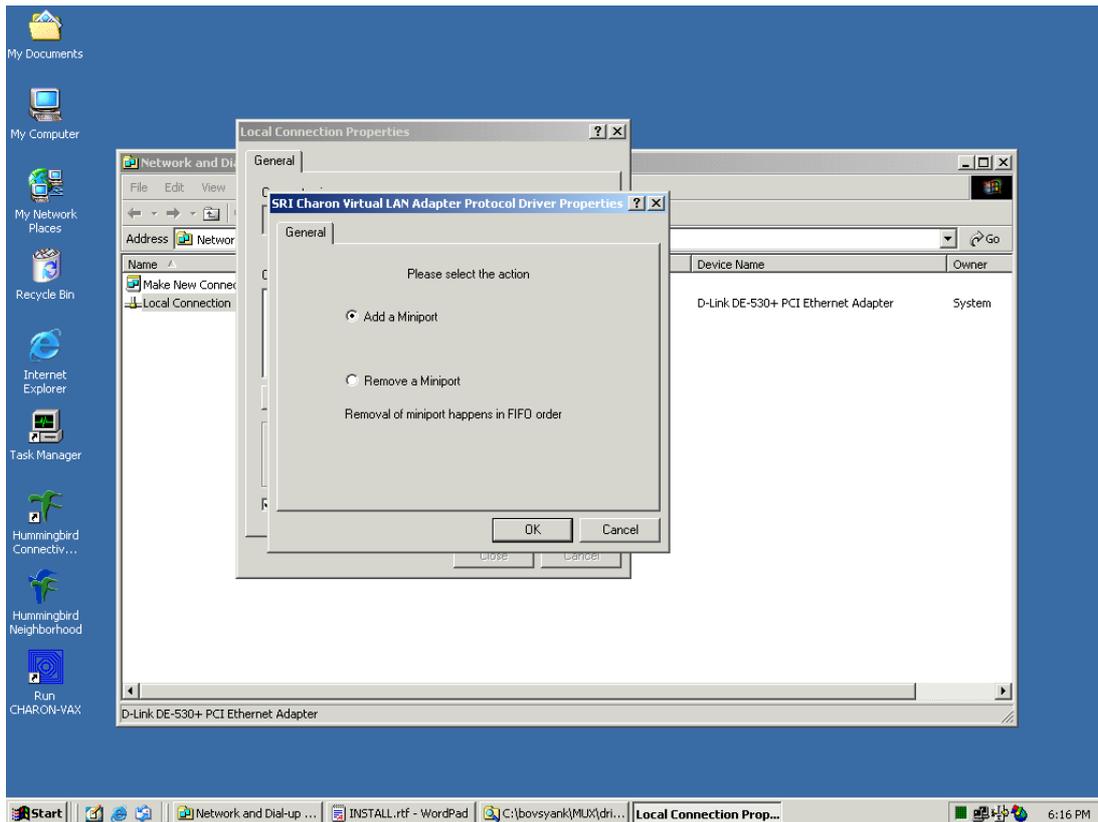
- Select the Protocol & press the 'Add...' button. In the resulting dialog press the 'Have Disk' button. Select the path to your SRIMUX.* driver files and click Ok.
- Click Yes to confirm that you want to install an unsigned package.
- Select the Protocol (the CHARON virtual LAN Adapter protocol driver) and click OK.
- This installs the CHARON virtual LAN Adapter protocol driver. Note that at this point the driver is enabled for all physical adapters you have installed on the host system. Since the purpose of the CHARON virtual LAN Adapter is to concentrate all relevant LAN traffic on a single physical adapter, disable the driver on any other adapters you might have in this system. When configured, this driver will spawn virtual NICs that can be used by the host system or CHARON (but never both).



- In the above screenshot we do not yet spawn any virtual adapter from the 'D-Link DE...' adapter and still have all protocols enabled. When at least one virtual adapter is created, all protocols except the CHARON virtual LAN Adapter protocol driver will be disabled on the physical adapter. Do not enable any other protocols manually on that **physical adapter** as it will result in networking problems. (In contrast, on any created **virtual adapter** the CHARON virtual LAN Adapter Protocol Driver should always be **disabled**).

Installation of a virtual adapter

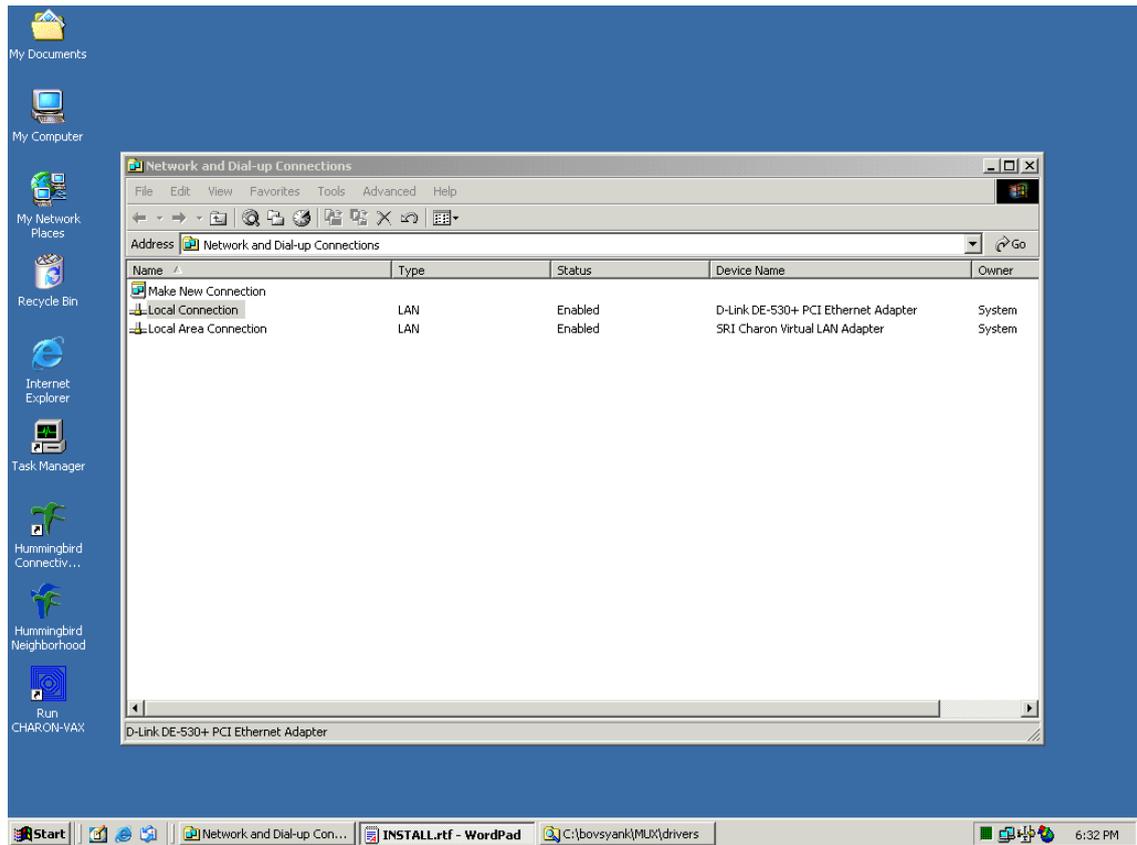
To add a virtual NIC, select the CHARON virtual LAN Adapter protocol driver and click properties.



From the physical adapter Properties applet you can Add or Remove a virtual NIC:

- Select Add and click 'Ok' to add a virtual adapter. (Note that this dialog will be re-designed to show a list of the currently spawned virtual adapters on this physical device).
- Confirm that unsigned software will be installed by pressing 'Yes' as many times as the system will ask.
- Close connection properties and go back to the Dial-Up & connections applet to finally create the virtual adapter.

CHARON-VAX Utilities and Interconnects

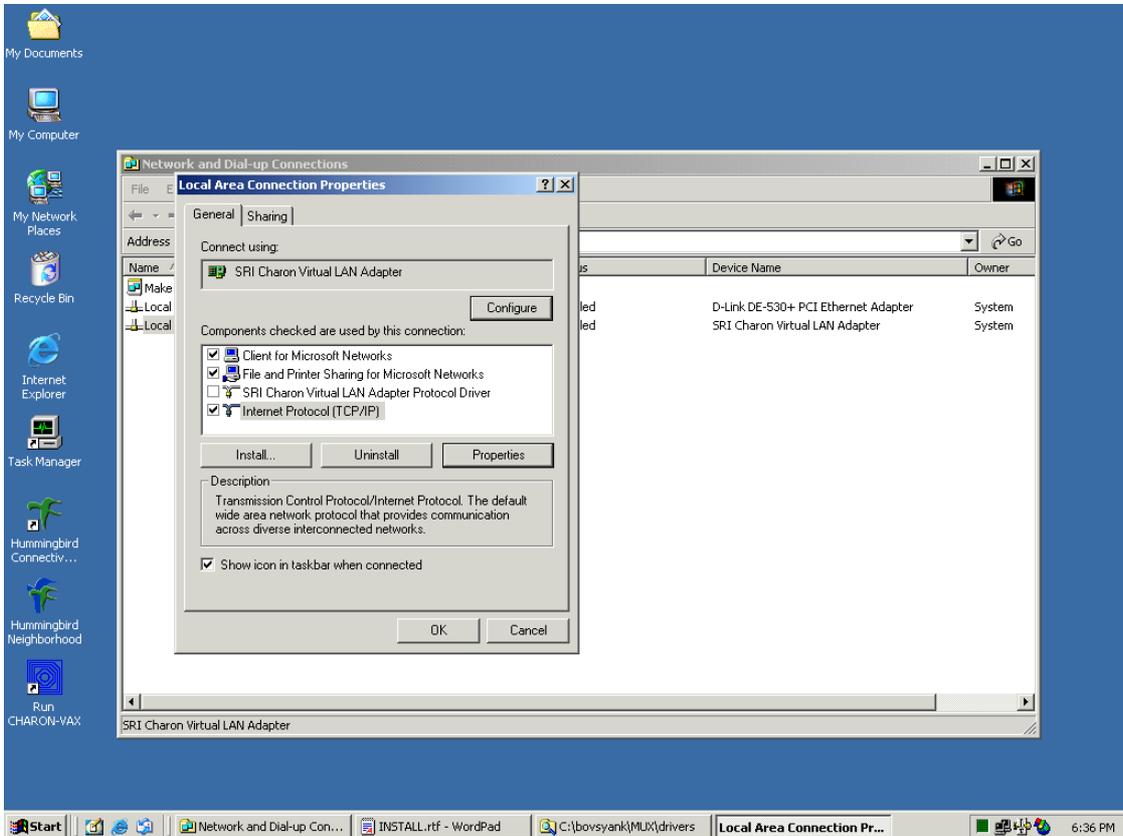


You now have a new connection for the CHARON virtual LAN Adapter' (In a next version the connections will have a #<number> at the end). This is an instance of the virtual LAN connector routed through the physical adapter using the CHARON virtual LAN Adapter protocol driver.

Verify that your physical adapter has only the CHARON virtual LAN adapter Protocol Driver' enabled: select the physical adapter and click properties.

Configuration of a virtual adapter

To configure a virtual adapter select it from the Dial-Up & Connections dialog and show its properties:

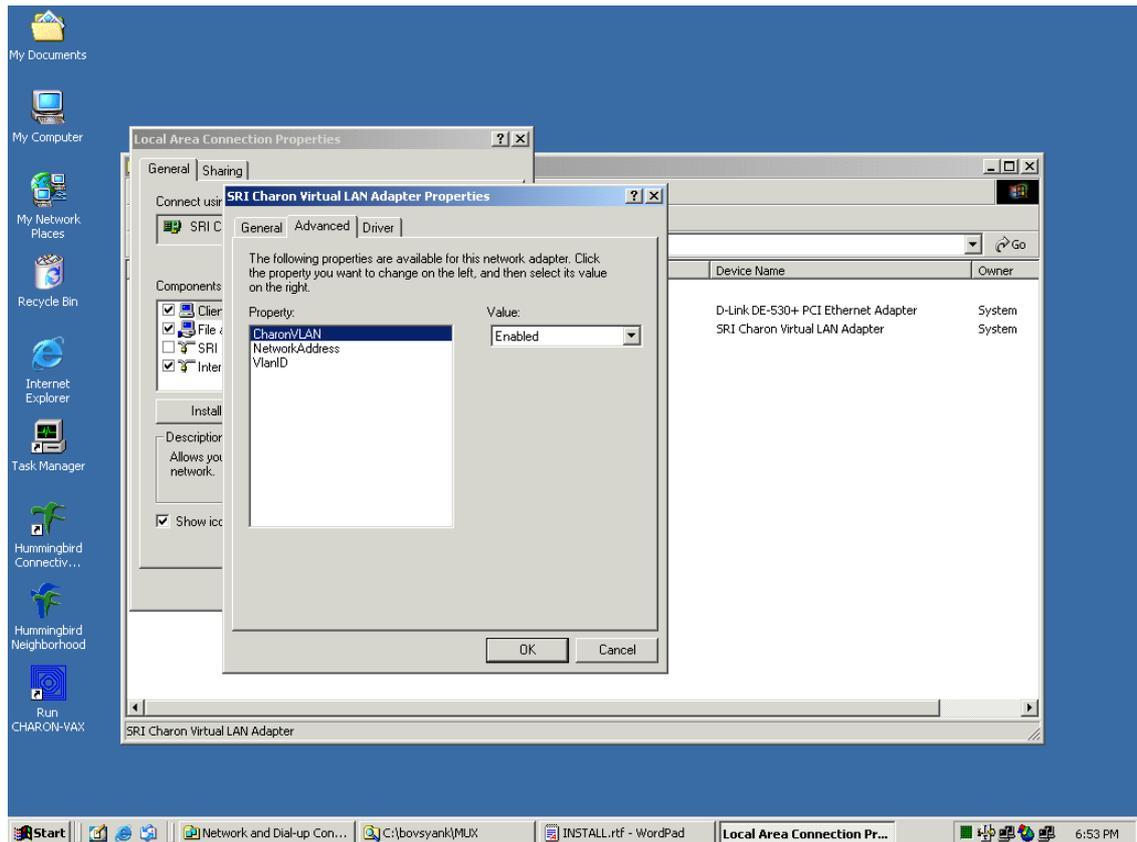


Every virtual adapter is configured in the same way as a physical one.

The host system link

In this example the first virtual adapter is used for the host system TCP/IP connection: configure TCP/IP, enable net bios, etc. Note that in the above screen shot for the host system the CHARON Packet Protocol Driver (NDIS5) is not installed.

Normally the default property settings of the virtual adapter are sufficient. To verify its settings, open the virtual connection's properties dialog and click configure and select the 'advanced' panel:



The 'advanced' panel sets the following parameters:

CharonVLAN

This Boolean parameter is set to enable by default. When enabled, the virtual adapter is able to communicate with other virtual adapters connected to the same physical one. Otherwise (in the pass through mode), all packets are sent to the network without an attempt to deliver them to other virtual adapters connected to this physical one.

NetworkAddress

This is the current MAC address, and has the same meaning as for a physical adapter. Note that the use of the Netsetup utility is not tested yet for virtual adapters.

VlanID

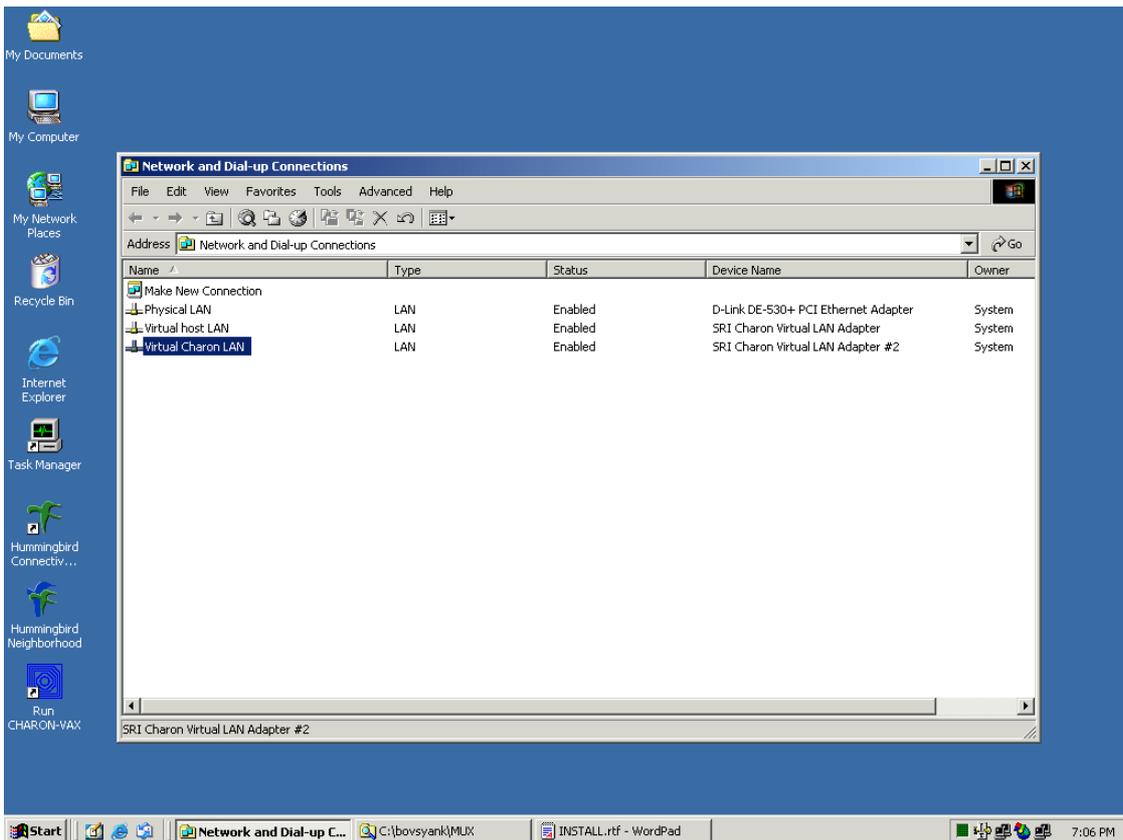
This is a unique number for all virtual adapters connected to the same physical adapter. This number is used to establish the virtual adapter's permanent MAC address; the driver insures that it is unique. The minimum is 1 and the maximum is 255.

Connecting the CHARON emulator:

The CHARON Ethernet implementation uses the CHARON Packet Protocol (NDIS5). Install this protocol as described in the CHARON-VAX product documentation. Up to now, we installed:

- The CHARON MUX driver on the 'D-Link DE-530+ PCI Ethernet Adapter', a connection re-named to 'Physical LAN'
- Configured TCP/IP for the host system on the CHARON virtual LAN Adapter', a connection re-named to 'Virtual host LAN'
- Installed the CHARON Packet Protocol Driver (NDIS5).

As we need an additional virtual connection for the CHARON emulator, install another virtual adapter. As result we have one more connection, renamed to 'Virtual CHARON LAN':



Verify the protocol settings:

- The 'Physical LAN' connection must only have the CHARON virtual LAN Adapter protocol driver' enabled.
- The 'Virtual host LAN' will have all protocols enabled required for the host system but **not** the CHARON Packet protocol driver (NDIS5) and **not** the

CHARON virtual LAN Adapter protocol driver.

- The 'Virtual CHARON LAN' must have **only** the CHARON Packet protocol driver (NDIS5) enabled.

In this way we have separated the host system networking from the CHARON emulator networking via separate virtual LAN adapters.

The 'Virtual CHARON LAN' connection has as device name 'CHARON Virtual LAN Adapter #2'. This name can be used in the CHARON configuration file. For example (CHARON release 1.2.3 or later):

```
load ndis5_packet_port eza_0 interface='CHARON Virtual LAN Adapter #2'
```

In this example, the host system and CHARON's network connect through the same physical adapter. Instead of a physical adapter you can use the 'Microsoft Loopback Adapter'. In that case there is no need for a physical Ethernet adapter to connect the Operating system running on the CHARON emulator to the host system via TCP/IP.

How does It work?

A simple configuration is the following:

LAN

<-> Physical NIC

<-> CHARON Packet Protocol Driver (NDIS5)

<-> ndis5_packet_port instance

<-> controller emulation (SGEC, DEQNA, etc...)

In this example the CHARON Packet Protocol Driver (NDIS5) is the only enabled protocol on the physical NIC.

Packets received on the physical adapter go through the Physical NIC driver to the CHARON Packet Protocol Driver (NDIS5) from where they are read by the ndis5_packet_port instance and delivered to the NIC card emulation (SGEC, DEQNA, etc...).

The transmit process happens in the opposite way. The NIC card emulator writes packets to the ndis5_packet_port instance, which in turn passes it to the CHARON Packet Protocol Driver (NDIS5), which delivers the packet to the physical NIC driver.

The example in the installation procedure in this chapter is more sophisticated:

LAN

- <-> Physical NIC (D-Link DE-530...) (Physical LAN)
- <-> CHARON Virtual LAN Adapter Protocol Driver'
 - <-> 'CHARON Virtual LAN Adapter' (Virtual host LAN)
 - <-> TCP/IP
 - <-> net client
 - <-> file printer share
- <-> CHARON Virtual LAN Adapter #2'(Virtual CHARON LAN)
- <-> CHARON Packet Protocol Driver (NDIS5)
 - <-> ndis5_packet_port instance
 - <-> controller emulation (SGEC, DEQNA, etc...)

Like in the simple configuration, the physical NIC driver receives a packet from the LAN. The Physical NIC driver sends the packet to the CHARON Virtual LAN Adapter Protocol Driver.

Note that if you would enable other protocols on the physical NIC, the same packet will be sent to these protocols as well which will create networking problems.

The CHARON Virtual LAN Adapter Protocol Driver sends the packet to the CHARON virtual LAN Adapter' or the CHARON virtual LAN Adapter #2' or both depending on the packet and on the packet filters installed on virtual adapters.

For example, when both adapters enable broadcasts, a broadcast packet will be delivered to both virtual adapters. At the same time, a direct frame will be delivered only to the virtual adapter it is targeted to (except in the situation where another virtual adapter is placed in promiscuous mode). Packets delivered to the virtual adapter are forwarded in the same way as in the previous simple configuration.

The transmit process also differs from the first example. When a packet is delivered to the virtual adapter, and the virtual adapter's CharonVLAN parameter is enabled, the virtual adapter will try to loop the packet to the other virtual adapters while applying the target virtual adapters packet filter in the same way as during packet receipt. At the same time, the packet is filtered and sent to the physical NIC driver when matched. This implies that the packet might be looped to other virtual adapter(s), sent to the physical connection or both depending on packet and virtual adapter packet filters.

Further information

The web site www.charon-vax.com provides information about the CHARON-VAX products. If you have questions or suggestions, you can contact us as by sending an Email to charon@softresint.com.

We also appreciate your comments, suggestions, criticism and updates of this manual. If you found any errors, please refer to:

The CHARON-VAX utilities and interconnects manual,

Document number: SRI-30-16-015

with the publication date and the page numbers concerned.