Virtual Alpha User Guide

for Avanti, AvantiFlex and FreeAXP

Version 5.0.0

Migration Specialties Europe

8-JAN-2025

migrationspecialtieseurope

Virtual Alpha User Guide

Printed: 8-JAN-2025 in the Netherlands

Inquiries

Please direct all inquiries regarding this document or the products described therein to:

Migration Specialties Europe
Slawijkseweg 1
7077AM The Netherlands

www.migrationspecialties.nl

camiel@migrationspecialties.nl

Introduction

VIRTUAL ALPHA

1 Introduction

This manual describes how to install and use Avanti, AvantiFlex, and FreeAXP, Migration Specialties' AlphaServer 400 hardware emulators.

Avanti and AvantiFlex are commercial products that require purchase of a product license. These products include 30 days of manufacturer support after purchase and the option to buy an extended support contract.

FreeAXP can be used for personal and commercial purposes. FreeAXP is unsupported without purchase of a support contract. If you have questions or problems with FreeAXP and have not purchased a support contract, visit the FreeAXP user forum at the OpenVMS Hobbyist web site.

1.1 Avanti, AvantiFlex and FreeAXP Links (2)

Avanti and AvantiFlex Info:	https://www.migrationspecialties.nl/products/avanti
FreeAXP Info:	https://www.migrationspecialties.nl/products/freeaxp
User Guide (PDF):	https://www.migrationspecialties.nl/Downloads/VirtualAlpha_UserGuide.pdf
Online User Guide:	https://www.migrationspecialties.nl/userguides/avanti
Release Notes:	https://www.migrationspecialties.nl/Downloads/VirtualAlpha_ReleaseNotes.pdf
Pricing Guide:	https://www.migrationspecialties.nl/Downloads/VirtualAlphaPricingGuide.pdf
User Forum:	https://www.vmshobbyist.com/forum/viewforum.php?forum_id=163
Download:	https://www.migrationspecialties.nl/products/freeaxp

1.2 Document Conventions

This document uses the following conventions:

1.2.1 Virtual Alpha Product References

Throughout this manual, Avanti will be used to represent the Migration Specialties virtual Alpha products Avanti, AvantiFlex, and FreeAXP. Most installation, configuration, and runtime tasks are the same for all three products. Where distinctions are necessary, the products are referenced by name.

1.2.2 Size Designations: K, M and G

Some parameters used by the configuration utility designate file or buffer sizes. The suffixes K, M, and G are used to designate Kilo, Mega, and Giga. For example, to set the autocreate size of a disk to 4 gigabytes, the user would enter 4G in the autocreate_size box.

1.3 Migration Specialties Products & Services

Product	Description
Avanti/AvantiFlex	Commercial Alpha emulator
FreeAXP	Free Alpha emulator
SimH VAX	Free open source VAX emulator
OpenVMS Consulting	OpenVMS consulting and programming services
Tru64 UNIX Consulting	Tru64 UNIX consulting and programming services

Quick Start

VIRTUAL ALPHA

2 Quick Start

This section describes the basic steps required to get up and running with Avanti.

1) Download and install the Avanti software: Installation 18
2) Install the License Key (Avanti & AvantiFlex only): 3.3 Avanti Licensing 23
3) Launch the Configuration Utility: 4.1 Avanti Configuration Utility 28
4) Configure a virtual Alpha system containing at least one…
   a) Serial Port: 4.1.7.5 Serial Ports: OPA0/tty00 – TTA0/tty01 34
   b) Disk Controller: 4.1.7.6.1 SCSI Controllers: KZPAA and KZPCA 36
   c) Disk Drive: 4.1.7.6.1.1 Image File 37
   d) CD-ROM: 4.1.7.6.1.2 CD-ROM 37
   e) NIC: 4.1.7.6.2 NIC 38
5) Save the configuration file: File 30
6) Load a Tru64 UNIX or OpenVMS installation CD in the CD-ROM device.
7) Launch Avanti and install Tru64 UNIX or OpenVMS.

2.1 Porting Services

Migration Specialties provides comprehensive fee-based Tru64 UNIX and OpenVMS porting services. We are available to help build and port to virtual Alpha systems. Contact us for additional information at camiel@migrationspecialties.nl.

2.2 Critical Factors Impacting Avanti Operation

This section highlights factors critical to reliable operation of an Avanti virtual system.

2.2.1 Unplanned Process Termination

Avanti processes run one of two ways: as a user mode process or as a service. If a user mode session is terminated, the Avanti process disappears with it. If a Windows system is rebooted, the Avanti service will be terminated. Either way, the virtual Alpha is crashed.

Recommendations:

- When running Avanti as a user process, disable auto-logout of the user process.
- When running Avanti as a service, disable Windows automatic updates and reboots.

2.2.2 VMware and DECnet

If DECnet is in use and Avanti is running on a VMware VM, the port group assigned to Avanti must have the following features enabled:

- Promiscuous mode
- MAC address changes

- Forged transmits

See 6.8 VMware NICs 69 for more details.

Installation and Licensing

3 Installation and Licensing

3.1 Installing Avanti

Note: Avanti, AvantiFlex, and FreeAXP cannot be installed if any of these virtual Alpha solutions are active. The installer checks for active instances during the installation process. If a virtual Alpha is running, an error box similar to one shown will be displayed.

3.1.1 Security

Avanti, AvantiFlex and FreeAXP installation kits utilize Microsoft Windows digital signatures. These signatures allow Windows to detect corrupt installation kits. If Windows objects to installing a kit, check with Migration Specialties or obtain a new installation kit from a trusted source before continuing the installation.

3.1.2 Prerequisites

Avanti requires the following minimum O/S and hardware.

3.1.2.1 Operating Systems

Avanti, AvantiFlex, and FreeAXP run under the following 64-bit Windows Operating systems:

- Windows 8 and newer
- Windows Server 2008 and newer

3.1.2.2 Antivirus Recommendation

It is recommended to place the Avanti executable (dec.exe) and Avanti disk container files (.vdisk) on security and anti-virus exclusion lists. Virus scans may interfere with Avanti operations and degrade performance.

3.1.2.3 Hardware

Avanti requires the following minimum host hardware. Faster hardware will yield better emulator performance.

- Dual core processor (3GHz or faster recommended)
- 8GB memory
- 2 NICs

3.1.2.3.1 Sizing Host Memory

Avanti requires the program code + base Alpha memory + Alpha shadow memory + JIT memory + asynch threads + IO buffers to run each Avanti instance. This allocation comes out of the physical memory of the host system. Physical memory in a physical host system is usually divided between the number of physical processors in the system. Cores are not allocated memory, they can access all of the memory assigned to the physical processor that they are on, and can access memory assigned to the other physical processors over a slightly slower internal bus. As long as the system has enough physical memory, there should be no problems in running Avanti. If there isn't enough physical memory, operations slow down as the system pages physical memory to virtual memory on disk.

Avanti utilizes host memory as follows:

- Configured Memory: One-for-one mapping of virtual Alpha memory to host memory. An Avanti system configured to use 1GB of memory will utilize up to 1GB of host memory.
- Shadow Memory: One-for-one mapping of shadow memory to host memory. Avanti pre-processes Alpha instructions, optimizing them to achieve better performance upon execution. These optimized instructions are referred to as shadow code. Shadow code is retained in host memory as long as the Alpha code remains in emulated memory.

Shadow memory matches the memory configured for an Avanti instance. An Avanti system configured with 1GB of memory will utilize up to 1GB of host memory for shadow memory.

- JIT Memory: Default maximum size is 2GB or ½ of physical memory, whichever is less. JIT memory utilization can be limited with the jit_memory advanced configuration option (see 4.1.8.3 JIT [+] 43).

  Avanti reserves JIT memory as follows:

  - Reserve 2GB of virtual memory for the JIT buffer.
  - Commit the first 4MB chunk.
  - If commit fails:
    - Unreserve the reserved 2GB.
    - Reserve and commit the entire 2GB buffer at once.
  - If that fails, error out
  - The emulator will quit and there will be an error in the log file stating that the JIT buffer could not be allocated.

Note:If running on a Windows host system equipped with more than 32GB of memory and the following error is encountered, set the jit_memory parameter to 2GB.

Unable to reserve JIT buffer area: error 87: The parameter is incorrect.

Optimum host memory on a Avanti virtual Alpha system configured with 1GB of virtual memory would break down as follows:

Memory	Avanti	Host
Host O/S		4GB

Avanti Configuration	1GB	1GB
Shadow Memory	1GB	1GB
JIT	2GB	2GB
Total		8GB

All the caveats of a Windows paging file apply. For most situations, and our testing supports this, a host system with 8GB of memory and a properly configured Windows paging file would be more than adequate to support a 1GB Avanti configuration. Only in situations where very high demand is put upon the Avanti virtual Alpha would maximizing physical memory be required

3.1.3 Rights

Avanti should be installed from an Administrative account. Users will need administrative rights to run the Avanti configuration utility and to launch Avanti from the command line or a batch procedure.

3.1.4 Installation Restrictions

- Avanti cannot be installed if another instance of Avanti is running. All instances of Avanti must be shut down before a new installation can be made.
- Avanti cannot be installed on a 32-bit Windows system.

3.1.5 Installation Steps

1) The installation package can be downloaded from:

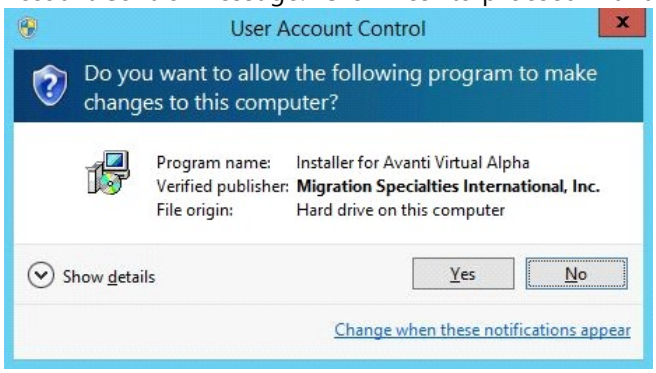
   http://www.migrationspecialties.nl/products/FreeAXP

2) Run the installation program as an administrator. Right click the installation program, then select "Run as administrator" from the pop-up menu.

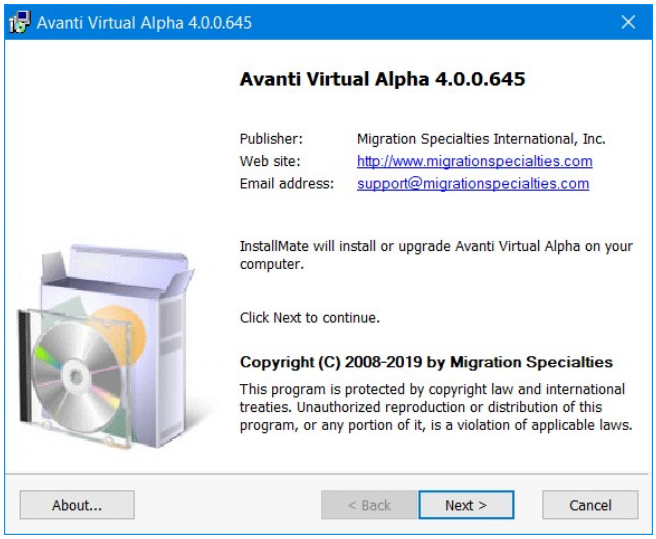
Note:Failure to run the installation as an administrator may result in no access to network devices.

Installation Program: decsetup_nnn.exe (where nnn is a version number)

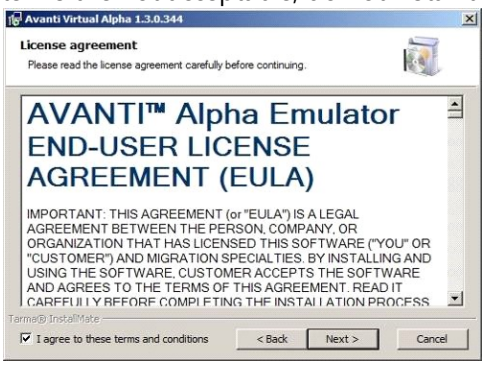
3) Installation Warning: When the installation is started, Windows may display a User Account Control message. Click "Yes" to proceed with the installation.



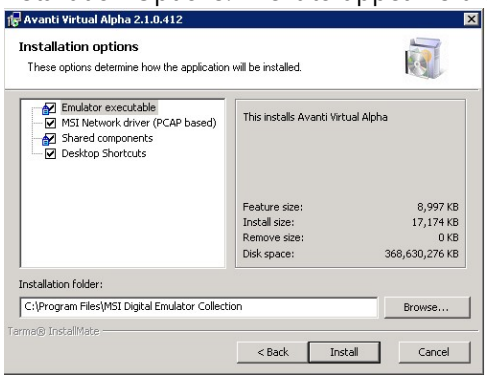
4) Information Window: Next, an informational window will appear. Note the copyright information, then click "Next" to continue.



5) License Window: Next, the Avanti license window will be displayed. The eminently fair and reasonable license terms should be read. Check the "I agree to these terms and condition" box and click "Next" to continue the installation. If the license terms are not acceptable, do not install the product.



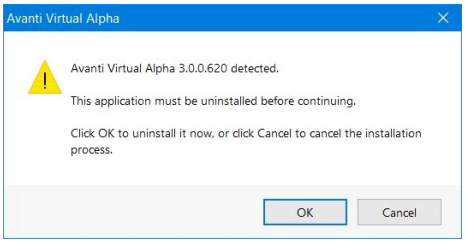
6) Installation Options: Next to appear is the Installation Options window.



The window lists the following components:

- Emulator executable: The virtual Alpha program. Emulator installation is required and cannot be unchecked.

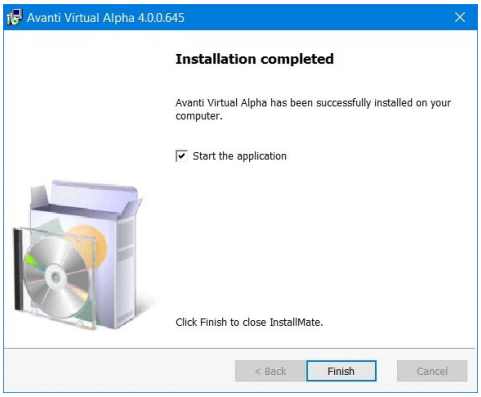
- MSI Network driver: The PCAP based network driver. Installation of the network driver is optional. If the network driver is not installed, the emulator cannot be configured with a NIC. The network driver is installed by default. If the Npcap driver is installed on the Windows system, Avanti can also use it as its pcap driver.
- Desktop Shortcuts: Adds desktop shortcuts for PuTTY and the Avanti Configuration Utility.
- Installation folder: Allows a different installation folder to be specified.



Previous Installation Warning: If a previous version of Avanti is installed on the system, a warning box will be displayed. Click "OK" to de-install the previous version of Avanti and install the current version or click "Cancel" to abort the installation process.

As the previous version of Avanti is de-installed, a command window may flash past. This is normal.

Active Instance Abort: If an instance of FreeAXP or Avanti is running, the installation will abort with a Cancel/Retry dialog.



Installation Completed: The Avanti installer will complete the installation. A progress bar and two command windows will briefly display during the installation process. When the installation is finished, a completion window will be displayed. The completion window contains a check box offering to launch the Avanti configuration utility. The default is to launch the utility. The Virtual Alpha Configuration 28 chapter describes the utility in detail. Click "Finish" to exit the installation process.

3.1.6 Installation Details

Installation Folder

By default, the installer creates the following folder on most English-language installations of Windows. Virtual Alpha components are installed in this folder. The Installation Options window allows the user to change the default installation folder.

Virtual Alpha	Default Installation Folder
Avanti	C:\Program Files\MSI Digital Emulator Collection

Firewall Exception

The Avanti installer will create an exception for Avanti in the default Windows firewall. If a non-Windows firewall is deployed, an entry may need to be added to support the emulator console connection.

Virtual Alpha	Default Firewall Entry
Avanti	C:\Program Files\MSI Digital Emulator Collection\dec.exe

Installation Log File

The Avanti installer creates an installation log file in the user's TEMP folder.

```
Installation Log File Folder:    c:
\users\<username>\appdata\local\temp folder
Installation Log File Name:      decsetup-yyyyddmmThhmmss-install.log
```

3.2 De-installing Avanti

Avanti can be de-installed from Windows using the Control Panel function Programs and Features.

When Avanti is de-installed, all configuration files created using the Configuration Utility are retained. Files created outside of the installation directory are also retained.

3.3 Avanti Licensing

Avanti and AvantiFlex require a license key to run. The license key, which gets loaded into the Windows registry, provides Avanti license units. Avanti feature availability is controlled by license units. Only one Avanti license is permitted per host system.

A full Avanti license provides 175 license units. An AvantiFlex base license provides 10 license units. AvantiFlex allows the user to select desired Avanti features without purchasing a full Avanti license. See the Avanti SPD for more information on product licensing.

FreeAXP does not need any license units and does not require a license key.

3.3.1 Transitioning from FreeAXP to Avanti

FreeAXP is Avanti with no license units. The same installation kit is used for both products. FreeAXP provides a limited subset of Avanti features. Installation, configuration, and running are identical to Avanti. Transitioning to Avanti is as simple as loading a license key into the registry. Both FreeAXP and Avanti can co-exist on a single host system.

3.3.2 Avanti Version and License Key Information

The Avanti log file header contains Avanti version and license keys information. The log file can be open by any text editor, such as Microsoft Notepad. 3.3.2.1 Temporary License Example 24 and 3.3.2.2 Permanent License Example 24 provide examples of log headers for temporary and permanent Avanti license keys.

3.3.2.1 Temporary License Example

```
Avanti Virtual Alpha version 4.0.0.645 (Feb  3 2020 13:57:49)

Windows 10 Pro Version 1903 Build 18362.476
(18362.1.amd64fre.19h1_release.190318-1202) (VersionEx: Windows
workstation version 6.2 SP 0.0, build 9200 () suite 100)
Intel(R) Core(TM) i7-7500U CPU @ 2.70GHz (4 processor cores of family
8e, stepping 09)
File opened at 2020-02-06 13:10:09

License key: s/n 133r23 (v. 1)   400 units
20200206131016.076:                          Expires in 328
days
20200206131016.076:                        Valid for releases up
to 2019-12-31
```

- **Avanti Virtual Alpha version 4.0.0.645 (Feb 3 2020 13:57:49)**
  - Avanti version 4.0.0.645 built February 3, 2020.
- **License key: s/n 133r23 (v. 1) 400 units**
  - Key #133 configured with 400 unit license
- **Expires in 328 days**
  - Temporary key expiring in 328 days
- **Valid for releases up to 2020-12-31**
  - Avanti updates released before 2020/12/31 can be applied to this key.
  - The date reflects the support contract period purchased for the key. When the support contract is renewed, this field is updated.

3.3.2.2 Permanent License Example

```
Avanti Virtual Alpha version 4.0.0.645 (Feb  3 2020 13:57:49)

Windows Server 2012 R2 Standard Build 9600.19599
(9600.19538.amd64fre.winblue_ltsb_escrow.191014-1700) (VersionEx:
Windows server version 6.2 SP 0.0, build 9200 () suite 110)
Intel(R) Xeon(R) CPU E31230 @ 3.20GHz (8 processor cores of family 2a,
stepping 07)
File opened at 2020-02-06 10:29:35

20200206102940.975: License key: s/n 2r21 (v. 1)    800 units
20200206102940.975:                          Valid for releases up
to 2020-12-31
20200206102940.975: %XNV-I-RESTST: NVRAM restored Figure 1: Temporary
License Key
```

- **Avanti Virtual Alpha version 4.0.0.645 (Feb 3 2020 13:57:49)**
  - Avanti version 4.0.0.645 built February 3, 2020.

- **License key: s/n 2r21 (v. 1) 800 units**
  - Key #2 configured with 800 unit license
- **Valid for releases up to 2020-12-31**
  - Avanti updates released before 2020/12/31 can be applied to this key.
  - The key itself is permanent. It does not expire. The date reflects the support contract period purchased for the key. When the support contract is renewed, this field is updated.

3.3.3 Updating a License Key

Migration Specialties' commercial Alpha virtualization solutions rely on a registry-based license key to enforce license compliance. Customers may need to update their license key under the following circumstances:

- A service contract renewal.
- A demonstration key is about to expire.
- A demonstration key is being upgraded to a permanent license.
- A permanent key is being updated with additional license units.

A license key for a new or updated license will be provided by Migration Specialties via e-mail. Each license key is unique to the license for which it is generated.

The license key will look like this:

```
== Paste from here ==>
akic1iralr85Vuqb4hu+
THIS IS NO VALID KEY
QP7CkgpSwg
<== to here ==
```

To install the license key, open the Virtual Alpha Configuration Utility, and go to Help -> Install License Key. You will be presented with a dialog box into which you can paste your license key. Another dialog box will confirm proper installation of the license key and display the key's details.

Why Update a Permanent License Key?

A permanent license never expires, so why update the key? The benefit to updating the key is that the support date field on the key gets updated to the new support expiration date. If you try to apply an Avanti software update that was issued after the support date recorded on the key, the update will fail.

Virtual Alpha Configurations

4 Virtual Alpha Configurations

Avanti virtual Alpha configurations are easily created using the Avanti Configuration Utility. Multiple configurations can be created and stored on a system. Avanti configurations support the following virtual hardware.

- 1 CPU supporting the EV4, EV5, and EV6 instruction sets.
- Two virtual or physical serial ports, one of which is the console port.
- 32MB to 3GB1.
- 1 - 3 NICs.
- 1 – 3 disk controllers supporting up to 45 disk devices, including multiple physical CD-ROM drives or hard drives.

AvantiFlex has the same maximum limits as Avanti. AvantiFlex installation capabilities are controlled by the number of available license units.

FreeAXP configurations are limited to the following virtual hardware.

- 1 CPU supporting the EV4, EV5, and EV6 instruction sets.
- Two virtual serial ports, one of which is the console port.
- 32MB to 128MB.
- 1 - 2 NICs.
- 1 disk controller supporting up to 7 disk devices on one controller, one of which can be a physical CD-ROM drive.

4.1 Avanti Configuration Utility

The Avanti Configuration Utility provides an intuitive GUI interface to define virtual Alpha hardware. Configurations are stored in ASCII text files that are easily transferred to other systems. This chapter describes the Avanti Configuration Utility in detail.

4.1.1 Avanti Configuration Files

The Avanti Configuration Utility generates a text file containing a script that defines the desired hardware configuration for a virtual Alpha system. This script can be viewed and modified using any text editor. Manually modifying a script may introduce errors that prevent the emulator from launching. Existing configuration scripts are automatically parsed for manual modifications and errors by the Configuration Utility when opened.

Manually Modified Condifuration Files

There are cases where manually modifying a configuration file is required. Adding a 5-port serial card is one example. Entering debug options under the guidance of a Migration Specialties support specialist is another. The Configuration Utility detects configuration files that have been manually modified with options that are not accessible through the GUI when the file is open and issue a warning message. Using the Configuration Utility to update manually modified configurations is not advised.

Updating Avanti Path Names

When moving an Avanti configuration file to a new location, the path names embedded in the file may change. Using the Configuration Utility to update each path reference in

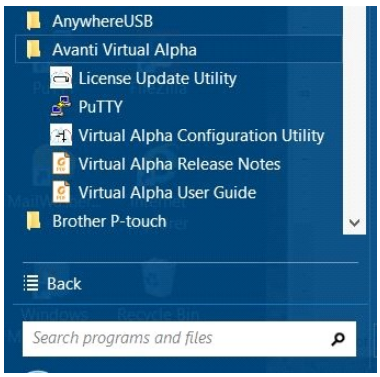
the configuration file can be a tedious process. The path names can be easily updated in one operation by opening the configuration file using a text utility such as Notepad and employing its Replace function. Manual modification such as path name changes are not flagged by the Configuration Utility GUI.

4.1.2 Specifying Sizes

Some configuration properties define element sizes. For example, a user may want to define a 4-gigabyte virtual disk. Avanti supports single character suffixes to indicate size magnitudes. To define a 4-gigabyte disk, the user would enter 4G. The following table lists accepted size suffixes.

Suffix	Size	Value
K	Kilo	1,024
M	Mega	1,024K
G	Giga	1,024M

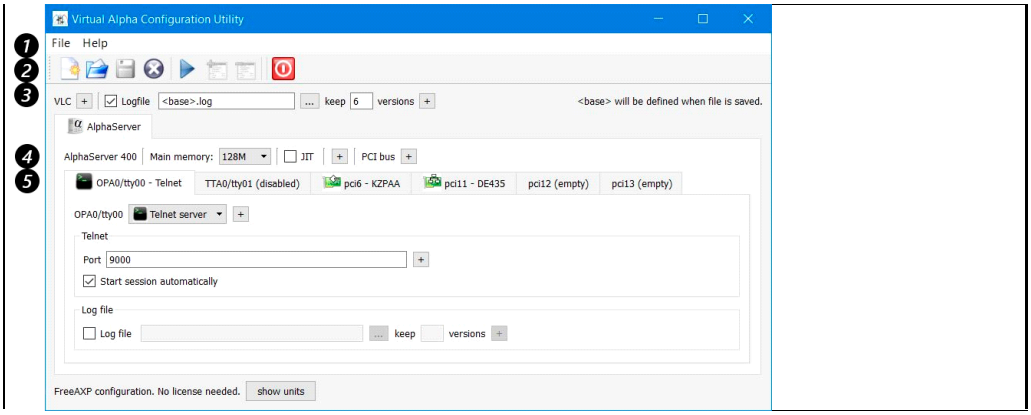
4.1.3 Running the Configuration Utility



The Avanti Configuration Utility is launched from Start> All Programs> Avanti Virtual Alpha> Virtual Alpha Configuration Utility.

Note: Administrative rights are required to run the Avanti configuration utility.

4.1.4 Configuration Utility Layout



1) Menu Bar containing drop-down menus

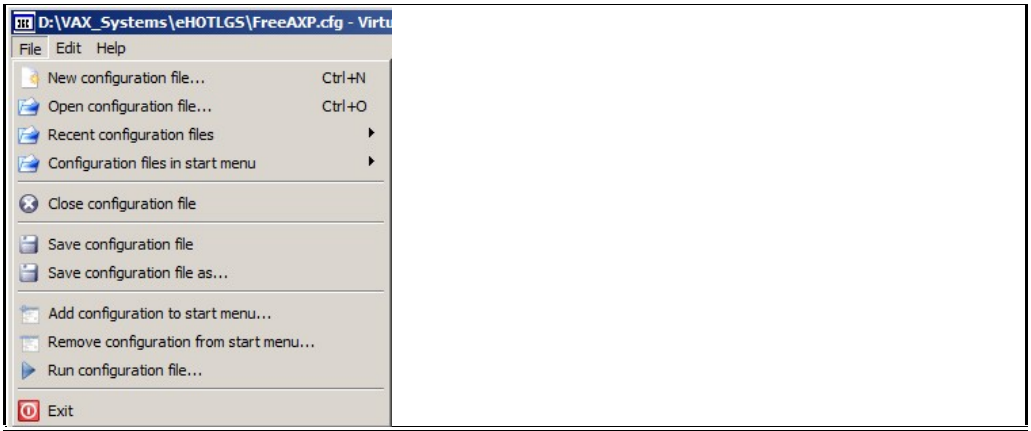
2) Tool Bar containing configuration shortcut buttons. Hovering over a button will display a pop-up description. The tool bar can be removed by right clicking in the bar and unchecking the toolbar box.

3) VLC Section provides access to basic emulator parameters.

4) Virtual Alpha Base Settings covering memory, JIT, NVRAM, and TOY settings.

5) Virtual Alpha Component Tabs defining PCI bus components such as serial ports, disk controllers, and NICs.

4.1.5 File Menu



The File menu provides functions to create, edit, and save virtual Alpha configurations.

Note:When selecting a function, if a configuration file is already open and has been modified, the utility will prompt to save the current file before executing the selected function.

New configuration file...
Opens a new configuration. The configuration utility will show a new configuration with default settings and no disk assignments.

Open configuration file...
Opens an existing configuration file. A file selection box will open to allow selection of the desired configuration file.

Recent configuration files
Opens a recently edited configuration file. A selection list appears of recently edited configuration files to choose from.

Configuration files in start menu
Presents a list of virtual Alpha configurations that have been saved to the Start menu. If a configuration is clicked, the associated configuration file is opened in the Configuration Utility.

Close configuration file
Closes the present configuration file.

Save configuration file
Saves the current configuration file. If the file is new, the utility will prompt for a file name.

Versioning: By default, Avanti retains 4 versions of each configuration file. If a configuration file is present when a Save is executed, it is renamed <base>.cfg;-1. On a subsequent save, it would become <base>.log;-2, and so on.

Save configuration file as...
Opens a file browser window to save the configuration file under a new name.

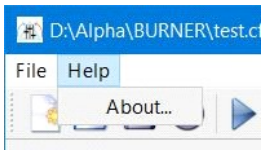
Add configuration to start menu...
Adds the current configuration file as a shortcut to the Start menu. Avanti shortcuts are stored under Start> All Programs> MSI Digital Emulator Collection.

Remove configuration from start menu...
If present, all shortcuts associated with the current configuration file are removed from the Start menu.

Run configuration file...
Runs the current configuration file.

Exit
Exits the Configuration Utility.

4.1.6 Help Menu



About...
Displays product, copyright, and version information.

4.1.7 Virtual Alpha Components

A virtual Alpha is constructed of virtual hardware components. The Configuration Utility provides the means to define the components required to build a working system. A basic system would include memory, a console, and a disk controller. If network connectivity is desired, a NIC is required.

4.1.7.1 Advanced Features

The [+] buttons seen on various configuration screens provide access to advanced Avanti features. Advanced features are normally not utilized in Avanti configurations. 4.1.8 Advanced Configuration Features 41 describes advanced features in details.

4.1.7.2 <base>

<base> will show up in the log file, TOY, and NVRAM file name fields. <base> is comprised of the path and file name under which an Avanti configuration file is saved. For example, if a configuration file is saved as D:\Avanti\Alpha01.cfg, <base> will equate to D:\Avanti\Alpha01. This information is displayed on the far-right side of the configuration screen on the VLC line.

When a new configuration is created, <base> is not assigned a value until the configuration file is saved. Until the configuration file is saved, the <base> definition area will show the following text:

```
<base> will be defined when file is saved.
```

4.1.7.3 VLC

The VLC line defines basic emulator parameters.

Logfile
Checking the Logfile box enables emulator logging. By default, the log file name will be <base>.log. The user has the option to modify the file name and location via the file name field and browse [...] button.

Keep
By default, Avanti retains 6 log files. If a log file is present when Avanti is launched, it is renamed <base>.log;-1. On a subsequent Avanti launch, it would become <base>.log;-2, and so on. The user can increase or decrease the number of log files retained using the keep field.

4.1.7.4 Virtual Alpha Base Settings

Virtual Alpha Base Settings covering memory, JIT, Auto-Exit NVRAM, and TOY settings.

Main Memory

Main memory defines the amount of memory assigned to the virtual Alpha system. Available memory is selected from a drop-down list.

Avanti supports 32MB to 3GB of memory. FreeAXP supports 32MB to 128MB of memory. Memory is mapped to match the three memory banks available in a real AlphaServer 400. The user need only specify the total memory desired in the Memory field. The following table describes the legal memory values for Avanti and depicts how they are calculated. It includes the license units required to support added memory beyond 128MB under AvantiFlex.

Total Memory	Bank 1	Bank 2	Bank3	FreeAXP	AvantiFlex	Avanti
32MB	16MB	16MB	-	ü	ü	ü
48MB	16MB	16MB	16MB	ü	ü	ü
64MB	32MB	32MB	-	ü	ü	ü
96MB	32MB	32MB	32MB	ü	ü	ü
128MB	64MB	64MB	-	ü	ü	ü
192MB	64MB	64MB	64MB	N/A	10 units	ü
256MB	128MB	128MB	-	N/A	10 units	ü
384MB	128MB	128MB	128MB	N/A	10 units	ü

512MB	256MB	256MB	-	N/A	10 units	ü
768MB	256MB	256MB	256MB	N/A	20 units	ü
1024MB (1.0G)	512MB	512MB	-	N/A	20 units	ü
1536MB (1.5G)	512MB	512MB	512MB	N/A	30 units	ü
2048MB (2.0G)	1024MB	1024MB	-	N/A	40 units	ü
3072MB (3.0G)	1024MB	1024MB	1024MB	N/A	60 units	ü

Attempting to run FreeAXP with more than 128MB of memory assigned will result in a license error.

JIT

Checking the JIT box enables Just-In-Time compilation. JIT is a performance enhancement feature that is not available to FreeAXP users. Attempting to run FreeAXP with JIT enabled will result in an error.

Auto-Exit

If checked, upon a normal OpenVMS or Tru64 UNIX shutdown, the emulator will terminate completely instead of stopping at the firmware prompt (>>>). With Auto-Exit set, the console prompt can still be reached using Ctrl-P or the Tru64 UNIX init 0 command.

Versions of Avanti prior to version 4.0.0 do not support the Auto-Exit feature. Remove Auto-Exit from the Avanti configuration before running an older Avanti version.

Auto-Idle

By default, Avanti utilizes one host CPU running at 100% to emulate the Avanti Alpha CPU.

With the Auto-Idle option selected, Avanti will idle the host CPU when the guest Tru64 UNIX or OpenVMS O/S is not busy.

Auto-Idle is not enabled by default. The Auto-Idle check box must be selected in the Avanti configuration file to enable the Auto-Idle feature

The Charon Idle and HP Idle utilities, available under OpenVMS only, provide similar functionality. These utilities work under Avanti and can continue to be used. Auto-Idle will not function if the Charon or HP Idle utilities are installed.

Note: Using Auto-Idle on I/O intensive applications will reduce application performance. Migration Specialties recommends disabling CPU idle features to achieve maximum performance.

4.1.7.4.1 Removing Charon and HP Idle Utilities (OpenVMS Only)

The Charon or HP Idle utilities can be removed using the following steps.

1. Verify the Idle Utility is present.

```
$ PRODUCT SHOW PRODUCT IDLE
```

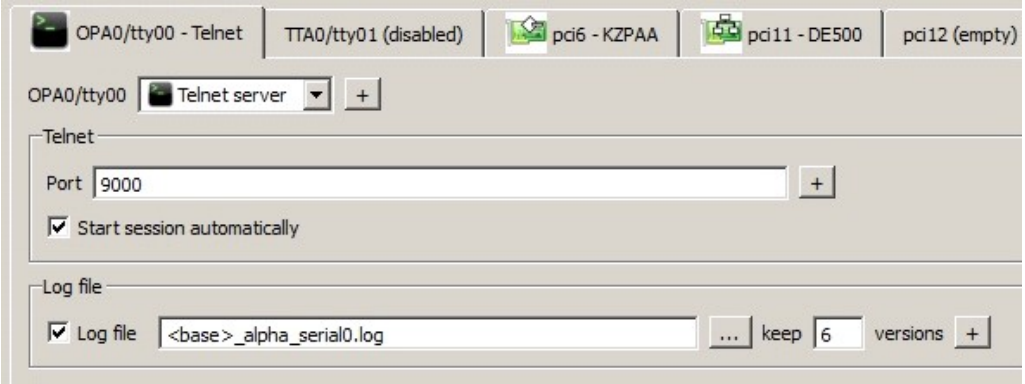
2. Remove the utility with the following command.

```
$ PRODUCT REMOVE IDLE
```

3. Reset the SYSGEN parameter CPU_POWER_MGMT.

```
$ MC SYSGEN
SYSGEN> USE CURRENT
SYSGEN> SET CPU_POWER_MGMT 0
SYSGEN> WRITE CURRENT
SYSGEN> EXIT
```

4.1.7.5 Serial Ports: OPA0/tty00 – TTA0/tty01



Avanti supports two serial ports. The OPA0/tty00 – TTA0/tty01 tabs provide the means to define each port. A drop-down box allows the user to define a virtual serial port via Telnet server or a physical serial port via Physical port. The ports will appear as OPA0 and TTA0 on OpenVMS systems and tty00 and tty01 on Tru64 UNIX systems.

Tip:Migration Specialties recommends enabling both virtual serial ports when building, modifying, or troubleshooting an Avanti configuration. Having two serial access points to a virtual Alpha system is useful in case one session locks up or network problems block standard telnet access.

Note:Baud rate mismatches between the Avanti serial port setting and the serial device can result in a hung serial port.

Log File

Checking the Log file box enables serial port logging. Data appearing on the serial port will also be written to an ASCII log file. By default, the log file name is <base>_VirtualAlphaName_serial#.log.

VirtualAlphaName: The name of the virtual Alpha definition as it appears on the definition tab.

serial#: serial0 or serial1, depending upon the serial port being logged.

Note:Caution should be used when logging serial port activity. A heavily used port will produce a large log file, which may impact disk resources on the host system.

Keep

By default, Avanti retains six serial log files. If a serial log file is present when Avanti is launched, it is renamed <base>_VirtualAlphaName_serial#.log;-1. On a subsequent Avanti launch, it would become <base>_VirtualAlphaName_serial#.log;-2, and so on.

The user can increase or decrease the number of serial log files retained using the keep field.

4.1.7.5.1 Telnet Server

Selecting the Telnet server option allows the user to define a virtual serial port.

Port

Port defines the telnet port number assigned to a virtual serial port. The port can be accessed via telnet as localhost:port# or from the network as Windows-ip-address:port# By default, telnet port assignments are made starting with 9000. The user can override default port settings via the Port field.

Tip: If running multiple instances of Avanti on a single host, don't forget to assign each virtual serial port a unique port number.

Avanti 4.0 restricts port access to the local host only. This is accomplished by only binding to the 127.0.0.1 interface, rather than all available interfaces. It means that outside of the localhost, it appears that the port has not been opened at all.

The port can be open for network wide access by enabling the allow_remote property under each serial ports Advanced properties settings. Allow network access to the ports is a security issue, particularly with the console port (OPA0/tty00).

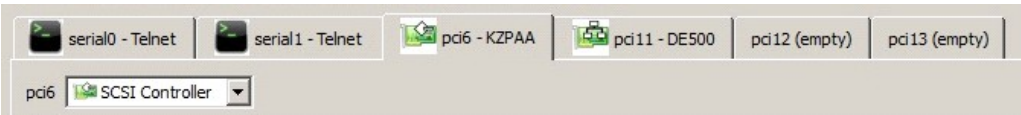
Start Session

Checking the Start session automatically box will automatically launch the virtual serial port when the virtual Alpha is started.

4.1.7.5.2 Physical Port

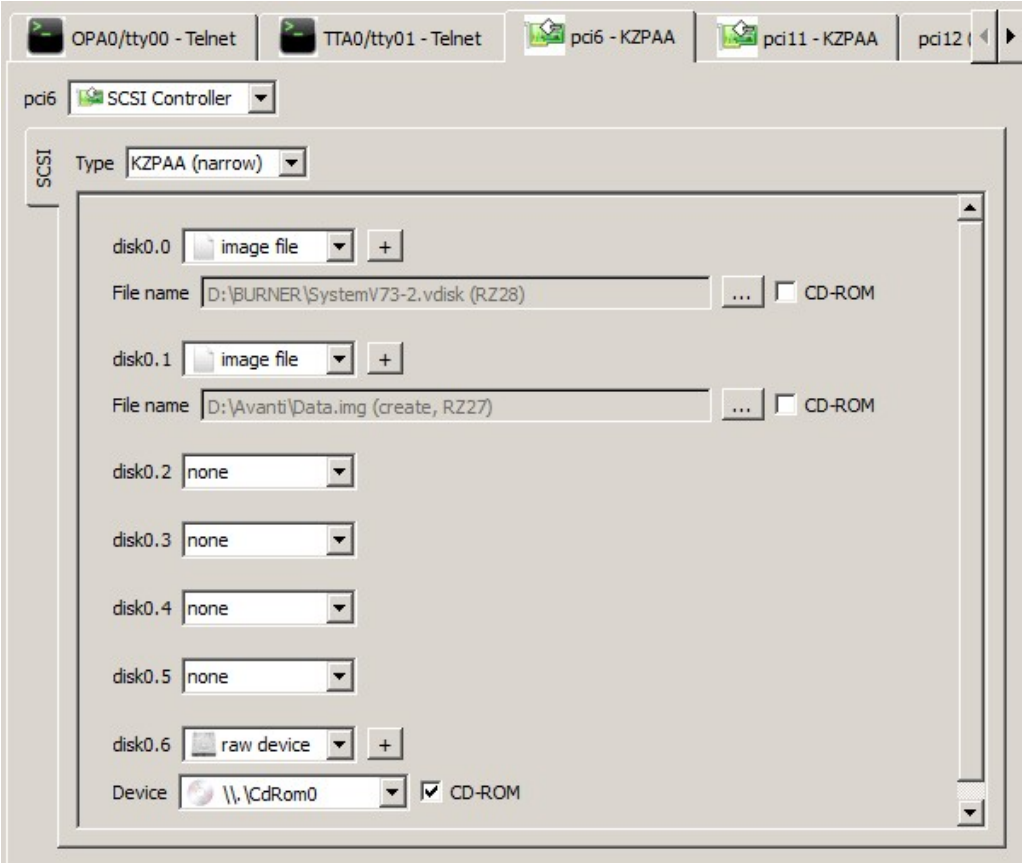
Selecting the Physical port option permits a physical COM port on the host server to be assigned to a virtual Alpha system. FreeAXP does not support physical ports.

4.1.7.6 PCI Slots



Avanti supports up to four PCI slots. The slots appear as tabs in the Configuration Utility. The PCI tabs are labeled pci6, pci11, pci12, and pci13, which are the same PCI slot id's in use on a real AlphaServer 400. On the real server, pci6 is reserved for the SCSI controller integrated on the main board. Each PCI slot can be assigned a SCSI controller or a Network card via a drop-down menu at the top of the tab. Not all PCI slots need to be assigned to build a working virtual Alpha system.

4.1.7.6.1 SCSI Controllers: KZPAA and KZPCA



Avanti supports two types of SCSI controllers: the 7-disk KZPAA narrow controller and the 15-disk KZPCA wide controller. Up to three SCSI controllers can be defined in an Avanti configuration. FreeAXP supports a single SCSI controller.

Tip:OpenVMS prior to V7.1 and Tru64 UNIX prior to V4.0F do not support the KZPCA wide controller.

Type

Type allows the selection of the KZPAA (narrow) or KZPCA (wide) controller via a drop-down menu.

Disk0.n

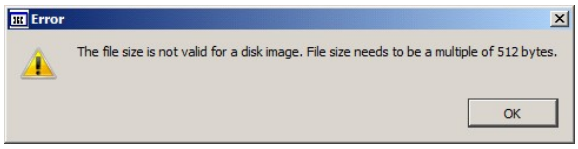
Disk0.n displays the disk image file or raw device associated with a disk on the virtual Alpha. For example, the picture above depicts the following disk definitions.

Definition	Virtual Alpha Device	Image file or Raw Device	File or Device String
disk0.0	DKA0	Image	D:\Avanti\System.img
disk0.1	DKA100	Image	D:\Avanti\Data.img
disk0.6	DKA600	Raw	\\CdRom0

Note: When using the KZPCA wide controller, disk numbering jumps from disk0.6 to disk0.8. This is normal and is due to the SCSI controller itself occupying slot 7.

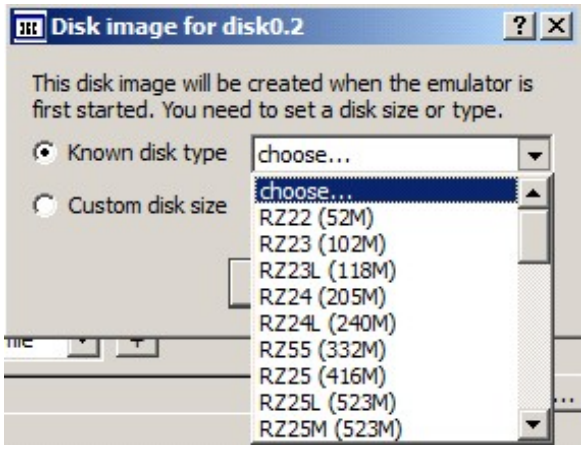
4.1.7.6.1.1 Image File

Image files are Windows container files that encompass the image of Alpha disks. These show up as virtual disks on the virtual Alpha system. Image files can be created by Avanti, copied from another Alpha emulator, or ported from a real Alpha system. By default, Avanti assigns a .img extension to disk image files.



Note: Image files need to be sized in multiples of 512 bytes to ensure they are properly recognized by Avanti. The configuration utility will display an error and will not map an image file with an invalid size.

To assign an existing disk container file to a virtual disk, select image file in an unused disk location and click the browse button [...]. Navigate to the desired host system folder and select the desired disk image file.



Avanti can create new image files, essentially creating new virtual disks, on startup. To create a new virtual disk, select image file in an unused disk location and click the browse button [...]. Navigate to the desired host system folder and type in the new disk file name. A disk type selection box will be displayed, as shown here. If a custom disk size is required, click the Custom disk size button and enter the desired size in Megabytes (M) or Gigabytes (G).

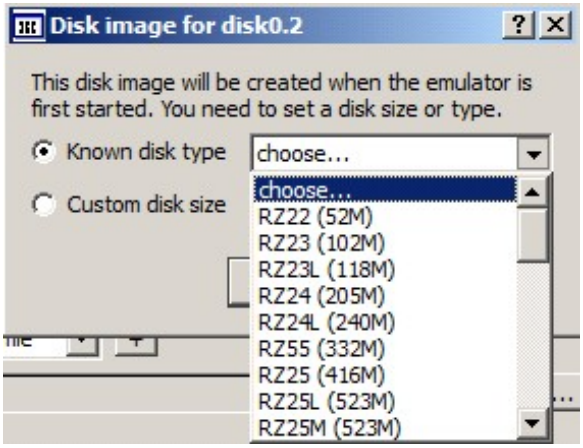
4.1.7.6.1.2 CD-ROM

Checking the CD-ROM box will define the device as a CD-ROM. Image files with an .iso extension are automatically assumed to be CD-ROM devices. FreeAXP is limited to one raw CD-ROM or DVD-ROM device. CD-ROM devices are mounted read-only and can be shared between FreeAXP and Avanti instances.

4.1.7.6.1.3 Raw Device

Raw devices are defined using the Windows raw device name. Available host system raw devices are listed in the Device drop-down box. FreeAXP does not support raw disk devices. FreeAXP supports one raw CD-ROM or DVD-ROM device.

4.1.7.6.1.4 Tape Image



Tape image files are Windows container files that contain the image of Alpha tapes. These show up as virtual tape devices on the virtual Alpha system. Tape image files can be created by Avanti, copied from another Alpha emulator, or ported from a real Alpha system. By default, Avanti assigns a .vtape extension to tape image files. FreeAXP does not support virtual tapes.

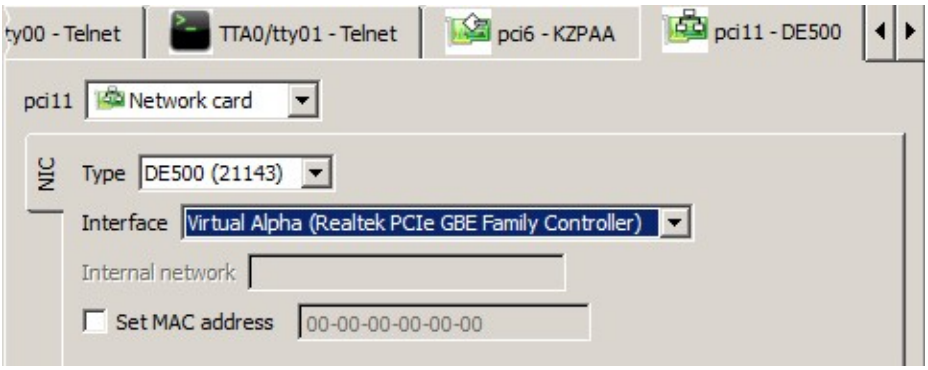
The vtape format utilized by Avanti is fully compatible with the vtape format used in SimH. Avanti virtual tape devices default to a DEC DLT2000 device type.

Note: Avanti puts no limit on the size of a vtape file. Continuously appending to a tape will eventually fill the host system disk.

To assign an existing tape file to a virtual tape device, select tape image in an unused disk location and click the browse button [...]. Navigate to the desired host system folder and select the desired tape image file.

Avanti will create a tape image file at startup if the file does not already exist. To create a new virtual tape, select tape image in an unused disk location and click the browse button [...]. Navigate to the desired host system folder and type in the new tape file name. When Avanti is started, an empty tape file will be created.

4.1.7.6.2 NIC



Avanti and AvantiFlex support up to three NICs. FreeAXP supports two NICs. NICs can be defined as the DE435 or the DE500 using the 21140 or 21143 chip sets. 6 Networking Avanti 66 discusses network configuration in more detail.

Tip: OpenVMS recognizes the DE435 and DE500 (21143) NIC types. The DE500 (21140) NIC type is only recognized by Tru64 UNIX.

Type

Type allows the user to choose between the following NIC types:

NIC	Recommended Usage
DE435	• OpenVMS version 6.1-1H1 and higher. • Tru64 UNIX version 3.2C and higher.
DE500 (21140)	• Tru64 UNIX version 4.0 up to 4.0E.
DE500 (21143)	• OpenVMS version 7.2 and higher. • Tru64 UNIX version 4.0F and higher.

Interface

Avanti provides NIC interface options in a drop-down list:

Dummy network: Defines a dummy NIC. The NIC will appear in the OpenVMS or Tru64 UNIX configuration, but will not be available to attached to an actual network. The Dummy network option acts as a place holder. It is useful in testing Avanti configurations prior to attaching them to a live network.

Host NIC: All available host NICs will be displayed in the drop-down list. Selecting a host NIC ties the NIC to the virtual Alpha system.

Note: If a host NIC is added after Avanti has been installed, it will not show up on the configuration utility pull down list. This is because the Avanti PCAP driver does not see the new device. To make the device visible to PCAP, open a cmd window, run "NET STOP MSIPCAP" followed by "NET START MSIPCAP" to restart the Avanti PCAP driver. Rebooting Windows will also correct the problem.

Set MAC Address

Checking the Set MAC address box enables the MAC address field, allowing the user to set a specific MAC address. Manually setting a MAC address is useful in the following situations:

- The generated MAC address causes a network conflict with another device.
- Software on the virtual Alpha system requires a specific MAC address.
- One host NIC is being shared between multiple Avanti instances2.
- When running Avanti under Windows hosted on a virtual machine such as VMware.

To avoid clashes of the assigned MAC addresses with those of other networking equipment or automatically generated virtual NIC MAC addresses, it is recommended that the assigned MAC addresses start with C6-00-00.

Note: The MAC address field will not accept an invalid MAC address. Trying to type or cut and paste an invalid MAC address will result in nothing being entered in the MAC address box.

4.1.7.6.2.1 Single Avanti Instance and MAC Addressing

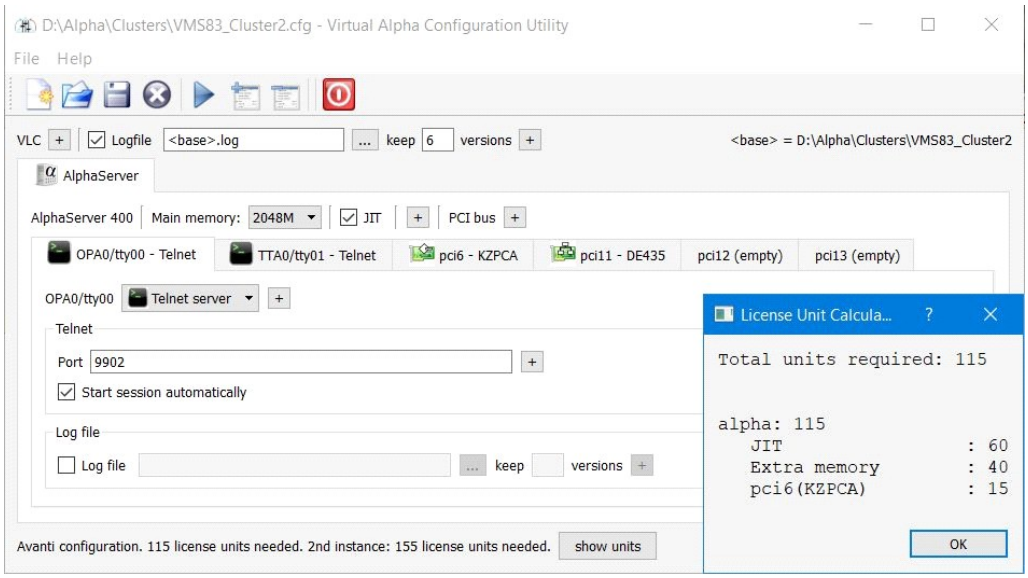
By default, Avanti assigns a unique MAC address to each virtual Alpha NIC in a single Avanti instance. The addressing scheme uses the host NIC MAC address as a base address to generate the virtual NIC addressing. As long as the generated MAC addresses do not conflict with another network device, there should be no need to manually configure MAC addressing.

4.1.7.6.2.2 Multiple Avanti Instances and MAC Addressing

If running multiple instances of Avanti, it is recommended that each instance be assigned its own host NIC. If this is the case, then the MAC addressing scheme described in 4.1.7.6.2.1 Single Avanti Instance and MAC Addressing 39 holds true and manual MAC assignments should not be necessary.

If multiple instances of Avanti share a single host NIC, then the MAC addressing scheme will assign each instance the same set of MAC addresses. This will lead to network address conflicts and block network access on all but the first instance. If a single host NIC is to be shared between Avanti instances, then manual configuration of the MAC addresses on each instance is required to ensure proper network functionality.

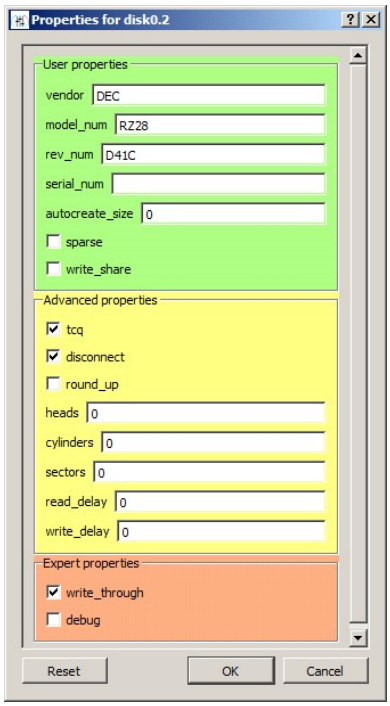
4.1.7.7 Show Units



Avanti and AvantiFlex configurations require a license key and license units to run. At the bottom of each virtual Alpha configuration window, the total number of license units required is displayed. Clicking the [show units] button provides a detailed list of the license units required.

The image above depicts an Avanti configuration that requires 115 license units. Note the summary of required license units at the bottom of the configuration window. Clicking the [show units] button displays the License Unit Calculation box which details the required license units.

4.1.8 Advanced Configuration Features

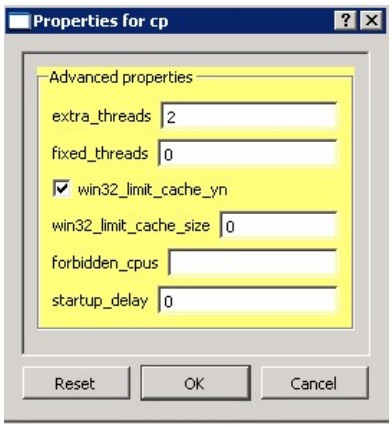


This section describes the advanced configuration features available via the advanced dialog boxes in the Configuration Utility. Advanced features are normally not utilized in Avanti configurations. They are available for special situations to tune features or troubleshoot problems.

Advanced dialog boxes are accessed by clicking the plus button next to specific configuration items. Items in each dialog box are labeled and color-coded as User properties (green), Advanced properties (yellow), and Expert properties (red).

User properties	Properties that can be safely modified and pose little danger to the virtual Alpha. These include things like log file names and terminal emulator launch strings.
Advanced properties	Advanced properties have the potential to adversely impact the virtual Alpha. These properties should only be modified by a knowledgeable user.
Expert properties	Expert properties will destabilize or disable the emulator if misused. Expert properties should only be modified under the guidance of Migration Specialties or a qualified support specialist.

4.1.8.1 VLC [+]



extra_threads (default: 2)
Defines the number of extra threads for the actor framework. The higher this number, the more I/O and device processes that can be active at the same time. A lower number may help overall I/O performance due to reduced overhead; a higher number may help prevent I/O stalls.

fixed_threads (default: 0)
Fixes the number the Theron threads used by the emulator. An entry of zero (0) allows the emulator to create as many Theron threads as required.

win32_limit_cache_yn (default: true)
When win32_limit_cache_yn is set to true, the emulator limits the Windows system file cache working set size. When the emulator performs heavy disk I/O's, the Windows file cache can grow rapidly. This can lead to insufficient remaining memory to satisfy other processes, including the emulator itself, which in turn leads to overall poor performance. Limiting the Windows system file cache working set size avoids this problem.

win32_limit_cache_size (default: 0, increment: 128M)
If win32_limit_cache_yn is set to true, the win32_limit_cache_size defines the limit to apply to Windows system file cache working set size. Setting the property to 0 defaults to 25% of physical memory.

forbidden_cpus
Reserves CPUs, blocking the emulator from using specific host CPU cores. Cores are numbered starting at zero (0). For example, a 4-core system has CPUs 0 – 3.

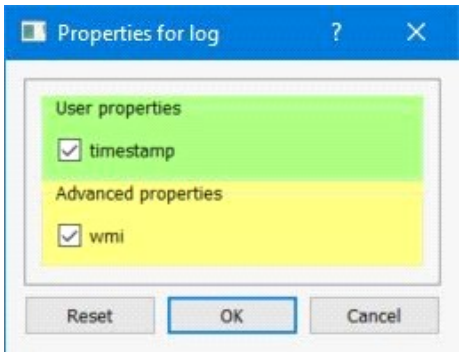
If blocking multiple CPUs, separate each CPU ID with a comma. An entry of 0,1 on a 4-core system would prevent Avanti and FreeAXP from using CPU cores 0 and 1, effectively treating the machine as a 2-core system.

Note: This is a global setting that applies to all Avanti instances that are running on the host machine. It takes its values from the first emulator that is started, so when running multiple Avanti instances it on a single host, it is best if the forbidden_cpus setting for each emulator is identical.

startup_delay (default: 0)
Time in seconds which the emulator will pause before starting.

use_npcap_driver (default: false)
By default, the Migration Specialties supplied msipcap NDIS network driver is used with Avanti. Checking the use_npcap option puts Avanti in automatic NDIS select mode. If a Winpcap compatible driver such as Nmap Project's npcap is installed, Avanti will default to it.

4.1.8.2 Logfile [+]



timestamp (default: true)
Inserts a time stamp at the beginning of each line in the Avanti log file. The timestamp is in the format YYYMMDDHHMMSS.mmm.

wmi (default: true)
When checked, wmi instructs the emulator to retrieve the Windows version, hardware platform, and related information and place it in the log file header. This information takes a small amount of time to gather, which may interfere with automated interaction with the emulator.

4.1.8.3 JIT [+]

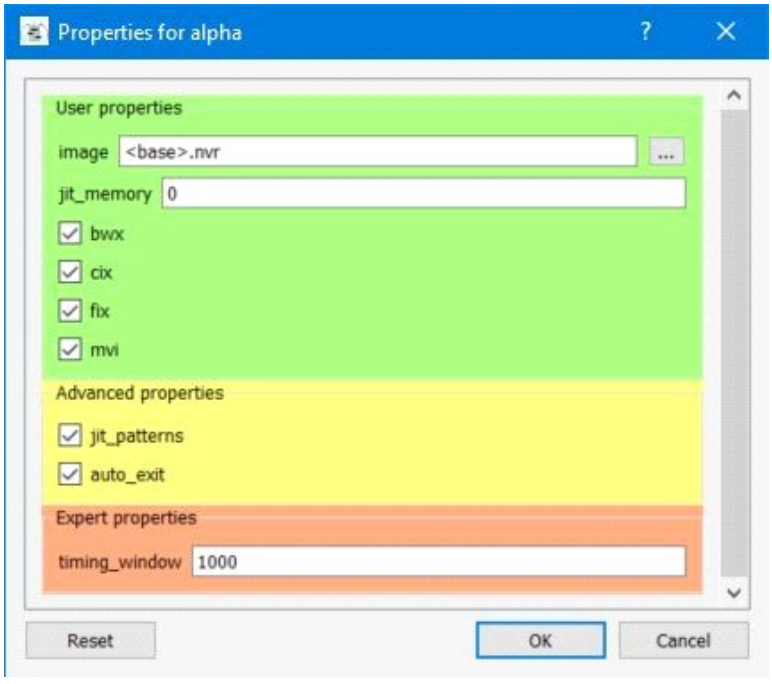


image (default: <base>.nvr)
Full path to the NVRAM image file. This file serves as the equivalent of non-volatile RAM on a real Alpha, storing configuration settings such as boot parameters.

jit_memory (default: 0, increment: 64M, maximum: 2G)
Maximum amount of memory to use for JIT. Specifying 0 enables dynamic allocation, which will utilize up to half of physical memory or 2GB, whichever is less.

Note: If running on a Windows host system equipped with more than 32GB of memory and the following error is encountered, set the jit_memory parameter to 2GB.

Unable to reserve JIT buffer area: error 87: The parameter is incorrect.

bwx (default: true)
Enables the post-EV4 Alpha processor Byte-Word eXtension.

cix (default: true)
Enables the post-EV4 Alpha processor Count Instructions eXtension.

fix (default: true)
Enables the post-EV4 Alpha processor Floating-Point Instructions eXtension.

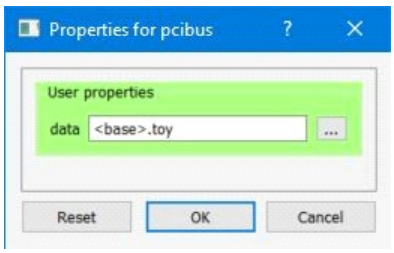
mvi (default: true)
Enables the post-EV4 Alpha processor Motion Video Instructions.

jit_patterns (default: true)
Enables pattern recognition features in the Avanti JIT engine, which replaces often used instruction sequences as a block rather than JIT each instruction individually.

auto_exit (default: false)
If checked, upon a normal OpenVMS or Tru64 UNIX shutdown, the emulator will terminate completely instead of stopping at the firmware prompt (>>>).

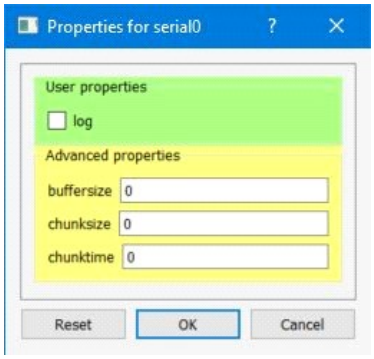
timing_window (default: 1000)
The timing window can be used to adjust the Avanti algorithm that tracks time within the emulator.

4.1.8.4 PCI bus [+]



data (default: <base>.toy)
Full path to the TOY image file. This file replaces the Time-Of-Year RAM on a legacy Alpha, storing configuration settings such as Time-of-Year and the time offset between the host system and the virtual Alpha.

4.1.8.5 Serial Ports (OPA0/tty00, TTA0/tty01) [+]



Avanti provides synchronous serial port support. With synchronous serial port I/O, a buffer normally isn't necessary and leads to unnecessary typing delays. Hence, buffer defaults are set to 0.

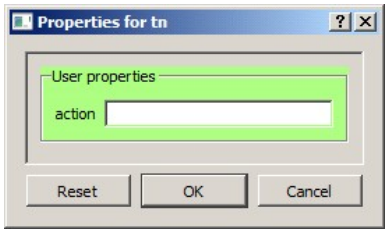
log (default: false)
Send serial port output to the main emulator log file. This helps correlate what is shown in the emulator log file to what appears on the serial port. Recommended for serial ports with a small volume of traffic, such as OPCOM messages. Not recommended for serial ports used for interactive sessions.

buffersize (default: 0)
Size of the buffer used to store incoming data.

chunksize (default: 0)
Maximum number of characters to allow in on the serial port per chunk.

chunktime (default: 0)
Time in milliseconds to wait between incoming chunks of data.

4.1.8.5.1 Telnet Port [+]



action
The Windows Command Line string needed to start a terminal emulator. The command line should include the telnet port number. Place an entry in the action box to specify a terminal emulator other than PuTTY

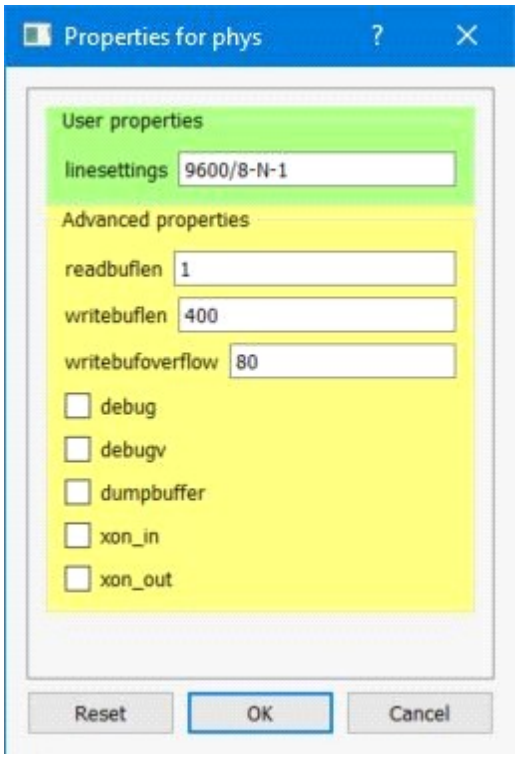
Tip: If running multiple instances of Avanti on a single host, don't forget to assign each virtual serial port a unique port number.

If the action box is left blank and the Start session automatically box is checked, a PuTTY string similar to the following will be automatically entered into the action box:

"C:\Program Files\Common Files\MigrationSpecialties Shared\putty.exe" telnet://127.0.0.1:9000

If a user-defined command string is placed in the action box, the Start session automatically box is automatically checked.

4.1.8.5.2 Device (Physical serial port) [+]



linesettings (default: 9600/8-N-1)
Physical serial or COM port setting.

Format: baudrate/databits-parity-stopbits or "host" for host settings.

Baudrates: 110, 300, 600, 1200, 2400, 4800, 9600, 11440, 19200, 38400, 57600, 115200, 128000, 256000.

Note: Baud rate mismatches between the Avanti serial port setting and the serial device can result in a hung serial port.

Databits: 5, 6, 7, 8

Parity: E - even, M - mark, N - none, O - odd, S - space.

Stopbits: 1, 1.5, 2

readbuflen (default: 1, maximum: 15)
Number of characters to attempt to read from the physical serial port in one operation. A higher number might work better with some serial ports if that data comes in bursts. On other serial ports, this fails. A value of 1 is always safe.

debug (default: false)
Send debugging information for the physical serial port to the log file.

debugv (default: false)
Send verbose debugging information for the physical serial port to the log file.

dumpbuffer (default: false)
Send buffer contents for the physical serial port to the log file for debugging.

xon_in (default: false)
If checked, turns on xon input control for serial port.

xon_out (default: false)
If checked, turns on xon output control for serial port.

4.1.8.6 SCSI Disk [+]

Specific disk information can be defined for emulated disks under Avanti. This is particularly useful to support older versions of Tru64 UNIX.

vendor
Disk vendor name.

model_num
Disk model name.

rev_num
Disk firmware revision.

write_share (default: false)
Make the emulated disk shareable with other Avanti instances. This feature is available to support shared emulated disks on an OpenVMS cluster.

Warning: Using the write_share feature to share disks with non-clustered Avanti instances is not recommended. Doing so may result in disk corruption.

serial_num
Disk serial number. If left blank, Avanti will generate a unique serial number for each disk and CD device.

Note: OpenVMS, Tru64 UNIX V3.x, and Tru64 UNIX V4.x ignore serial number settings. Tru64 UNIX V5.x requires unique serial numbers for all disk and CD devices.

autocreate_size (default: 0)
If the disk container file doesn't exist when the emulator is started, it will be created with this size. Use K to designate kilobytes, M to designate megabytes, and G to designate gigabytes. 4G would specify creating a disk image file 4 gigabytes in size.

sparse (default: false)
Determines whether the file will be created as a sparse file. A sparse file uses less space on disk initially and grows as data is written to the disk container file. A sparse file is created faster than a normal file, but write operations may take longer.

tcq (default: false)
Set the emulated disk to support tagged command queuing (TCQ) so the disk can be used as a quorum disk in an OpenVMS cluster.

Warning: Having TCQ enabled during a Tru64 UNIX installation may interfere with the installation process. It is recommended that TCQ be disabled during an install.

disconnect (default: true)
Let the emulated disk disconnect from the SCSI bus on large I/Os. Requires sync_io to be false. Disconnecting is normal SCSI behavior that allows other devices on the SCSI bus to be active while an I/O for this disk is in progress.

sync_io (default: false)
Perform all I/O's synchronously. Asynchronous I/O's provide the ability to have more than one I/O active at the same time.

round_up (default: false)
By default, the size of a disk container file needs to be a multiple of 512 (the sector size). When round_up is set to "true", this is no longer necessary and the disk image is padded with zeroes to the next biggest multiple of 512 bytes. This feature can be used to transfer files from the guest operating system to the host operating system.

heads (default: 0)
Number of read/write heads on drive.

cylinders (default: 0)
Number of cylinders on drive.

sectors (default: 0)
Number of sectors per track.

read_delay (default: 0)
Delay disk reads by specified number of microseconds. A 1 millisecond delay would be indicated by an entry of 1000.

write_delay (default: 0)
Delay disk writes by specified number of microseconds. A 1 millisecond delay would be indicated by an entry of 1000.

write_through (default: true)
Enable write-through caching. Changes are written directly to the raw disk or container file. If write through is disabled, there is a risk of losing data if the host system goes down unexpectedly.

debug (default: false)
Enable disk debugging. This is an internal use setting. An end-user would only set it if requested by Migration Specialties.

4.1.8.7 SCSI Tape [+]

Specific tape drive information can be defined for emulated tapes under Avanti. This is useful to support older operating systems, less common tape drives, or non-DEC tape drives.

vendor
Tape drive vendor name.

model_num
Tape drive model name.

rev_num
Tape drive firmware revision.

serial_num
Tape drive serial number.

advanced_behavior (default: false)
Adds device characteristics to device description returned to O/S. This feature is primarily useful with Tru64 UNIX installations.

disconnect (default: true)
Let the emulated tape drive disconnect from the SCSI bus on large I/Os. Requires sync_io to be false. Disconnecting is normal SCSI behavior that allows other devices on the SCSI bus to be active while an I/O for this disk is in progress.

read_delay (default: 0)
Delay tape reads by specified number of microseconds. A 1 millisecond delay would be indicated by an entry of 1000.

write_delay (default: 0)
Delay tape writes by specified number of microseconds. A 1 millisecond delay would be indicated by an entry of 1000.

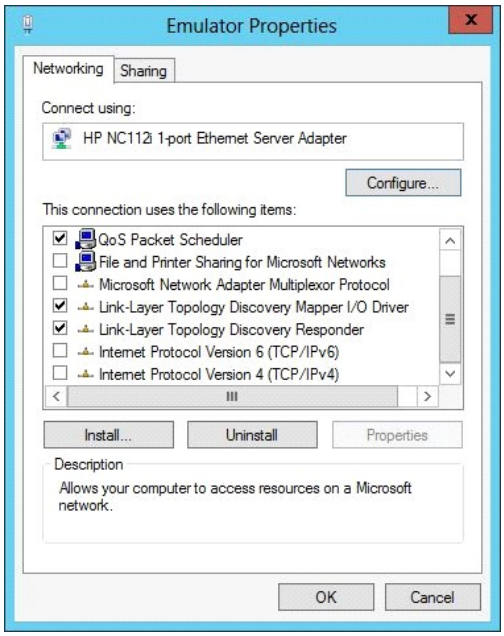
debug (default: false)
Enable tape drive debugging. This is an internal use setting. An end-user would only set it if requested by Migration Specialties.

Running Avanti

VIRTUAL ALPHA

5 Running Avanti

5.1 Host Network Configuration



On systems using wired NICs, open the Properties box of the NIC assigned to Avanti and uncheck all items assigned to the NIC. This will prevent Windows from trying to share the NIC with Avanti. This action is not recommended for virtual NICs hosted by VMware.

For users running Avanti on a system equipped with a wired and wireless NIC, such as a laptop, assigning Avanti the wired NIC will provide more stable network performance.

5.2 Avoiding Avanti Crashes

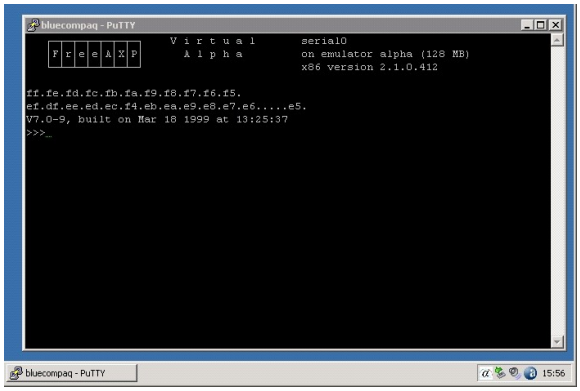
If a user logs out of Windows with a local Avanti instance running, or if Windows is shutdown with one or more Avanti instances running, the Avanti instances will crash. This is equivalent to pulling the plug on a physical Alpha system and crashing the OpenVMS or Tru64 UNIX operating system. All of the potential problems and caveats of an OpenVMS or Tru64 UNIX hardware crash apply when Avanti is terminated in this fashion.

Avanti Warning on Windows Shutdown

Avanti blocks Windows shutdown and sign-out to avoid inadvertently crashing an Avanti instance. If a local Avanti instance is running when a Windows user attempts to log out, or if Avanti is running anywhere when a Windows shutdown is issued, the logout/shutdown process will pause and a "The following applications block Windows shutdown" message is issued.

Warning: Terminating an Avanti instance without first shutting down the guest O/S is equivalent to pulling the plug on a legacy Alpha server.

5.3 Launching Avanti



Avanti can be launched interactively from the Configuration Utility using the Run button, or from a command line. Avanti can also be run as a Windows service, which is covered in 5.13 Installing Avanti as a Service 58. Launching Avanti is equivalent to powering up a real Alpha system.

The emulator program will display an Avanti icon in the notification area of the Windows task bar. It runs a series of checks on its virtual hardware, ensuring that all of its devices are present. If a disk image file has been misnamed or a NIC has changed, the emulator will display an error, wait for a user acknowledgement, and exit. If the configuration is valid, the emulator will launch a console window, load the SRM console firmware, and display the three-chevron console prompt (>>>).

Once the console prompt appears, the virtual Alpha is ready to boot a supported O/S. The two most common O/S run on Alpha systems are Tru64 UNIX and OpenVMS.

5.4 Avanti Control Icon

When Avanti is launched as a local process, an Avanti control icon is added to the notification area of the Windows task bar. Each instance of Avanti generates its own icon. Right clicking the Avanti icon display a control menu. The control menu provides the same halt and power off functions that a real Alpha system would offer via its front panel.

Note: No Avanti control icon is displayed when Avanti is run as a service.

Avanti Control Menu

First Line
Displays the name of the Avanti configuration file associated with the Avanti instance. It also shows the version of Avanti in use. Hovering over the Avanti icon will also provide this information via a tooltip.

Halt
Provides the same function at the Halt button on Alpha hardware. The emulator interrupts operating system execution and displays the console prompt (>>>) in the console window. The Halt option also writes debug information to the Avanti log and crash dump files, which is useful to isolate issues such as a hung emulator.

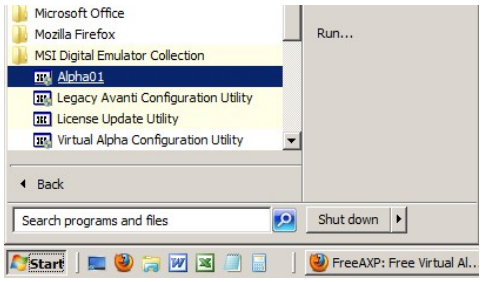
Power off
Provides the same function as the power button on Alpha hardware. When selected, a confirmation box is displayed. Using Power off or Exit from the console prompt will shutdown the Avanti instance and remove the related Avanti icon from the notification area.

Warning: Powering off a running Avanti instance is equivalent to crashing real hardware. Data loss may occur and the TOY and NVRAM settings may be lost.

About
Display information about the current version of Avanti.

Tip: During the Avanti installation process, the Avanti notification icon is marked as "Only show notifications". This is a Windows default. To have the Avanti icon always visible when an Avanti instance is running, right-click the Start button, select Properties, select the Taskbar tab, click the Customize button in the Notification area section, and select "Show icon and notifications" for the Avanti icon.

5.5 Launching Avanti from a Shortcut



Avanti is designed to be easily launched from Start Menu shortcuts created by the Avanti Configuration Utility using the File> Add configuration to start menu... option. See 4.1 Avanti Configuration Utility 28 for a description of the Configuration Utility. To launch an Avanti configuration, navigate to the MSI Digital Emulator Collection folder in the Start Menu and select the desired configuration.

Note: When accessing a physical device like a CD-ROM, Avanti needs to run from an account with administrator rights.

5.6 Launching Avanti Manually

Avanti virtual Alpha systems can be launched manually from a command window. The configuration file needs to be declared in the command line using the /CONFIGURATION_FILE= parameter. The following command line would launch the Alpha01 configuration file.

```
"C:\Program Files\MSI Digital Emulator Collection\dec.exe"
  /CONFIGURATION_FILE="C:\Alpha\Alpha01.cfg"
```

Note: The above command would appear on a single line.

Note the use of quotes in the command and parameter strings to ensure the file names are correctly passed to the emulator. This is particularly important when passing file names containing blanks.

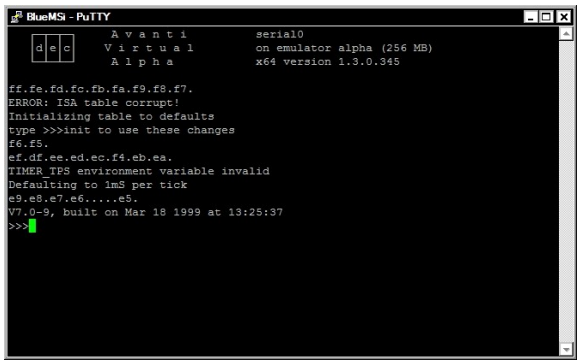
5.7 Launching Avanti from a Batch File

The following batch file would launch Avanti.

```
rem Alpha01.bat
rem Batch file to launch Avanti Alpha01 virtual Alpha configuration.

echo on
start "Avanti" "c:\Program Files\MSI Digital Emulator
Collection\dec.exe"
   /config="d:\VirtualAlpha\Alpha01.cfg"
exit
```

5.8 First Launch Messages



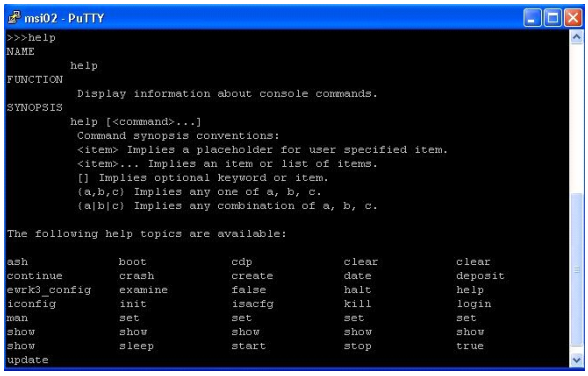
When a new configuration is launched for the first time, two error messages are displayed in the console window during the virtual hardware boot:

```
ERROR: ISA table corrupt!
Initializing table to defaults
type >>>init to use these changes

TIMER_TPS environment variable invalid
Defaulting to 1mS per tick
```

This behavior is similar to what happens on a real AlphaServer the first time it is powered on or when the CMOS battery is replaced. The Time-Of-Year (TOY) and NVRAM variables have not yet been initialized by the console firmware. Once the emulator has successfully booted to the console prompt and has then been shut down in a controlled fashion using the *exit* command, TOY and NVRAM information will be written to .toy and .nvr files. The next time the emulator boots, the required information will be extracted from these files and the above errors will not occur.

5.9 Console Operations



Once the system console prompt appears (>>>), operations proceed just as they would on a real AlphaServer 400. Avanti runs the actual AlphaServer 400 console firmware, providing all of its functionality and limitations. From this point forward, refer to the actual legacy hardware documentation for configuration and setup options. The hardware is virtual, but it acts like the real thing.

The Avanti console supports the same help functions as a real AlphaServer 400. Type Help to view available console help text.

5.10 Launching Avanti Without the Console

Like a real Alpha system, it is possible to launch Avanti without a console if the console boot parameters are properly configured.

Note: Launching Avanti as a service requires that firmware parameters be properly set and the console and TTA0/tty01 "Start session automatically" options be disabled. See 5.13 Installing Avanti as a Service 58 for details.

5.10.1 OpenVMS Auto Boot

Under OpenVMS, the virtual Alpha can be configured to boot automatically by setting the following console variables:

```
>>> SET AUTO_ACTION BOOT
>>> SET BOOT_OSFLAGS 0
>>> SET BOOTDEF_DEV ddua
```

Where ddua is a disk device, such as DKA0.

5.10.2 Tru64 Auto Boot

Under Tru64 UNIX, the virtual Alpha can be configured to boot automatically by setting the following console variables:

```
>>> SET AUTO_ACTION BOOT
>>> SET BOOTDEF_DEV DKxnnn
>>> SET BOOT_OSFLAGS A
```

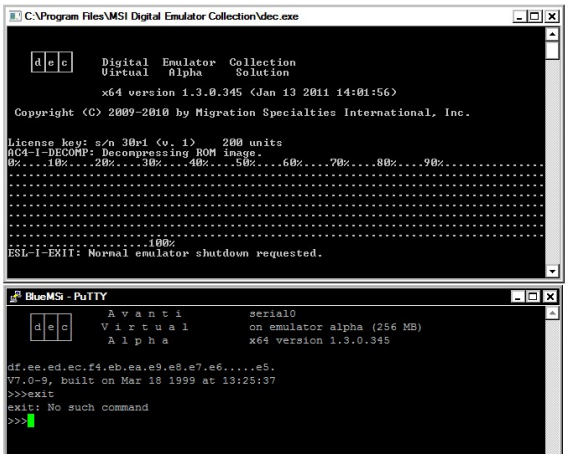
Note: By default, root login is only allowed via the console. For continued root access without a console, edit /etc/securettys to allow root access via the network.

5.10.3 Ambiguous Boot Device

The full device designation at the console level for a disk device is something like dka100.1.0.1.8. The Avanti firmware allows device names to be abbreviated as long as they remain unique. For example, both boot dka200 and boot dka2 would be valid commands. If a device name is not unique, a boot or set bootdev_dev will generate a device ddvu is ambiguous warning. This problem is likely to be encountered when trying to set DKA100, DKB100, or DKC100 as the boot device on a wide controller where DKA1000, DKB1000, or DKC1000 is also present.

To make DKA100 a unique device name, add a dot(.). Thus, "boot dka100" becomes "boot dka100.".

5.11 Exiting Avanti



A normal O/S shutdown under Avanti will return control to the system console, just like on real Alpha hardware. Under OpenVMS, <Ctrl^P> entered from the console window will also return control to the console. To exit Avanti from the console in a controlled fashion, type exit at the console prompt. The message exit: No such command will appear in the console window and Avanti will shut down. The TOY and NVRAM settings for the configuration will be saved.

5.11.1 Avanti Auto Exit

Avanti can be configured to auto exit by setting the auto_exit parameter in the configuration files. See 4.1.7.4 Virtual Alpha Base Settings 32 for details.

5.12 Multiple Avanti Instances on a Single Host

Avanti supports multiple instances running on a single host system. Each Avanti instance requires the following:

- One host CPU
- A unique configuration file
- Sufficient license units

Avanti requires that one host CPU be dedicated to the host O/S. Thus, a 4-core system could conceivably support three instances of Avanti. Use of hyperthreading is not recommended, as it may introduce instability to the emulator.

Running multiple virtual Alphas under Avanti requires robust host hardware. Sufficient CPUs, additional memory, and a fast I/O bus will be required to achieve adequate performance. The following table provides minimum host system configuration guidelines.

CPUs	3/2: 3 host CPU per 2 Avanti instances
Memory	4GB for Windows + 2GB for JIT + 2 * Avanti Memory. See 3.1.2.3.1 Sizing Host Memory 19 for additional information.
Disks	• One disk dedicated to Host O/S. • At least one disk per Avanti instance.

5.12.1 Overloading a Host System with Multiple Avanti Instances

As a rule, two host CPUs per Avanti instance are recommended. This ensures sufficient host system resources are available to handle peak application loads on the emulators, particularly I/O loads. If sufficient CPUs are not available to the host, it will begin to utilize one or more CPUs dedicated to the Avanti instances. This will adversely impact overall performance of the Avanti instances. Implementing Avanti instances on a host system at a greater than 1 – 2 ratio of Avanti to host CPUs is a process that should be done carefully and monitored closely.

5.12.2 Avanti and FreeAXP on a Single Host

Avanti and FreeAXP can be run simultaneously on a single host system provided sufficient host resources are available.

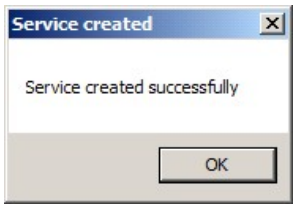
5.13 Installing Avanti as a Service

Avanti can be installed as a service to automatically start when the host system boots. When installed as a service, Avanti enables the Automatic Delayed Start option by default on Windows 7 and higher.

When installing Avanti as a service, considering the following:

- The Avanti configuration must be set up to auto boot. See 5.10 Launching Avanti Without the Console 56 for details on auto booting.
- No control icon is displayed when Avanti is run as a service. The Avanti service can be stopped and started from the Windows Services management console.
- Stopping the Avanti service crashes the virtual Alpha system. Always halt the Avanti guest O/S before stopping the Avanti service
- Disable serial telnet sessions (OPA0/tty00, TTA0/tty01) in the virtual Alpha configuration file by unchecking the Start session automatically box. When run as a service, the serial telnet session will not have a user session in which to start.
- Enable the Avanti auto_exit feature. See 4.1.7.4 Virtual Alpha Base Settings 32 for details.

5.13.1 Avanti Service Creation



To install Avanti as a service, execute the following command from an administrator Windows cmd session:

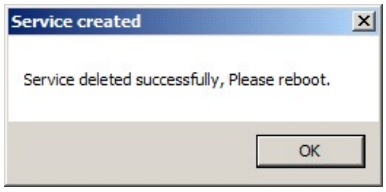
```
"C:\Program Files\MSI Digital Emulator
Collection\dec.exe" /CONFIGURATION=D:\vAlpha\Alpha1.cfg /INSTALL
```

Substitute your virtual Alpha configuration file path for D:\vAlpha\Alpha1.cfg.

A successful install will display a Service created successfully message. The /INSTALL qualifier will create a service entry named Avanti_Alpha1 set for automatic startup on the next system boot. The service can be started immediately by going to the Services manager (Start> Administrative Tools> Services), right clicking the Avanti entry, and selecting Start.

Multiple instances of Avanti can be installed as a service. Each instance must have a unique configuration file name and unique serial port number assignments (see 4.1.7.5 Serial Ports: OPA0/tty00 – TTA0/tty01 34).

5.13.2 Avanti Service Removal



To remove an Avanti instance running as a service, open a Windows cmd session in an administrator session. Execute the following command to remove the service entry:

```
"C:\Program Files\MSI Digital Emulator
Collection\dec.exe" /CONFIGURATION=D:\vAlpha\Alpha1.cfg /UNINSTALL
```

A successful uninstall will display a Service deleted successfully. The service will be marked for removal on the next Windows shutdown.

5.14 Virtual Host Environments

FreeAXP and Avanti will run under virtual hosts such as VMware, Hyper-V, KVM, XEN, and VirtualBox. Avanti supports dynamic VM environments such as VMware vMotion.

5.14.1 NIC Configuration

If DECnet support is desired under a virtual host, the NIC must be configured to support promiscuous mode. See 6 Networking Avanti 66 for additional information about networking Avanti.

5.14.1.1 VMware Virtual NICs

The following VMware virtual NIC devices are known to work well with Avanti:

VMware NIC Type	Guest O/S
E1000	VMS & Tru64 UNIX
VMXNET 3	VMS

5.14.2 Allocating CPU Resources

Avanti requires a minimum of 2 host server CPU cores. One CPU core is dedicated to emulating the Alpha CPU and will run at 100% when Avanti is active. Plan adequate VM resources to support this requirement.

It is important to allocate sufficient CPU resources on a VM to support Avanti. If a VM manager throttles CPU resources to a VM hosting Avanti, poor performance and slow clock speeds resulting in time drift will occur. Recommend best practice is to put a CPU reservation on the Avanti host VM such that two full CPUs are always available to the Avanti system. I.e.: if running on a VM host equipped with 3.0GHz CPUs, reserve 6000MHz of CPU to the host.

5.14.2.1 Allocation of Virtual Alpha CPUs

In Windows, a program cannot allocate or dedicate cores for execution; it is the responsibility of the operating system to allocate processes and threads across cores. Avanti requests that each simulated Alpha CPU thread be run on a separate physical core by setting the affinity of the virtual CPU thread to the desired core number and requesting that the simulated Alpha CPU thread run at highest priority. This should give the virtual Alpha CPU thread exclusive access to that core, provided Windows honors the request.

There are various other threads created by Avanti which take care of asynchronous IO to disks and Ethernet controllers, but these are requested at normal priority, and can be freely scheduled across available cores.

Note that Windows is free to ignore Avanti affinity and priority requests. Also note that if there are enough physical cores to run the desired workloads, system performance can be increased by disabling hyperthreading, which will give a speed boost to the physical cores since they no longer have to co-schedule the hyper-threaded virtual cores.

5.14.3 General VM Resource Management

Care must be taken not to overload the VM host hardware supporting the Avanti instance. An overloaded host will result in poor performance on the Avanti system which can lead to VMS bug checks.

5.15 Avanti Performance

No specific tuning is required to run OpenVMS or Tru64 UNIX on Avanti. Use of the fastest available host hardware will help improve overall performance. Using quality host hardware components will improve overall system reliability.

Alpha CPU's process commands sequentially. There is no known method of distributing CPU commands across multiple host CPUs that provides performance gains.

Utilizing a low-performance host system, overloading the amount of memory available to the virtual Alpha system, and excessive I/O will have negative performance consequences, just as on a real AlphaServer. Tuning the O/S may help rectify the situation. If not, consider using a more powerful host system or reducing the workload. If FreeAXP is in use, consider upgrading to Avanti with the JIT option.

A common problem when Avanti is hosted in a VM such as VMware is over commitment of the VM host hardware. I.e. Running to many VMs on a single host. Ensure adequate resources are allocated to the VM hosting Avanti.

5.15.1 CPU Instruction Sets

In addition to the standard Alpha EV4 instruction set, the Avanti CPU also supports the BWX, CIX, FIX, and MVI instruction set extenstions. These instructions were not natively supported on the EV4 CPU. They were introduced on the EV56, EV6, and EV68 series Alpha CPUs. Native EV4 systems running code generated for EV5 and EV6 processors would trap to the operating system and emulate these instructions. Adding the instructions to Avanti's emulated EV4 CPU allows the instructions to be processed "natively", resulting in improved performance.

For example, Python is compiled by default for the EV56 processor. Most of what the Python interpreter does is string manipulation and it uses the byte/word-extension instructions to do so. These instructions do not exist on an EV4 system, so each of these instructions takes a TRAP, firmware passes the trap to VMS, and VMS emulates the instructions. This is very slow, both on real and virtual EV4 hardware.

The addition of the EV56 instruction set to Avanti eliminates the EV4 performance issue.

5.15.2 Memory

Section 3.1.2.3.1 Sizing Host Memory 19 provides tips on ensuring adequate memory in the host system.

5.15.3 Disk I/O

Avanti allows simultaneous access against up to 45 virtual disks (FreeAXP supports seven disks). These disks can all exist as container files on a single physical disk on the host system. If an application running under Avanti puts high I/O loads on multiple virtual Alpha disks, the host system physical disk controller can be saturated, which in turn will adversely impact performance on Avanti and the host system. To alleviate this problem, spread virtual disks across multiple physical disks and controllers.

Note: Virtual disks that are hosted on a RAID controller should not have this issue.

5.15.4 Using CDs and DVDs

Avanti permits mounting and using real OpenVMS and Tru64 UNIX CDs and DVDs. However, better performance can be obtained by creating an ISO image of these devices. I/O on virtual CDs and DVDs is very fast, speeding up installations and data access.

ImgBurn, available at http://www.imgburn.com/, has proven a good tool to create Tru64 UNIX and VMS ISO images.

5.15.5 Virtual Tape Performance

When possible, it is recommended that virtual tape devices be placed on the KZPCA (wide) SCSI controller for better performance. OpenVMS versions prior to 7.1-2 and Tru64 UNIX versions prior to V4.0E do not support the KZPCA wide controller.

5.15.6 TCQ and Performance

Tagged command queuing's (TCQ) primary purpose is to enable quorum disk support in OpenVMS clusters. TCQ is disabled by default because it can interfere with the Tru64 UNIX installation process. In-house OpenVMS and Tru64 UNIX performance benchmarking indicates that enabling TCQ on all disks provides a small overall improvement in Avanti performance. Tru64 UNIX versions 4.n and 5.n along with all supported versions of OpenVMS see a benefit. If performance is critical, test enabling TCQ on all disks.

5.15.7 Floating Point Calculation Performance

Applications heavily dependent upon floating point calculations will not perform well on emulators such as Avanti. Floating point calculations are handled quickly in special hardware on real legacy Alpha systems, but must be emulated in software on virtual systems.

5.15.8 Hyperthreading

If there are enough physical cores to run the desired workloads, system performance can be increased by disabling hyper-threads, which will give a speed boost to the physical cores since they no longer have to co-schedule the hyper-threaded virtual cores.

5.15.9 Parking Cores

Parking cores is a feature that allows the BIOS or Windows decides not to run all of the cores in a system, usually for power, performance, or cooling reasons. Parking cores can impact Avanti performance, as it overrides Avanti's affinity requests. Ensure that there are no BIOS or Power Manager settings reducing power to the CPUs or forcing Turbo Boost mode to be enabled if maximum Avanti performance is desired.

5.15.10 Avanti Clock

A legacy Alpha system uses hardware to maintain accurate time. An Alpha emulator must achieve the same clock accuracy using software. Doing so is not a straight forward task.

On a real AlphaServer, two mechanisms are used to keep track of time. One is the clock interrupt, which interrupts the processor 1024 times per second, the other is the

cycle counter, which is an internal processor register that increments at the rate of the processor's clock (i.e. 100,000,000 times per second for a 100 MHz CPU).

There is no easy way to provide direct emulation of this functionality in the emulator. Windows does not offer a method to have a program interrupted at a steady rate of 1024 times per second. It would be feasible to have the emulator run a dedicated execution thread to generate the clock, but this would keep an additional CPU on the host system busy 100% of the time. So, in Avanti, the clock is generated on the one host CPU that is already 100% busy, the CPU that executes the Alpha CPU emulation thread.

If the emulator were to execute Alpha instructions at a steady rate, e.g. 50 million instructions per second, providing the timing function would be easy. The cycle counter could be incremented by 2 every instruction and trigger a clock interrupt every 48828 instructions. However, the speed of executing Alpha instructions on the emulator varies, so a mechanism is required to stabilize the clock. Avanti does not increment the cycle counter after every instruction, but after executing a number of instructions. The cycle counter increment amount is variable. When the emulator starts, it runs a short performance test to get an estimate of the host CPU's speed and uses the result to set a reasonable default value for the cycle counter increment.

To keep the clock in sync, a second thread wakes up periodically (about 100 times per second). When this second thread wakes up, it gets the current time from the Windows system and uses it to calculate what the value of the cycle counter, based on the current time, should be. If it finds that the cycle counter maintained by the CPU thread is off, it will apply a correction to the cycle counter increment to either slow down or speed up the cycle counter. This way, the cycle counter increments at a reasonably stable rate, with very little clock skew over a long period of time.

Finally, the clock interrupt is derived from the stabilized cycle counter. The CPU thread keeps track of the cycle counter value at which the next clock interrupt should be triggered.

The emulated clock accuracy is critical to precise Alpha emulation and has been tested extensively. Avanti exhibits no clock drift over extended run periods under varying load conditions.

Warning: The Avanti clock is dependent upon the Windows clock. Changing the Windows clock while Avanti is running has consequences. Windows clock changes will impact Avanti's perception of time until Avanti can re-sync with Windows. In short, do not change Windows time while Avanti is running. For more details, read on.

5.15.10.1 Avanti Clock Dependency

When VMS or Tru64 UNIX boot on a real system, it reads the current time from the battery-backed time-of-year clock chip. It also reads the time from the clock chip when a SET TIME operation is executed. It writes to the chip when SET TIME is used to change the system time. The rest of the time, VMS relies on getting a timer interrupt at regular intervals. VMS knows how many timer interrupts go into a second, and increments its own internal time accordingly.

As described in 5.15.10 Avanti Clock 62, on the emulator, we have to emulate both the battery backed clock chip and the timer interrupt.

For the clock chip, when it is written to, i.e. whenever VMS does a SET TIME=, the emulator records the difference between the time set and the current Windows time in the .TOY file. Whenever the chip is read from, the current time is obtained from

Windows and the saved difference applied. Thus, Avanti time is always an offset of Windows time.

For the timer interrupt, the Windows timer is not used because it is much to coarse-grained. Instead, Avanti estimates how many instructions the emulated CPU can execute in the time between two timer interrupts. It then uses a simple instruction counter in the CPU thread. After the CPU has executed that number of instructions, Avanti generates a timer interrupt. In the main emulator thread, Avanti periodically queries Windows time and use that information to adjust the information the CPU uses to keep track of time. If Avanti determines the emulated CPU has not generated enough interrupts, it lowers the counter threshold.

If Avanti sees it has generated too many interrupts, it raises the counter threshold. This mechanism depends on Windows time progressing normally. If one were to set the Windows time ahead by 4 years, the emulator would then believe that it has missed a massive number of interrupts and lower the threshold, telling the emulated CPU to start generating timer interrupts much more rapidly: i.e. Avanti time speeds up. Vice versa, if the time is set back 4 years, the emulator would believe it has generated too many interrupts and raise the threshold, making time slow to a virtual standstill: i.e. Avanti time slows down.

In short, avoid changing Windows time while Avanti is running. Doing so impacts Avanti time keeping and may have undesirable consequences in running applications.

5.15.11 Charon Idle Loop/Powersaver Driver

Note: Avanti's Auto-Idle feature provides similar power saving functionality under both OpenVMS and Tru64 UNIX. See 4.1.7.4 Virtual Alpha Base Settings 32 for details.

Alpha CPU's were designed to run at 100% all the time. They do not have a reduced power mode like more modern Integrity and X86 processors. The exception is the processor used in the Tadpole, a laptop version of the Alpha. A hook was added to the firmware to signal the processor when VMS was not busy and temporarily pause the CPU by putting it into a low power state.

SRI (now Stromasys) developed the Idle Loop to take advantage of the Tadpole hook for Charon-AXP. They now call the Idle Loop the Idle Powersaver Driver and they explicitly warn against using it in production environments. See this link:

https://stromasys.atlassian.net/wiki/spaces/DocCHAXPSMAv2240W/pages/34603068/Idle+Powersaver+Driver

Avanti also supports the Idle Loop software with all its caveats. We've found the Idle Loop can have a substantial performance impact on application with high I/O thru put and low CPU utilization. If your applications are I/O intensive, we recommend disabling the Idle Loop. The Idle Loop can be disabled using the following command from the OpenVMS SYSTEM account, followed by a system reboot:

```
$ PRODUCT REMOVE IDLE
```

The Idle Loop can also be enabled and disabled using the system parameter LOAD_SYS_IMAGES. The default is 7. Changing it to 6 (clearing bit 0 = SGN$V_LOAD_SYS_IMAGES) prevents SYS$IDLE from being loaded during boot. See 4.1.7.4.1 Removing Charon and HP Idle Utilities (OpenVMS Only) 33 for instruction on removing the Idle Loop completely.

Networking Avanti

VIRTUAL ALPHA

6 Networking Avanti

Avanti supports virtual versions of the DE435 and DE500 Network Interface Cards. Two versions of the DE500 utilizing the 21140 and 21143 chip sets are supported. The DE435 is the default NIC selected by the Configuration Utility when creating a new configuration.

The following table outlines recommended NIC usage based on the operating system being run by the virtual Alpha.

NIC	Recommended Usage
DE435	• OpenVMS version 6.1-1H1 and higher • Tru64 UNIX version 3.2C and higher
DE500 (21140)	• OpenVMS version 7.2 and higher • Tru64 UNIX version 4.0 up to 4.0E
DE500 (21143)	• OpenVMS version 7.2 and higher • Tru64 UNIX version 4.0F and higher

The DE435 is technically a 10Mb device, while the DE500 is capable of 100Mb. However, as virtual devices, all NIC variants offer the same level of performance. Actual virtual Alpha NIC performance is closely tied to the capabilities of the host NIC with which it is associated and to the software applications that use the NIC. An I/O or CPU bound application will be unable to take full advantage of a fast NIC.

6.1 PCAP Drivers

Avanti supplies its own NDIS5 network driver, msipcap. However, if the Nmap.org Npcap NDIS6 driver is installed on the Windows system, Avanti can be configured to default to it as its network driver. See 4.1.8.1 VLC [+] 42 for additional information.

6.2 Adding a Host NIC

If a host NIC is added after Avanti has been installed, it will not show up on the configuration utility pull down list. This is because the Avanti PCAP driver does not see the new NIC. To make the NIC visible to PCAP, open a cmd window, run “net stop msipcap” followed by “net start msipcap” to restart the Avanti PCAP driver. Re-installing Avanti and rebooting Windows will also correct the problem.

6.3 Dedicating a NIC

On dual NIC systems (this includes systems with a hardware NIC and a wireless NIC), it is best to disable TCP/IP services on the NIC assigned to Avanti. Prior to launching Avanti, open the Properties box of the NIC assigned to Avanti and uncheck all items assigned to the NIC. This will prevent Windows from trying to share the NIC with

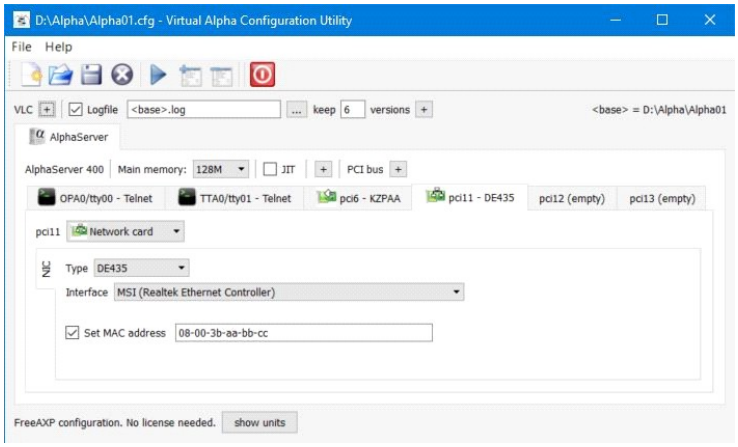
Avanti. For users running Avanti on a system equipped with both a wired and wireless NIC, assigning Avanti the wired NIC will provide more stable network performance.

6.4 NICs and Multiple Instances of Avanti

When running multiple Avanti instances on a single host machine, it is recommended that a physical NIC be dedicated to each virtual NIC. If multiple emulators share a single physical NIC performance degradation may occur. Should multiple Avanti instances share a single host NIC, then manually assigning MAC address to each Avanti NIC is required, as described in 4.1.7.6.2 NIC 38 and 4.1.7.6.2.2 Multiple Avanti Instances and MAC Addressing 40. To avoid clashes of the assigned MAC addresses with those of other networking equipment or automatically generated virtual NIC MAC addresses, it is recommended that the assigned MAC addresses start with C6-00-00.

Note: The MAC address field will not accept an invalid MAC address. Trying to type or cut and paste an invalid MAC address will result in nothing being entered in the MAC address box.

6.5 Manually Modifying the Avanti MAC Address



Avanti automatically assigns a unique MAC address to each virtual Alpha NIC. The assigned MAC address is calculated based off the MAC of the associated host NIC. In cases where a specific MAC address is desired, or when the generated MAC address causes a network conflict, a MAC address can be manually assigned via the Configuration Utility. Select the desired NIC definition in the Configuration Utility; check the Set MAC address box; and enter a valid, unique MAC address.

Warning: Do not set the LSB (multicast bit) of the first octet. Doing so makes the first number odd (ex. "09-00-2b-aa-bb-cc"), which violates the Ethernet MAC standard and will interfere with NIC functionality.

Note:The MAC address field will not accept an invalid MAC address. Trying to type or cut and paste an invalid MAC address will result in nothing being entered in the MAC address box.

6.6 DE435 for Older O/S

If you are experiencing network problems with older versions of OpenVMS or Tru64, ensure you are using the DE435. The DE435 offers better compatibility with older O/S. Use the DE435 with versions of OpenVMS prior to VMS 7.2 and Tru64 UNIX prior to 4.0.

6.7 Networking with Wireless Ethernet

Using a wireless NIC with Avanti requires special handling. Try one of the following options:

- Create a bridge TAP adapter.
- Set the virtual Alpha NIC MAC address to the same MAC address as the wireless card. THIS SOLUTION IS INCOMPATIBLE WITH DECnet Phase IV and DECnet Phase V running in compatibility mode. DECnet Phase IV will forcibly change the virtual Alpha NIC's MAC address to the DECnet phase IV address (AA-00-04-xx-xx-xx), breaking the wireless connection. Do not share the same TCP/IP address between the wireless host and the virtual Alpha system or it will confuse both systems.

Wireless Ethernet looks like Ethernet, but it isn't Ethernet. Using a wireless NIC with Avanti can be problematic. Here's why.

If the standard MAC of the Avanti NIC is used and the virtual NIC is attached to a wireless controller, then any packets which are not addressed to/from the "registered" PC's wireless MAC address may be legally dropped by the wireless router to conserve bandwidth.

The best way to debug a wireless router connection is to run a packet sniffer like WireShark on the host PC and watch the packet traffic, while also running a packet sniffer on a wired PC on the network to see if all the packets are traversing the wireless link correctly. It almost never operates as expected, because wireless routers are allowed to drop non-essential packets to conserve bandwidth. Non-TCP protocols like DECnet tend to get axed because modern routers, particularly consumer-grade routers, usually don't need to deal with non-TCP packets.

Spoofing the MAC address by assigning the PC wireless MAC to the Avanti NIC will avoid the wireless packet dropping, but only if DECnet Phase IV or DECnet Phase V (DECnet OSI) running in compatibility mode are not in use. DECnet Phase IV forcibly changes the hardware MAC to a DECnet-encoded value, which overrides the manually configured Avanti MAC address.

Spoofing the MAC will work when only TCP/IP is in use on the Avanti system, since the Avanti NIC will see all the matching MAC IP packets and throw out the incorrect IP-addressed packets.

MAC spoofing will work for DECNET Phase V in non-compatibility mode, LAT, and Clustering if the router/card combination does not filter out non-TCP protocols. DECnet Phase V will display a lot of unknown protocol messages to the console. Under OpenVMS, using a SET TERM OPA0: /NOBRO/PERM will hide the messages, or NCL can be used to disable the warnings.

It is possible, but difficult, to support DECNET IV use over wireless. A very few wireless routers support bridge mode, which passes all traffic over the wireless link as through it were a real Ethernet. For wireless combinations that can't bridge, the MAC of the wireless NIC can be manually changed to a DECNET IV AA-00-04-style MAC address by using a utility like SMAC or similar reverse-spoof techniques. This will work as long as

the wireless router/card combination supports non-TCP packets, but it is tricky to get DECNET IV started in the reverse-spoof environment, since it 'panics' when it sees the duplicate DECNET source address on the wire during protocol startup.

Some router/wireless card combinations will work flawlessly to route non-registered MAC addresses, most will not.

6.8 VMware NICs

The following VMware virtual NIC devices are known to work well with Avanti:

VMware NIC Type	Guest O/S
E1000	VMS & Tru64 UNIX
VMXNET 3	VMS

VMware ESXi, network cards are connected to portgroups on a DSwitch. For Avanti use we recommend creating a separate portgroup. Under the Security settings tab for this portgroup the following options should all be set to accept:

- Promiscuous mode
- MAC address changes
- Forged transmits

6.8.1 VMware Network Troubleshooting

This section lists potential VMware network problems and solutions.

6.8.1.1 Promiscuous Mode

Older versions of VMware ESXi and VMware Server may require that the virtual NICs be manually set to promiscuous mode. Depending on the VMware product, accomplishing this may require adding an entry to a VMware configuration file or finding the right configuration option in the configuration menus.

If the VMware NIC cannot be set to promiscuous mode, treat the VMware NIC like a wireless NIC. Create a bridged TAP adapter or assign the NICs MAC address to the virtual Alpha NIC. All the caveats in 6.7 Networking with Wireless Ethernet 68 apply.

6.8.1.2 Disappearing NIC

VMware has a bad habit of disabling a NIC if it does not appear to be in use at the Windows level. The symptoms are the NIC is still visible to the host system and to Avanti, but no longer communicates on the network. If this problem is encountered, enabling some Windows protocols on the NIC should fix it. First try enabling the following protocols:

- QoS Packet Scheduler
- Link-Layer Topology Discovery Mapper I/O Driver
- Link-Layer Topology Discovery Responder

If that does not correct the problem, enable Internet Protocol Version 4 (TCP/IPv4) on the NIC and allow it to be assigned an IP on the Windows side.

6.9 Hyper-V Virtual NICs

If DECnet Phase IV support is required under Hyper-V, MAC address spoofing must be enable. However, Microsoft Hyper-V versions prior to Windows Server 2008 R2 Hyper-V and Hyper-V Server 2008 R2 do not support MAC address spoofing. This prevents the MAC address change inherent in a DECnet Phase IV installation. DECnet Phase IV support under Hyper-V requires a Hyper-V release that supports MAC address spoofing.

6.10 Network Issues

6.10.1 TCP Offload Features May Cause Network Instability

Many modern NICs include offload features to reduce the load on a host system CPU. These features occasionally cause problems with DECnet, TCP/IP, and System Communications Services (SCS) communications. If one of the following problems is encountered, try turning off TCP related offloading under the Windows Configuration> Advance tab for the NIC assigned to Avanti, starting with TCP Connection Offloading.

- Unable to establish communications to the Avanti NIC.
- Communications with Avanti are unstable (time outs, packet loss)
- Cluster communications are unstable (SCS cluster time outs, packet loss)

6.10.2 Switch Drops DECnet and SCS Packets

Some modern switches drop DECnet and cluster (SCS) packets. Older Cisco Catalyst 4000 series switches were infamous for this. Often, when a switch gets busy, it will prioritize traffic and start dropping packets it does not recognize. Many modern switches do not recognize SCS and DECnet packets.

If you encounter this problem, try the following solutions:

- If using a managed switch, limit speeds to 10/100 on the port assigned to the Avanti NIC.
- If using an unmanaged gigabit switch, try replacing it with a 10/100 switch.
- If the first two options do not work, try substituting a different switch brand or model.

In-house, we have had good luck with Netgear, Linksys, and HP Procurve equipment. However, we have had issues with Netgear gigabit equipment in clustered environments. Cisco and Netgear 10/100 switches have worked well with clusters.

6.11 Third Party NDIS Drivers: WinPCAP, Win10Pcap, Npcap

Avanti has its own built in NDIS5 network driver called msipcap. This is installed by default with Avanti. Msipcap is based on WinPCAP and uses the same call interface.

Software applications like Wireshark and SimH require a third party NDIS driver for network support. In the past, WinPCAP was the preferred choice. However, WinPCAP is no longer supported and the last release, 4.1.3, conflicts with the msipcap driver. The

preferred NDIS driver for Wireshark and SimH is now Nmap.org's Npcap. Npcap has the advantage of being an NDIS6 driver.

Avanti can be configured to use npcap as its network driver. See 4.1.8.1 VLC [+] 42 for additional information. The following table lists the characteristics and limitations of the available WinPCAP NDIS drivers.

NDIS Driver	Characteristics
Npcap	Supported. NDIS6 driver. Migration Specialties recommends using Npcap if a third party NDIS driver is required for products like Wireshark. If Npcap is installed, Avanti will utilize it instead of the msipcap driver.
Win10Pcap	Incomplete implementation of WinPCAP for Windows 10. Only appears to work with Wireshark. Not compatible with Avanti. Migration Specialties recommends using Npcap.
WinPCAP	Depreciated. No longer officially supported. Migration Specialties recommends using Npcap. Avanti and software utilizing WinPCAP 4.1.2 will co-exist peacefully under Windows 7, 8, and Server 2012. WinPCAP 4.1.3 conflicts with Avanti's NDIS network driver. Under Windows 10 and Server 2016 and higher, all versions of WinPCAP conflict with the Avanti NDIS driver. Npcap is recommended if a tool like Wireshark is being used on a Windows 10 or Windows Server system that also supports Avanti.

Note: If installing a third party NDIS driver, reboot before starting Avanti to ensure it sees the new driver.

6.11.1 Npcap Installation

When installing Npcap, it is recommended that the following options be checked:

- Support raw 802.11 traffic (and monitor mode) for wireless adapters
- Install Npcap in WinPcap API-compatible Mode

Avanti Device Notes

VIRTUAL ALPHA

7 Avanti Device Notes

7.1 Default Serial Ports

Avanti and AvantiFlex support two default virtual or physical serial ports. FreeAXP supports two virtual serial ports only. Virtual serial ports can be accessed using the supplied PuTTY terminal emulator or a commercial terminal emulator such as KEA or Reflection. Physical serial ports can be assigned to physical COM ports on the host system.

Tip: Migration Specialties recommends enabling both virtual serial ports when building, modifying, or troubleshooting an Avanti configuration. Having two serial access points to a virtual Alpha system is useful in case one session locks up or network problems block telnet access.

7.1.1 Serial Port Telnet Port Numbers

When using Avanti virtual serial ports, the Avanti configuration utility default port number assignments are OPA0/tty00 = 9000 and TTA0/tty01 = 9001. If multiple Avanti instances are to be run on a single host, then unique port number assignments need to be made for each virtual serial port.

Example:

Avanti Instance	OPA0/tty00	TTA0/tty01
Alpha01	9000	9001
Alpha02	9002	9003
Alpha03	9004	9005

7.1.2 Serial Port Logging

Avanti supports logging activity at each serial. To enable logging on a serial port, check the Log file box in the serial port definition tab. By default, the log file name will use the configuration <base> value.

7.1.3 Console Port

The default console port on an OpenVMS system is OPA0 and tty00 on a Tru64 system. The console port is always enabled by default on new configurations.

7.1.4 Physical Serial Devices

Avanti supports the use of physical serial devices. Physical devices can be connected to an Avanti system by assigning a host system COM port to the Avanti virtual serial port. See 4.1.7.5 Serial Ports: OPA0/tty00 – TTA0/tty01 34, for instructions on assigning a serial port to a COM device.

Note: Baud rate mismatches between the Avanti serial port setting and the serial device can result in a hung serial port.

7.2 5-Port Serial Card Support (OPenVMS)

Under OpenVMS only, in addition to the console ports OPA0 and TTA0, Avanti supports up to 25 additional serial ports via five virtual 5-port serial cards. The serial cards are modified emulations of the Digiboard PC/8 ISA-based serial card that was supported by older AlphaServers.

Note: The Digiboard serial ports are not recognized by Tru64 UNIX and may cause unpredictable behavior if defined.

The serial cards are not supported by the Avanti configuration utility. They must be manually added to the configuration file. The following sample code demonstrates adding a card. A complete configuration file appears at the end of this chapter.

```
isa0 = digiboard
 { serial0 = i16550
  { tn = telnet
    { port = 9010;
action = """C:\Program Files\Common Files\MigrationSpecialties
Shared\putty.exe"" telnet://127.0.0.1:9010";
    }
  }
  serial1 = i16550
  { phys = ext_serial
    { device = "\\.\COM1";
    }
  }
  serial2 = i16550
  { phys = ext_serial
    { device = "\\.\COM2";
    }
  }
  serial3 = i16550
  { phys = ext_serial
    { device = "\\.\COM30";
    }
  }
  serial4 = i16550
  { phys = ext_serial
    { device = "\\.\COM31";
    }
  }
}
```

This example defines an emulated Digiboard card containing five serial ports; TXA0 - TXA4. TXA0 is a telnet port. TXA1 – TXA4 are assigned to physical COM ports 1, 2, 30 and 31.

Up to five serial ports can be defined on each card, running from serial0 thru serial4. Up to five Digiboard cards can be defined, running from isa0 thru isa4, for a total of 25 additional serial ports. However, isa2, isa3, and isa4 conflict with pci13, pci12, and pci11 respectively. That means that isa2 can only be used if pci13 is not used, etc. This limits the number of Digiboard cards to four for most systems (where one disk controller and one NIC are required).

The sections within “serial = i16550 { … }” follow the same rules as the regular serial ports, so in addition to defining them as telnet server ports, they can also be assigned to physical serial ports (COMn) on the host server.

The emulated Digiboard PC/8 is supported by the VMS SYS$YSDRIVER. As on a real Alpha system, the card is not recognized by VMS by default. It requires additional set up steps to make it work.

7.2.1 Defining the Serial Card to the Firmware

First, the card must be made known to VMS through the firmware ISACFG utility. Once the serial cards and ports are defined within the configuration file, the exact ISACFG invocations to be used will be written to the emulator log file along with the associated VMS 7.1 and higher configuration commands. After creating the configuration file, start the emulator and type "exit" at the >>> prompt, then open the log file in a text editor to determine the correct set up commands for each Digiboard card. The following section of log file was generated from the configuration code shown earlier.

```
00011.78: Digiboard multi-serial card configured.
00011.78: Firmware configuration commands:
00011.78:    >>> isacfg -mk -slot 1 -dev 0 -etyp 1 -enadev 1 -handle
DIGI
00011.78:    >>> isacfg -mod -slot 1 -dev 0 -iobase0 100 -iobase1 108
-iobase2 110 -iobase3 118 -iobase4 120 -iobase5 128 -irq0 15
00011.78: Add these lines to SYS$SYSTEM:SYS$USER_CONFIG.DAT:
00011.78:    device          = "Digiboard Serial Port"
00011.78:      name          = TX
00011.78:      driver        = SYS$YSDRIVER
00011.78:      adapter       = ISA
00011.78:      id            = DIGI
00011.78:    end_device
00011.78: Add these lines to your startup procedure:
00011.78:    $ mcr sysman io connect /vector=60 /driver=sys$ysdriver
txa1:
00011.78:    $ mcr sysman io connect /vector=60 /driver=sys$ysdriver
txa2:
00011.78:    $ mcr sysman io connect /vector=60 /driver=sys$ysdriver
txa3:
00011.78:    $ mcr sysman io connect /vector=60 /driver=sys$ysdriver
txa4:
```

Log file output providing the firmware and VMS commands required to complete the configuration of the serial ports. These lines will appear near the beginning of the Avanti log file.

Using the commands provided above, the following console commands define the first serial card. These commands only need to be entered once. Their settings are retained in the Avanti configuration's .nvr file.

```
>>> isacfg -mk -slot 1 -dev 0 -etyp 1 -enadev 1 -handle DIGI
>>>init
>>> isacfg -mod -slot 1 -dev 0 -iobase0 100 -iobase1 108 -iobase2 110
-iobase3 118 -iobase4 120 -iobase5 128 -irq0 15
>>>init
```

Cut and paste the ISACFG commands from the Avanti log file to the console to define the serial card in the firmware.

A "sh config" or "isacfg –all" command can be used to confirm the presence of the ISA card definition in the firmware.

```
>>>sh config

Firmware
SRM Console:    V7.0-9
show_arc: Can't find Flash Rom containing ARC console.
PALcode:        VMS PALcode V5.56-2, OSF PALcode X1.46-2
Serial Rom:     V4.6
Diag Rom:       V1.7

Processor
DECchip (tm) 21064-3    166Mhz 512KB Cache

MEMORY
     512 Meg of System Memory
     Bank 0 = 256 Mbytes(128 MB Per Simm) Starting at 0x0
     Bank 1 = 256 Mbytes(128 MB Per Simm) Starting at 0x10000000
     Bank 2 = No Memory Detected

PCI Bus
     Bus 00  Slot 06: NCR     810A Scsi Controller
                                   pka0.7.0.6.0          SCSI Bus ID 7
                                   dka200.2.0.6.0         RZ28
                                   dka300.3.0.6.0         COMPAQ
BA03611C9B
                                   dka600.6.0.6.0         RRD42

     Bus 00  Slot 07: Intel SIO 82378


     Bus 00  Slot 11: DECchip 21040 Network Controller
                                   ewa0.0.0.11.0         02-13-46-6A-
94-24

ISA
Slot    Device  Name            Type         Enabled  BaseAddr  IRQ
  DMA
0
        0       MOUSE           Embedded        Yes     60      12
        1       KBD             Embedded        Yes     60      1
        2       COM1            Embedded        Yes     3f8     4
        3       COM2            Embedded        Yes     2f8     3
        4       LPT1            Embedded        Yes     3bc     7
        5       FLOPPY          Embedded        Yes     3f0     6
  2
1
        0       DIGI            Singleport      Yes     100     15
                                                        108
                                                        110
                                                        118
                                                        120
                                                        128
>>>
```

Results of a sh config after a serial card has been defined. The serial card results are highlighted in blue.

7.2.2 Defining the Serial Card to OpenVMS

Once the firmware changes have been completed, boot VMS and apply the VMS configuration updates to enable the card and serial ports. Versions of OpenVMS prior to V7.1 use different configuration options from VMS 7.1 and higher.

7.2.2.1 Serial Card Configuration Prior to OpenVMS 7.1

Version of VMS prior to V7.1 use the SYS$SPECIFIC:[SYSMGR]ISA_CONFIG.DAT to define ISA devices. If the file does not already exist, it can be created from the SYS$COMMON[SYSMGR]:ISA_CONFIG.TEMPLATE.

1. The commands written to the Avanti log file are for VMS 7.1 and higher. However, the information provided in these commands can be used to configure earlier versions of VMS. Using the data written to the Avanti log file, adding the following lines to SYS$SPECIFIC:[SYSMGR]ISA_CONFIG.DAT will enable serial card recognition by OpenVMS 6.2.

```
[TXA0]
NAME=TX
DRIVER=SYS$YSDRIVER
IRQ=15                                          ← irq0
NODE=1                                          ← slot
PORT=(100:8,108:8,110:8,118:8,120:8,128:8)      ← iobase0..iobase5
```

2. If more than one Digiboard is defined in the configuration file, add similar sections for [TXB0], [TXC0], [TXD0] and [TXE0] as needed. For these sections, replace the text highlighted in red with the data for that card obtained from the Avanti log file.

3. Once ISA_CONFIG.DAT has been updated, reboot VMS.

4. After the reboot is complete, a TXA0 device will be present that corresponds to serial0 on the Digiboard card. If additional cards are defined, the first port on each of these will be present as well (TXB0, TXC0, TXD0, TXE0). TXu1 thru TXu4 can be added using the SYSMAN IO CONNECT command. For example, placing the following commands into a VMS startup file such as SYS$MANAGER:SYSTARTUP_VMS.COM will enable devices TXA1 – TXA4.

```
$! Define additional serial ports.
$!
$ MCR SYSMAN
IO CONNECT/NOADAP/DRIVER=SYS$YSDRIVER TXA1:
IO CONNECT/NOADAP/DRIVER=SYS$YSDRIVER TXA2:
IO CONNECT/NOADAP/DRIVER=SYS$YSDRIVER TXA3:
IO CONNECT/NOADAP/DRIVER=SYS$YSDRIVER TXA4:
EXIT
$!
```

With the firmware and VMS configuration changes completed, the additional serial ports are available for use.

Note: Errors in ISA_CONFIG.DAT or in the SYSMAN commands could crash OpenVMS. Be very careful in applying these settings. If VMS will not boot after modifying ISA_CONFIG.DAT, use a conversational boot with STARTUP_P1 set to "MIN" to regain access to the system and remove the offending ISA_CONFIG.DAT file.

7.2.2.2 OpenVMS 7.1 and Higher Serial Card Configuration

The firmware and VMS commands written to the Avanti log file support OpenVMS 7.1 and higher without modification. OpenVMS 7.1 and higher uses the SYS$SYSTEM:SYS$USER_CONFIG.DAT to define ISA devices.

1. Per the commands provided, add the following lines to SYS$SYSTEM:SYS$USER_CONFIG.DAT to enable serial card recognition by VMS. If the SYS$SYSTEM:SYS$USER_CONFIG.DAT does not exist, create it.

```
device = "Digiboard Serial Port"
  name = TX
  driver = SYS$YSDRIVER
  adapter = ISA
  id = DIGI
end_device
```

2. Reboot VMS.
3. After the reboot is complete, a TXA0 device will be present, which corresponds to serial0 on the Digiboard card. TXA1 thru TXA4 can be added using the SYSMAN IO CONNECT command. Place the following commands into a VMS startup file such as SYS$MANAGER:SYSTARTUP_VMS.COM:

```
$! Set up system serial ports.
$!
$ MCR SYSMAN
IO CONNECT /VECTOR=60 /DRIVER=SYS$YSDRIVER TXA1:
IO CONNECT /VECTOR=60 /DRIVER=SYS$YSDRIVER TXA2:
IO CONNECT /VECTOR=60 /DRIVER=SYS$YSDRIVER TXA3:
IO CONNECT /VECTOR=60 /DRIVER=SYS$YSDRIVER TXA4:
EXIT
$!
```

Similar commands are required to define TXB0 thru TXB4, TXC0 thru TXC4, TXD0 thru TXD4, and TXE0 thru TXE4. If additional serial cards are defined, the appropriate commands will be written to the emulator log file.

With the firmware and VMS configuration changes completed, the additional serial ports are available for use.

Note: Errors in SYS$USER_CONFIG.DAT or in the SYSMAN commands could crash OpenVMS. Be very careful in applying these settings. If VMS will not boot after modifying SYS$USER_CONFIG.DAT, use a conversational boot with STARTUP_P1 set to "MIN" to regain access to the system and remove the offending SYS$USER_CONFIG.DAT file.

7.2.3 Sample Configuration File With Serial Card

The following example is a complete Avanti configuration file that includes one serial card.

```
cp = control
{
  cp_log = logfile
```

```
  {
    filename = "D:\Avanti\Alpha07.1-2.log";
  }
  alpha = alpha
  {
    image = "D:\Avanti\Alpha07.1-2.nvr";
    memory = 512M;
    jit_level = 1;
    pcibus = dc21071da
    {
      data = "D:\Avanti\Alpha07.1-2.toy";
      serial0 = i16550
      {
        s0_tn = telnet
        {
          port = 9000;
          action = """C:\Program Files\Common
Files\MigrationSpecialties
Shared\putty.exe"" telnet://127.0.0.1:9000";
        }
      }
      serial1 = i16550
      {
        tn = telnet
        {
          port = 9001;
          action = """C:\Program Files\Common
Files\MigrationSpecialties
Shared\putty.exe"" telnet://127.0.0.1:9001";
        }
      }
      pci6 = symbios
      {
        disk0.2 = file
        {
          vendor = "DEC";
          model_num = "RZ28";
          rev_num = "D41C";
          image = "D:\Avanti\Alpha\SystemV71-2.vdisk";
        }
        disk0.3 = file
        {
          vendor = "COMPAQ";
          model_num = "BA03611C9B";
          rev_num = "3B07";
          image = "D:\Avanti\Alpha\Data.vdisk";
        }
        disk0.6 = device
        {
          device = "\\.\CdRom0";
          cdrom = true;
          read_only = true;
        }
      }
      pci7 = i82378
      {
      }
      pci11 = de435
      {
```

```
        adapter = "Emulator1";
      }
      isa0 = digiboard
      {
        serial0 = i16550
        {
          tn = telnet
          {
            port = 9010;
            action = """C:\Program Files\Common
Files\MigrationSpecialties Shared\putty.exe""
telnet://127.0.0.1:9010";
          }
        }
        serial1 = i16550
        {
          phys = ext_serial
          {
            device = "\\.\COM1";
          }
        }
        serial2 = i16550
        {
          phys = ext_serial
          {
            device = "\\.\COM2";
          }
        }
        serial3 = i16550
        {
          phys = ext_serial
          {
            device = "\\.\COM30";
          }
        }
        serial4 = i16550
        {
          phys = ext_serial
          {
            device = "\\.\COM31";
          }
        }
      }
    }
  }
}
```

7.3 Virtual Tapes

Like virtual disks, virtual tape devices appear as files to the host system. By default, Avanti assigns the extension .vtape to a virtual tape file. The vtape format utilized by Avanti is fully compatible with the vtape format used in SimH. Avanti virtual tape devices default to a DEC DLT2000 device type.

Note:Avanti puts no limit on the size of a vtape file. Continuously appending to a tape will eventually fill the host system disk.

Avanti treats virtual tape files as follows:

- If a virtual tape files does not exist on Avanti startup, it is created.
- If a virtual tape file is removed while Avanti is running, Avanti will create a new tape file then next time the associated tape device is accessed. See 7.3.1 Changing Tapes 82 for more on this feature.
- If a virtual tape file exists on Avanti startup, it is used.
- Avanti locks access to a tape file upon the following operations:
  - Tru64: Tape read/write operations commence.
  - VMS: Tape INITIALIZE or MOUNT operation is executed. If a tape is INITIALIZE'd, its underlying file will not be unlocked until a MOUNT/DISMOUNT operation is performed or VMS is shutdown.
- Avanti releases a tape file when an UNLOAD command is sent to the virtual tape drive. The following operations issue an UNLOAD command:
  - Tru64: mt unload or mt offline operation is executed.
  - VMS: Tape DISMOUNT operation is executed.
- Under VMS, when a tape is initialized, the underlying tape file is reduced to a few Kbytes in size.

7.3.1 Changing Tapes

Virtual tapes can be changed while Avanti is running by renaming the virtual tape file. Avanti will recreate the original tape file the next time the tape device is accessed. To change the virtual tape Tape1.vtape, follow these steps:

1) Stop tape access.
   a) Tru64: Stop writing to the tape.
   b) VMS: DISMOUNT the tape device.
2) From the host O/S, rename the file Tape1.vtape to a new name, such as AUG03_Backup.vtape.
3) The next time a tape operation takes place against the virtual tape device, the Tape1.vtape file will be recreated.

7.3.1.1 Virtual Tape File Space Allocation

When created or initialized, a virtual tape file allocates 100MB of space on the host system disk. If the virtual tape exceeds the initial allocation, subsequent allocations to the tape file are made in 100MB increments. Allocating space in large increments is done to avoid file fragmentation problems with Windows. When the virtual tape device is dismounted, the tape file is resized to its actual size.

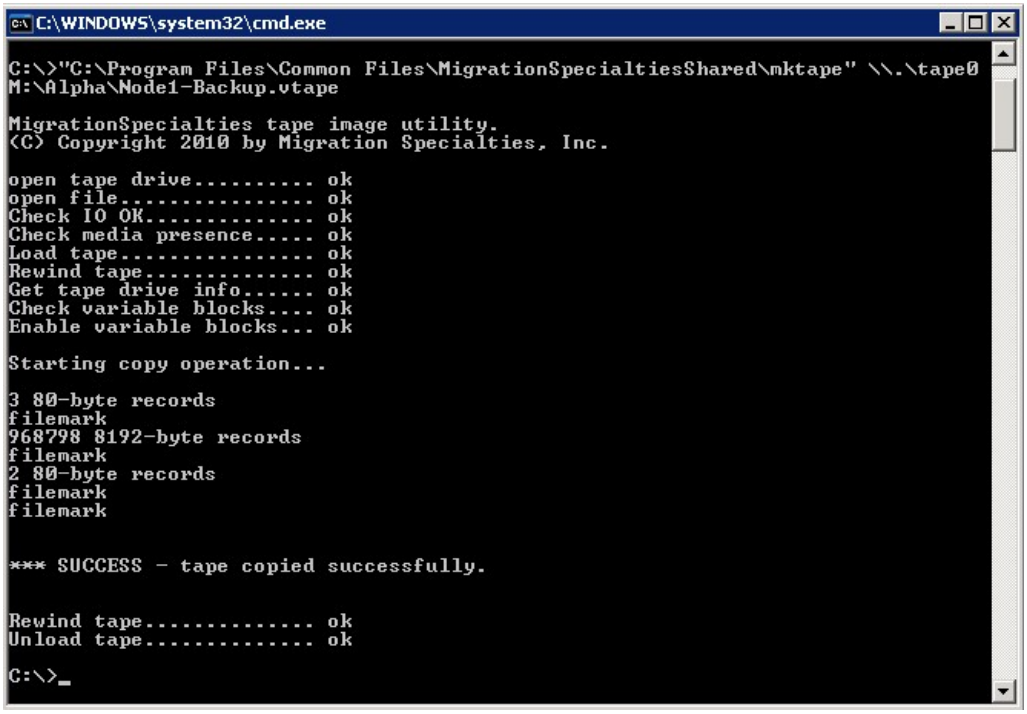
7.3.1.2 Make Tape Utility

To facilitate converting real tapes to virtual tapes, Avanti includes the mktape utility. Mktape requires a Windows system with a SCSI controller and compatible SCSI tape drive to read OpenVMS or Tru64 UNIX generated tapes.

To create a tape image file from a real tape using mktape, open a command window and enter a command similar to the following:

```
"C:\Program Files\Common Files\MigrationSpecialties
Shared\mktape" \\.\tape0 c:\dec\tape1.vtape
```

Where \\.\tape0 is the SCSI tape drive attached to the windows system and c:\dec\tape1.vtape is the name of the virtual tape file that is to be created from the contents of the physical tape.



Example of mktape usage.

OpenVMS Specific Notes

8 OpenVMS Specific Notes

8.1 Porting an OpenVMS System

Porting an OpenVMS system to Avanti can be as simple as backing up the legacy Alpha system across a network to an Avanti system. If you are comfortable in the OpenVMS environment, this is a straightforward process.

If you are not comfortable with OpenVMS, Migration Specialties offers turnkey OpenVMS porting services. Contact us for more information at sales@migrationspecialties.nl.

8.2 OpenVMS Auto Boot

Under OpenVMS, the virtual Alpha can be configured to boot automatically by setting the following console variables:

```
>>> SET AUTO_ACTION BOOT
>>> SET BOOT_OSFLAGS 0
>>> SET BOOTDEF_DEV ddua
```

Where ddua is a disk device such as DKA0. Use the exit command to save console parameters in the .nvr file.

8.3 Reconfiguring the Network

Avanti uses the EWA0, EWB0, and EWC0 network devices. Porting a system that uses different network controllers will require reconfiguration of the network settings in VMS to support the EWnn adapters. Here is a simple set of steps to accomplish this for DECnet Phase IV and Digital/Compaq/HP TCP/IP services.

1) Disable the DECnet (STARTNET.COM) and TCP/IP (UCX$STARTUP.COM) startups in SYSTARTUP_VMS.COM. Comment out commands similar to these:

```
$ IF F$SEARCH("SYS$SYSTEM:NETACP.EXE") .NES. "" THEN
   @SYS$MANAGER:STARTNET
```

```
$ @SYS$MANAGER:UCX$STARTUP.COM
```

```
$ @SYS$MANAGER:TCPIP$STARTUP.COM
```

2) Reboot VMS.

3) Run NETCONFIG.COM and reset the DECnet parameters.

4) Enable DECnet startup in SYSTARTUP_VMS.COM (STARTNET.COM).

Note: It is very important that DECnet services start before TCP/IP services. Some VMS startup configurations submit DECnet and TCP/IP startup commands as batch procedures. This practice should be discontinued; the startup calls should be executed directly from the SYSTARTUP_VMS procedure as inline commands.

5) Reboot. DECnet and LAT should now come up normally.

6) Run UCX$CONFIG.COM or TCPIP$CONFIG.COM and reset the TCP/IP configuration.

7) Enable TCP/IP in SYSTARTUP_VMS.COM (UCX$STARTUP or TCPIO$STARTUP).

8) Reboot. TCP/IP should now come up normally.

8.4 Bypassing a NIC

If porting a system with two NICs installed, but only one in use, the Dummy network option can be used to define a NIC to the emulator to retain the existing system's network configuration. For example, if a system has NICs EWA and EWB defined, but only EWB is used, EWA can be assigned to Dummy network while EWB is assigned to a host NIC. The original Alpha configuration is thus preserved, which may negate the necessity of updating the VMS network configuration.

8.5 Maximum Physical Disk Size Under OpenVMS

OpenVMS V6.0 through V8.3 support a maximum physical disk size of 1TB. OpenVMS V8.4 introduced a maximum physical disk size of 2TB. See Section 9.5 of the OpenVMS FAQ for more details: http://www.hoffmanlabs.com/vmsfaq/vmsfaq_013.html#file5

Caveats:

- Pre 7.2 version do not support a system disk larger than 32GB.
- Pre 7.2 version have a maximum of 1 million clusters, so for an n-GB drive the cluster size will be 2*n, which may get rather large.
- The SWXCR RAID controllers had their own hardware limitations.

8.6 OpenVMS Clustering

- OpenVMS clusters between Avanti and real VAX, Alpha, and Itanium hardware have been successfully created and are stable.
- Avanti supports satellite booting into a cluster.

8.6.1 Quorum Disk and TCQ

When a cluster is configured with a quorum disk, Tagged Command Queuing (TCQ6) must be enabled on the quorum disk. See 4.1.8.6 SCSI Disk [+] 47 for instructions on enabling TCQ.

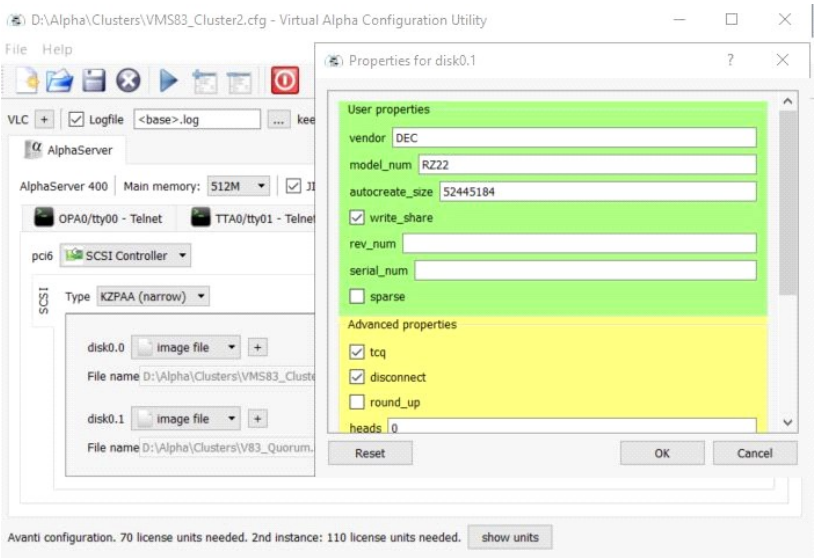
8.6.2 Clusters and Disk Sharing

Avanti supports shared disks for clustering, where several nodes share one or more disk drives. Disk sharing between Avanti instances running on the same host system is enabled using the write_share option in the advance disk options (see 4.1.8.6 SCSI Disk [+] 47).

Warning: Using the write_share feature to share disks with non-clustered Avanti instances is not recommended. Doing so may result in disk corruption.

If Avanti is running on separate host systems, disk sharing is only possible using physical devices such as SAN disks that are physically shared between the host systems.

To enable disk sharing, point each Avanti configuration file at the exact same disk image file(s) and enable write_share for each shared disk.



Example: An OpenVMS 8.3 cluster shares a system disk and quorum disk. The disks are store in image files VMS83_Clustering.img and V83_Quorum.img. This image depicts the use of the write_share option to enable disk sharing between Avanti instances. It also depicts the TCQ option required on the quorum disk. Each disk definition in the other cluster member Avanti configuration files is tagged with the write_share option. All configuration files point at the same disk image files.

8.7 VMS 6.n and Large Disk Support

Starting with Avanti 2.4, VMS 6.n systems properly recognize large disks. Past versions of Avanti limited VMS 6.n systems to a maximum disk size of 8GB. Use this feature with caution, as VMS 6.n may have other issues working with large disks.

8.8 VMS 6.2 Installation

VMS 6.2 requires a minimum disk size of 635MB (RZ56) for installation. Migration Specialties recommends a minimum disk size of 1GB.

8.9 VMS 6.2 Image Backups

To create a proper system disk image backup under V6.2, it is necessary to issue the BACKUP command with the /NOALIAS qualifier. BACKUP/IMAGE without the /NOALIAS qualifier does not pick up the VMS$COMMON directory or its contents.

These commands properly copy and restore the VMS$COMMON directory under VMS 6.2.

Image Backup Command

```
$ BACKUP/IMAGE/NOALIAS/IGNORE=INTERLOCK DKA0: MKA600:SYSTEM.BCK/SAVE
```

Image Restore Command

```
$ BACKUP/IMAGE MKA600:SYSTEM.BCK/SAVE DKB0:
```

8.10 VMS 6.2 Network Support

Avanti supports both DE435 and DE500 virtual NICs. Versions of OpenVMS prior to 7.2 are not fully compatible with the DE500. It is recommended to use the DE435 with versions of VMS prior to VMS 7.2.

If using the DE500 with a version of VMS prior to 7.2, do the following:

- Use the DE500 21143 NIC definition in the Avanti configuration file.
- Insert the following code into SYSTARTUP_VMS.COM to work around the compatibility issues and to bring up DECnet and TCP/IP services.

```
$ @SYS$MANAGER:STARTNET.COM      !Start DECnet
$ MC NCP SET EXEC STAT OFF       !Shut down DECnet
$ WAIT 00:00:10                  !Pause 10 seconds*
$ @SYS$MANAGER:STARTNET.COM      !Restart DECnet
⋮
$ @SYS$STARTUP:UCX$STARTUP.COM   !Start TCP/IP services
```

*Failure to pause long enough may result in a NETNOSTATE bug check

8.11 VMS 7.0 Login After Installation

After installing OpenVMS 7.0, it is not possible to log in from the console. This is not an Avanti bug. The real AlphaServer 400 behaves the same way.

An alternative is to install the OpenVMS licenses during the installation process. It is then possible to log into the system via TTA0.

Another option is to execute a conversational boot to log into the system and install licenses. Here is one way to do this:

8.11.1 VMS 7.0 Conversational Boot

```
>>> B DKA0 -FLAG 0,1
SYSBOOT> SET/START=OPA0:
SYSBOOT> CONTINUE
```

```
The system will boot to the $ prompt.  Spawn a subprocess to stay
logged on to the console; then run the system startup procedure.
```

```
$ SET NOON
$ SPAWN
%DCL-S-SPAWNED, process SYSTEM_1 spawned
%DCL-S-ATTACHED, terminal now attached to process SYSTEM_1
$ @SYS$SYSTEM:STARTUP
```

```
The system will come up normally and give back the $ prompt when the
startup procedure logs out. From there, spawn again to prevent being
logged out of the console.  It will not e possible to work on the
system.
```

```
$ SET NOON
$ SPAWN
%DCL-S-SPAWNED, process SYSTEM_1 spawned
%DCL-S-ATTACHED, terminal now attached to process SYSTEM_1
$ SET DEF SYS$SYSTEM:
```

```
When finished, do another conversational boot and reset the startup
file.
```

```
>>> B DKA0 -FLAG 0,1
SYSBOOT> SET/START=SYS$SYSTEM:STARTUP.COM
SYSBOOT> CONTINUE
```

8.12 VMS 7.1 Installation Halt

When installing OpenVMS 7.1 from an ISO image of the installation media, the boot process occasional halts during the first O/S boot after the installation process has completed. If this occurs, simply boot the O/S again to recover.

8.13 VMS 7.3 Install from Virtual RRD42

We have seen cases where the OpenVMS 7.3 installation from an ISO image goes into a mount verify and does not recover when run from an emulated RRD42. If this problem is encountered, ensure the emulated CD drive advanced properties are set as follows:

- vendor: DEC
- model_num: RRD40

These setting eliminate the problem. The VMS 7.3 device driver seems to have issues with the RRD42 device.

8.14 OpenVMS 8.3

OpenVMS 8.3 requires a minimum of 64MB of memory to run.

8.15 OpenVMS Alpha V8.4-2L2

VSI's OpenVMS V8.4-2L2 Performance Release for Alpha requires Avanti version 3.0.0.617 or higher.

8.16 VSI OpenVMS Rebranding Error

If a rebranding error is encountered while upgrading a non-VSI OpenVMS system disk to a VSI version of OpenVMS, try one of the following solutions:

- Upgrade the disk to ODS-5 and enable hard links.
- Check and repair the SYSCOMMON/VMS$COMMON directory aliasing. Read the following article for details: ftp://ftp.hp.com/%2F/pub/enterprise/programming_code/c00811161Checking_the_Alias_Directory....txt
- Use the DFU utility off the OpenVMS Freeware CD to analyze and repair the disk file structure.

8.17 OpenVMS SCSI Support Limitations

OpenVMS versions prior to 7.1-2 do not support the KZPCA wide SCSI controller. VMS 7.1-1H2 and earlier are limited to the KZPAA narrow controller and seven disks per controller.

8.18 Serial Devices

By default, Avanti presents two serial ports under OpenVMS, OPA0 and TTA0. Procedures that check for serial devices by scanning for a TX device will need to be updated to reflect the ports presented by Avanti.

Example of obtaining the terminal device name:

```
$ TERM = F$GETDVI("SYS$INPUT","DEVNAM")
$ SH SYM TERM
  T = "_TTA0:"
```

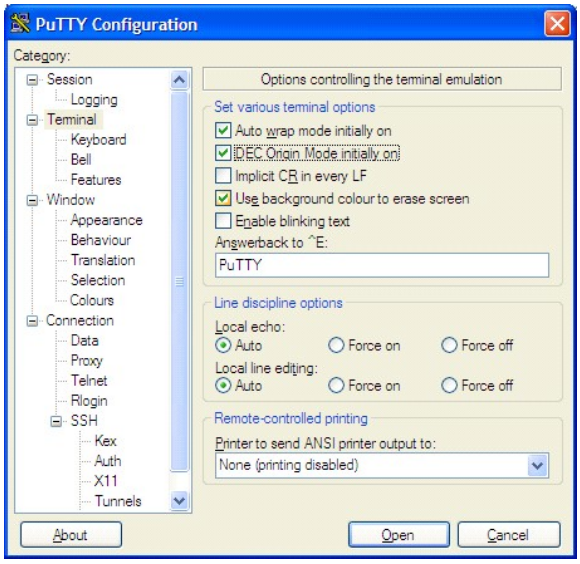
Note: TX serial devices can be added using the Digiboard serial card. See 7.2 5-Port Serial Card Support (OPenVMS) 75.

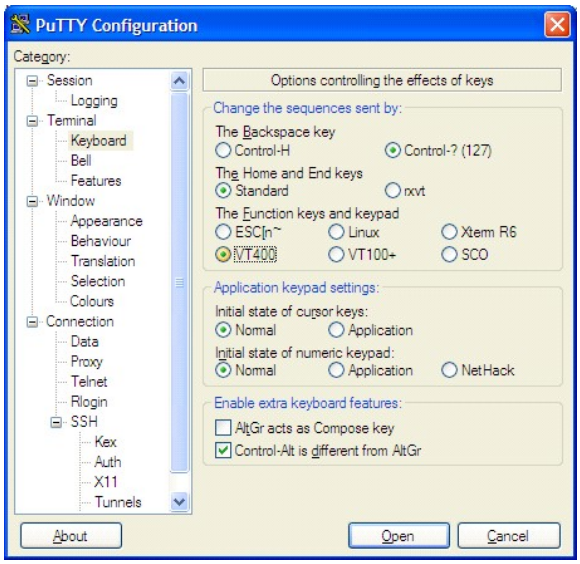
8.19 Reflection 2011 & E-Term32 Virtual Sessions

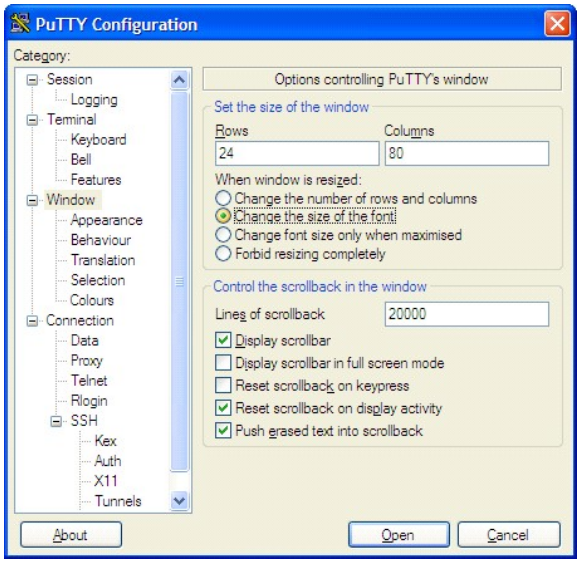
Prior to Avanti 2.4, OpenVMS login's using Reflection 2011 or E-Term32 VT terminal emulators to the OPA0 or TTA0 virtual serial ports may not be possible. The Reflection and E-Term32 emulators appear to skip password entry. PuTTY, SmarTerm, KEA, Hyperterm, and most other VT terminal emulators do not exhibit this behavior.

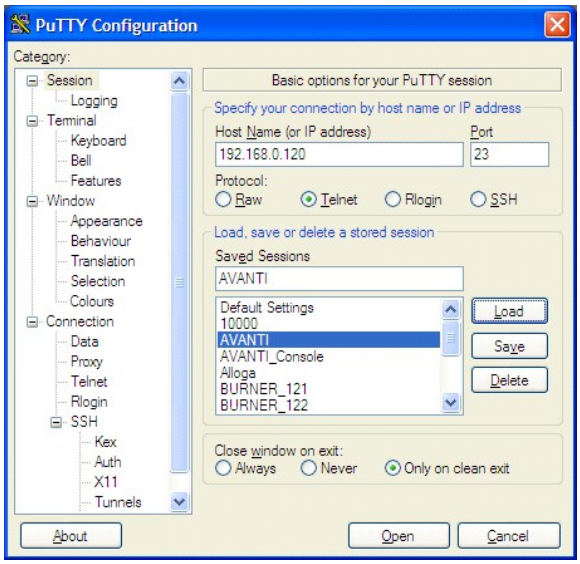
The Telnet standard states that a <CR> that is not followed by an <LF> should instead be followed by a <NULL>. Most VT terminal emulators do not conform to this standard. However, Reflection and E-Term32 closely follow the Telnet standard and do implement this feature. VMS does not handle the <NULL> on the serial port very well. Avanti has been updated to handle the <NULL> properly, which resolves the problem with Reflection and E-Term32 logins against virtual serial ports.

8.20 OpenVMS PuTTY Settings









Starting with a default PuTTY configuration, the settings shown above are typically used to access OpenVMS on Avanti virtual Alpha systems via serial or telnet connections. Products like Oracle Forms will appreciate the VT400 keyboard setting.

8.20.1 Transferring PuTTY Settings

PuTTY stores saved session settings in the Windows registry. The settings can be transferred to another Windows machine using regedit.

To export the PuTTY registry key on the source Windows machine, click on Start> Run> and enter the following regedit command in the run dialog box:

```
regedit /e "%userprofile%\desktop\putty-registry.reg"
HKEY_CURRENT_USER\Software\Simontatham
```

This command places the PuTTY registry key on the desktop in the file putty-registry.reg. Note that the name of the registry key (Simon Thatham) is the author of PuTTY.

Copy the putty-registry.reg file to the new Windows system. Right click the file and select the Merge option to add the key to the Windows registry. The PuTTY sessions defined in the transfer file will be added to any existing PuTTY sessions already saved on the machine.

8.21 UNXSIGNAL Bug Check

The UNXSIGNAL bug check has been observed on versions of VMS prior to OpenVMS 7.3 when attempting to write to a new disk.

```
**** OpenVMS (TM) Alpha Operating System V6.2     - BUGCHECK ****
** Code=0000041C: UNXSIGNAL, Unexpected signal name in ACP
** Crash CPU: 00    Primary CPU: 00    Active CPUs: 00000001
** Current Process = BPCD_BG30
** Image Name = BUILD$DKC0:[OPENV.NETBACKUP]BPCD_AXP.EXE;1
```

This crash is due to an old disk geometry problem. The problem causes an overflow in one of the following UCB fields:

UCB$B_SECTORS
UCB$B_TRACKS
UCB$W_CYLINDERS

This results in the VCB$L_BLOCKFACT field in the VCB being zero. The first space allocation on the volume generates a divide-by-zero error, which crashes the system with a UNXSIGNAL bugcheck.

The fix is in MOUNTSHR.EXE in V7.3 (VAX and Alpha).

Fixes are also available in the following patch kits:

MOUNTSHR.EXE_V62R
VMOUNT.EXE_V62R
VMS73_F11X-V0200
VMS73_F11X-V0300

The workaround is to change the size of the virtual disk. For example, if an RZ25 generates the bug check, change the disk to an RZ56.

Tru64 UNIX Specific Notes

9 Tru64 UNIX Specific Notes

9.1 Porting a Tru64 System

From a hardware perspective, porting a Tru64 UNIX system to Avanti is equivalent to moving it to an AlphaServer 400. If you are comfortable in the Tru64 UNIX environment, this is a straightforward process. If you are not comfortable with Tru64 UNIX, Migration Specialties offers turnkey porting services. Contact us for more information at camiel@migrationspecialties.nl.

There are two choices when moving from one Alpha type to another:

1) Boot from the O/S CD, build a new system disk, and restore/merge the files from your original system or disks.

2) Boot from the old system disk using the Generic kernel, genvmunix, and build a new kernel that supports the new Alpha's architecture (i.e. the platform type).

To boot the generic kernel, use something like:

```
>>> boot dkaXXX -flags 1 -file /genvmunix
```

This should boot single user with the hardware recognized and root mounted read-only. If it works, then you have the option of running from genvmunix going forward or rebuilding a new kernel.

Section 5 of the Tru64 System Admininstrator's Guide has good discussion of how to do this. The V4 documentation set is available online at HP. The following link is a good starting point:
http://www.frascati.enea.it/documentation/tru6450/ARH8RATE/FLLDCSTC.HTM

You will need to run bcheckrc, sizer -n, do some editing of the new config file, and use doconfig to build a new 'vmunix', as per section 5.4.1 of the V4 manual above. Do not delete or overwrite genvmunix.

9.2 Tru64 Auto Boot

Under Tru64 UNIX, the virtual Alpha can be configured to boot automatically by setting the following console variables.

```
>>> SET AUTO_ACTION BOOT
>>> SET BOOTDEF_DEV DKxnnn
>>> SET BOOT_OSFLAGS A
```

Where DKxnnn equates to a disk device such as DKA0. Use the exit command to save console parameters in the .nvr file.

9.3 Tru64 Version Support

Avanti supports Tru64 UNIX V3.2C and higher. Avanti has not been tested with unqualified External Field Test (EFT) or pre-Release versions of Tru64 UNIX.

9.4 Network Notes

9.4.1 NIC Support in V3.2X Releases

Tru64 versions in the V3.2X family require that the DE435 NIC be used since they pre-date the introduction of the DE500 NIC. A DE500 will not be recognized by a V3.2x release.

9.4.2 NIC Console Variables

Regardless of the settings of the physical NIC used in Windows, set and lock the DE500 mode to 'Fast', and the DE435 to 'Twisted Pair'. Do NOT use Auto at any time.

```
>>> set ewa0_mode Fast
>>> set ewb0_mode Twisted Pair
>>> init
```

Use the exit command to save console parameters in the .nvr file.

Failing to select and lock the NIC speed from the console may result in unpredictable behavior by the Tru64 Tulip NIC driver. Symptoms include not recognizing the card, poor performance, and only uni-directional traffic.

9.4.3 Sharing NICs with Windows

In a dual-NIC configuration, some Windows NIC drivers do not behave properly when sharing the Windows active NIC with the primary NIC for Tru64. Symptoms may include poor throughput, packet loss/re-transmission, or even Tru64 UNIX 0x670 Machine Checks associated with incoming broadcast packets. The solution is to use the recommended configuration of dedicated NICs for the virtual Alpha. There is a wide range of NICs and associated drivers available for Windows and many will not exhibit this shortcoming.

9.4.4 LAT and Dual NICs

When operating with dual NICs and running LAT, if both NICs are on the same segment or switch, exclude one NIC from LAT. This prevents creation of ghost services or nodes.

```
# latcp -E tu1
```

9.4.5 Switching NICs

If the Tru64 system being ported to Avanti does not already use a DE435 or DE500 NIC, then a new kernel will need to be built which includes 'tu' support before it recognizes Avanti's DE435/500 hardware. There is a section on doing this in the Tru64 Administration Guide.

- Tru64 UNIX V5.1B System Admin Guide AA-RH9FE-TE
  Sections 4.6.1, 4.6.2 and 4.6.2 - Building the Kernel under various circumstances.

Or

- Tru64 UNIX V4.0F System Admin Guide AA-PS2RF-TE
  Sections 5.4.1, 5.4.2 and 5.4.3 - Building the Kernel under various circumstances.

9.5 SCSI Support Limitations

Tru64 V3.2x does not support the KZPCA wide SCSI controller. Both V3.2x and V4 do not support SCSI IDs greater than 7; therefore, these versions of Tru64 are limited to seven disks per SCSI controller.

9.6 V5 Disk Considerations

All V5 releases of Tru64 UNIX attempt to interrogate HP/Compaq storage arrays to acquire essential capability information. Consequently, whenever an HSG, HSZ, EVA or XP disk unit is detected, V5 will request the appropriate array specific code pages containing the unit information. However, when Avanti is using disk volume files there is no array information to be discovered. Therefore, when using Tru64 V5 on Avanti, labeling disk volume files as array volumes is not recommended. Instead, for disks above 18GB, use the default RZ58 catch-all type. For all other volumes, use one of the supplied disk types.

Warning: Labeling a volume as an HSZ, HSG, etc. unit will often cause Tru64 UNIX V5 to mis-identify or reject the volume because the additional array information does not exist.

9.6.1 V5 Disk Serial Number Issue

Tru64 UNIX versions prior to V5 do not utilise disk serial numbers or WWIDs, whereas V5's support for multi-pathing requires unique disk serial number or WWIDs to correctly identify each discrete storage element.

Avanti provides for this in one of two ways:

1. By default, if a serial number field is left blank in the Avanti configuration file, a unique serial number is generated for the disk using the format SRL00.
2. The user can enter a unique serial number in the serial number field of a disk's advanced properties using the Configuration Utility. Disk serial number can by any unique value except in the case of HSZ22 disks. More on HSZ disks below.

Warning: Once a Tru64 UNIX configuration is built, do not modify the disk serial numbers. Doing so may cause Tru64 UNIX to think a disk has been changed out and invalidate the configuration.

The exception to the above rule is the HSZ disk type. When creating a custom disk size using the Avanti Configuration Utility, the disk type is assigned as HSZ22. If the disk serial number is left blank in the configuration file, Avanti assigns a fixed, valid, HSZ serial number to the first HSZ disk in the configuration: 00000002198505070c012c21bba00300. All subsequent HSZ disks with empty serial number fields are assigned the normal positional serial numbers 'SRLx0y' as above.

The structure of this special HSZ serial number is important to a small number of pre-V5 releases with certain patchkits when handling HSZ disks; for them at least one valid HSZ serial number/WWID must be present for those releases to recognize the HSZ type. However, if the HSZ type is not needed, use one of the other types.

When using disk serial numbers or WWIDs with V5 releases, care must be taken when migrating an existing installation to ensure the donor's disk serial numbers are carried forward to the migrated installation. Under V5, the WWIDs can be viewed via:

```
# hwmgr -show scsi -full
 …
        SCSI                 DEVICE    DEVICE  DRIVER NUM  DEVICE FIRST
 HWID:  DEVICEID HOSTNAME    TYPE      SUBTYPE OWNER  PATH FILE   VALID
PATH
------------------------------------------------------------------------
---
   34:  2        mareth      disk      none    0      1    dsk1
[0/1/0]

      WWID:04100038:"DEC     HSZ22
0000000219850507 0c012c21bba00300"
…
        SCSI                 DEVICE    DEVICE  DRIVER NUM  DEVICE FIRST
 HWID:  DEVICEID HOSTNAME    TYPE      SUBTYPE OWNER  PATH FILE   VALID
PATH
------------------------------------------------------------------------
---
   43:  9        mareth      disk      none    0      1    dsk8
[1/2/0]

      WWID:0410001f:"DEC     HSZ22           SRL0d02"
```

9.7 Maximum Physical Volume Size Under Tru64 UNIX

The version of Tru64 UNIX deployed dictates the maximum allowed physical volume size. The table below shows the maximum volume sizes allowed according to the Tru64 UNIX SPDs. Check the SPD for the version of Tru64 in use to verify actual supported maximum values.

Tru64 Version	Maximum Volume Size	Details
V3	128GB	V3.2C & V3.2G both state 128GB for AdvFs and UFS
V4	512GB	V4.0A states 512GB for AdvFs and UFS
V5	1TB	V5.0A, V5.1, V5.1B-6 state 1TB for AdvFs and UFS and add a total domain size of 16TB for AdvFs, that is 16 x 1TB volumes

9.8 TRU64 Unix Virtual Tape Support

Avanti virtual tape support for Tru64 UNIX is limited in scope.

- Tru64 UNIX V3.X - No support
- Tru64 UNIX V4.X - Limited support
- Tru64 UNIX V5.X - Limited support

Avanti does not implement or support tapes in their fullest form under Tru64 UNIX.

Read/write operations can be made to a virtual tape file via the /dev/*mt* or /dev/tape/* device special files. The resultant file cannot be detached from Avanti, nor can a new file be attached 'on the fly'. However, the tape file can be copied from the Windows side.

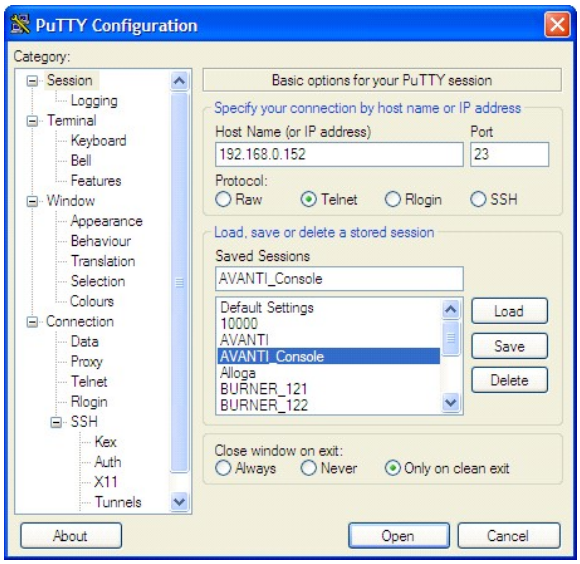
Tru64's mt unload or mt offline both serve to close the tape drive's file. They do not detach the tape file. Once closed, the file can be copied via Windows processes.

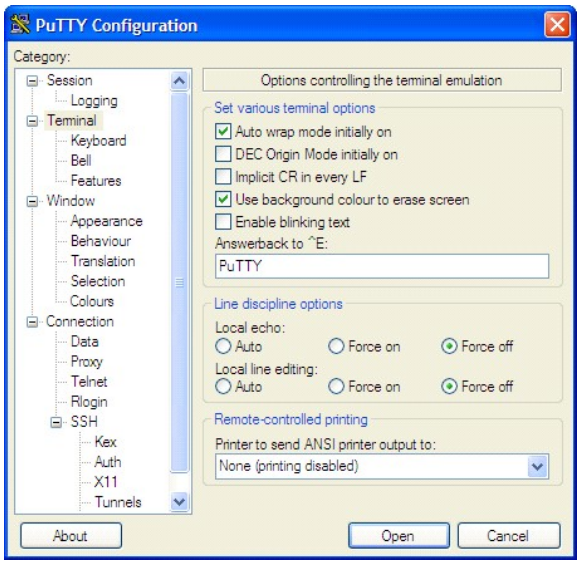
Limited Support

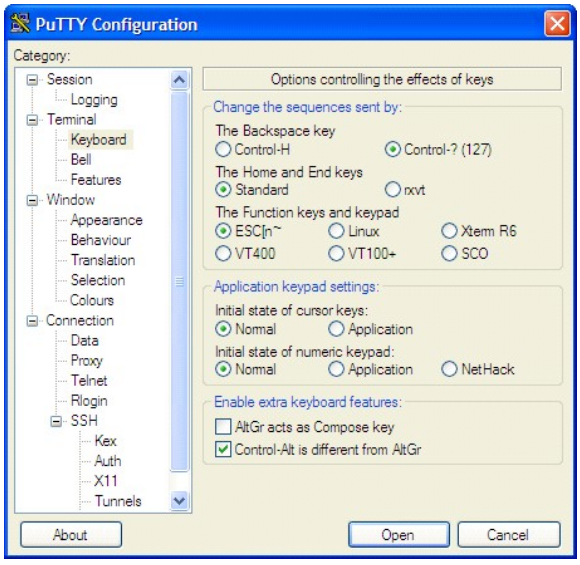
- Tape operations are only supported with explicitly set block sizes. I.e. Explicit I/O sizes MUST be specified on write operations (tar, cpio, vdump), and read operations MUST use dd with an explicit read size (eg. bs=50k) with a pipe to tar, cpio, vrestore, etc.
- The virtual tape does not differentiate between high and low densities.
- The virtual tape does not support compression.

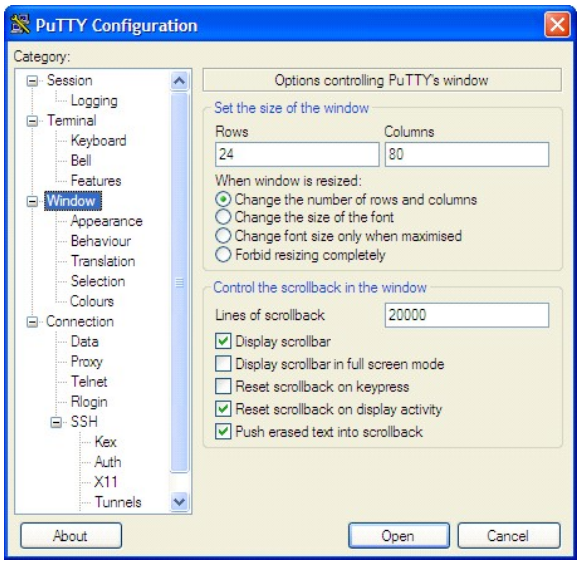
While the KZPCA wide controller is recommended as the HBA of choice, support for that device only arrives with Tru64 UNIX V4.0E.

9.9 Tru64 UNIX PuTTY Settings









Starting with a default PuTTY configuration, the settings shown below are typically used to access Tru64 UNIX on Avanti virtual Alpha systems via serial or telnet connections:

9.9.1 Transferring PuTTY Settings

PuTTY stores saved session settings in the Windows registry. The settings can be transferred to another Windows machine using regedit.

To export the PuTTY registry key on the source Windows machine, click on Start> Run> and enter the following regedit command in the run dialog box:

```
regedit /e "%userprofile%\desktop\putty-registry.reg"
HKEY_CURRENT_USER\Software\Simontatham
```

This command places the PuTTY registry key on the desktop in the file putty-registry.reg. Note that the name of the registry key (Simon Thatham) is the author of PuTTY.

Copy the putty-registry.reg file to the new Windows system. Right click the file and select the Merge option to add the key to the Windows registry. The PuTTY sessions defined in the transfer file will be added to any existing PuTTY sessions already saved on the machine.

Miscellaneous Notes

VIRTUAL ALPHA

10 Miscellaneous Notes

10.1 Boot Console Variables

Avanti emulates the Alphaserver 400/166 and its boot variables behave exactly as the real Alphaserver. To save the current boot variables between boots, ensure the emulator writes them to the NVRAM file via an exit.

```
>>> set bootdef_dev DKA0
>>> set boot_osflags 0
>>> exit
```

While running, Avanti saves the NVRAM contents to the NVRAM file approximately every 10 seconds.

10.2 Floating Point Values

Extensive tests have been run against the VAX and IEEE floating point instructions. There are two minor deviations under Avanti from the behavior of real AlphaServer 400 hardware.

1. With IEEE floating point instructions, the least significant bit of the fraction may differ between Avanti and a real AlphaServer 400. This is due to a rounding difference. There should be very few applications where this difference is significant.
2. After an overflow occurs in a multiplication or division instruction, the result (which is always invalid) differs between Avanti and a real AlphaServer 400. The overflow condition is properly signaled, so this should not be a problem for a well-behaved application.

10.3 Lantronix Serial to Ethernet Device Support

Lantronix serial devices such as the UDS-10, UDS-100, and UDS-1000 can be connected to a physical serial port assigned to Avanti and used to facilitate serial communications over IP. In some cases communicating with a Lantronix device causes buffer overruns. If this occurs, disable "Buffer Writes" in the Lantronix device configuration.

10.4 Graphics Support

X-windows graphics is supported using X-windows clients that run on PCs such as ReflectionX, eXceed, Excursion, and Xming. They provide the same functionality as an X-windows terminal.

10.5 Shared CD-ROMs

CD-ROM drives are read-only devices and can be shared between emulator instances. Use this feature with caution, as several emulators accessing a single CD-ROM could overload the underlying host I/O subsystem.

10.6 SCSI Controller Enumeration

By default, the AlphaServer 400 firmware used by Avanti enumerates the KZPAA controllers first, then the KZPCA controllers, regardless of the card's position on the PCI bus. This means that in a two-controller system consisting of KZPAA and KZPCA controllers, the KZPAA is assigned PKA and the KZPCA is assigned PKB, regardless of the PCI slot chosen in the Avanti configuration.

In a more elaborate example with 4 SCSI controllers:

```
Sl  Ca Device Name
ot rd

pciKZ PKC
6   PC
    A

pciKZ PKA
11 PA
    A

pciKZ PKD
12 PC
    A

pciKZ PKB
13 PA
    A
```

There is a firmware setting to change this behavior:

```
>>> set bus_probe_algorithm new
>>> init
```

after which the controllers will be enumerated in PCI bus order.

```
Sl  Ca Device Name
ot rd

pciKZ PKA
6   PC
    A

pciKZ PKB
11 PA
    A

pciKZ PKC
12 PC
    A

pciKZ PKD
13 PA
    A
```

The original settings can be restored using the following firmware commands.

```
>>> set bus_probe_algorithm old
>>> init
```

Troubleshooting

VIRTUAL ALPHA

11 Troubleshooting

This chapter outlines troubleshooting tools and known problems.

11.1 Log Files

If a problem with Avanti is encountered, log files are the first thing the Migration Specialties support team will request. Avanti offers the following logging options.

11.1.1 Avanti Log File

The Avanti log file is enabled by default. This file logs the emulator startup, shutdown, runtime, and error messages. 6 log files are retained by default. A version number appended to each log file name denotes past log files. A fresh log file is created each time the emulator is launched.

Current Log:	.log
Previous Log:	.log;-1
...	
Oldest Retained Log:	.log;-5

11.1.2 Terminal Log Files

The OPA0/tty00 and TTA0/tty01 serial terminals each provide a logging option. Terminal logging is disabled by default. Logging can be enabled on each terminal to a user specified file. Terminal logging records everything entered and received by the terminal to a text file. Care should be taken when using terminal logging, as the logs have the potential to become very large. A fresh log file is created each time the emulator is launched.

11.1.3 PuTTY Log Files

PuTTY sessions can be configured to log activity, as can most other terminal emulators. Such logs can be used in conjunction with the Avanti logs for problem diagnosis.

Tip: Capturing History

Two changes to the PuTTY configuration are useful when capturing information. Similar changes are recommended if using other terminal emulators.

1) Under Change Settings> Window modify "Lines of scrollback" to 2000. PuTTY then retains 2000 lines of history in the scroll back buffer.
2) Under Change Settings> Sessions select "Never" for "Close window on exit". This preserves the PuTTY window if the virtual Alpha crashes.
3) Once these changes have been made, highlight "Default Settings" under the Saved Sessions box and click the [Save] button. PuTTY will then retain the setting for future console sessions.

11.2 Avanti Crash Dump File

In the unlikely event of an emulator crash, Avanti may write a crash dump file comprised of the configuration file name and a .cra extension. The file will be located in the same folder as the configuration file. Certain VMS bug checks may also produce a Avanti crash dump file. If you are experiencing a problem with Avanti or VMS crashes, check for a .cra file. If found, send it to Migration Specialties for analysis.

Using the Halt option on the Avanti Control Icon in the Windows Notification Area always generates an Avanti crash dump file.

11.3 Halting a Hung System

Using the Halt option on the Avanti Control Icon in the Windows Notification Area forces output to the Avanti log file and generates an Avanti crash dump file.

11.4 Halt on Guest O/S Boot

We occasional see halts on boot of the guest O/S, particularly with older O/S releases. These are typically consistent with behavior on real hardware.

11.4.1 XDELTA not loaded

The halt is similar under OpenVMS and Tru64 UNIX, in that an XDELTA not loaded message is displayed. The solution is to issue an INIT command at the console prompt to clear the problem. The example below depicts the problem during an OpenVMS boot.

```
>>>b
(boot dka0.0.0.6.0 -flags 0)
block 0 of dka0.0.0.6.0 is a valid boot block
reading 904 blocks from dka0.0.0.6.0
bootstrap code read in
base = 1f2000, image_start = 0, image_bytes = 71000
initializing HWRPB at 2000
initializing page table at 1e4000
initializing machine state
setting affinity to the primary CPU
jumping to bootstrap code

    OpenVMS (TM) Alpha Operating System, Version V7.1

ff.halt code = 5
HALT instruction executed
PC = 110678
fe.fd.fc.fb.fa.f9.f8.breakpoint at PC 1108e0 desired, XDELTA not
loaded
f7.f6.f5.
ef.df.ee.ed.ec.f4.eb.ea.e9.e8.e7.e6.....e5.
V7.0-9, built on Mar 18 1999 at 13:25:37
>>>
```

11.4.2 Unexpected exception/interrupt through vector 420

The unexpected exception/interrupt through vector 420 problem typically occurs during the O/S boot process and is well documented in real Alpha hardware behavior. The system self corrects by rebooting, as shown in the example below.

```
ff.fe.fd.fc.fb.fa.f9.f8.f7.f6.f5.
ef.df.ee.ed.ec.f4.eb.ea.e9.e8.e7.e6.....e5.
V7.0-9, built on Mar 18 1999 at 13:25:37

CPU 0 booting

(boot dka200.2.0.6.0 -flags 0)
block 0 of dka200.2.0.6.0 is a valid boot block
reading 1226 blocks from dka200.2.0.6.0
bootstrap code read in
base = 1f2000, image_start = 0, image_bytes = 99400
initializing HWRPB at 2000
initializing page table at 1e4000
initializing machine state
setting affinity to the primary CPU
jumping to bootstrap code
unexpected exception/interrupt through vector 420
process idle, pcb = 001159D0

 pc: 00000000 00116A24  ps: 00000000 00001F04
 r2: 00000000 001112A0  r5: 00000000 00001F04
 r3: 00000000 00022650  r6: 00000000 00001259
 r4: 00000000 000EE658  r7: 00000000 0000000A

exception context saved starting at 00116900

GPRs:
  0: 00000000 00000000  16: 00000000 00116A08
  1: 00000000 000001F4  17: 00000000 00006000
  2: 00000000 000ED330  18: 00000000 000384C0
  3: 00000000 00076CC0  19: 00000000 0012DD10
  4: 00000000 000EE658  20: 00000000 0012DD10
  5: 00000000 00038B88  21: 00000000 00000000
  6: 00000000 00001259  22: 00000000 00116A08
  7: 00000000 0000000A  23: 00000000 00000001
  8: 00000000 00110670  24: 00657174 00000000
  9: 00000000 00110678  25: 00000000 00000001
 10: 00000000 000D0DE8  26: 00000000 00116A20
 11: 00000000 00000001  27: 00000000 00111940
 12: 00000000 001159D0  28: 00000000 00038BD8
 13: 00000000 00000010  29: 00000000 00000000
 14: 00000000 00000011  30: 00000000 00116A40
 15: 00000000 00000000

dump of active call frames:

breakpoint at PC 1108e0 desired, XDELTA not loaded
ff.fe.fd.fc.fb.fa.f9.f8.f7.f6.f5.
ef.df.ee.ed.ec.f4.eb.ea.e9.e8.e7.e6.....e5.
V7.0-9, built on Mar 18 1999 at 13:25:37

CPU 0 booting
```

```
(boot dka200.2.0.6.0 -flags 0)
block 0 of dka200.2.0.6.0 is a valid boot block
reading 1226 blocks from dka200.2.0.6.0
bootstrap code read in
base = 1f2000, image_start = 0, image_bytes = 99400
initializing HWRPB at 2000
initializing page table at 1e4000
initializing machine state
setting affinity to the primary CPU
jumping to bootstrap code


    OpenVMS (TM) Alpha Operating System, Version V8.3
    © Copyright 1976-2006 Hewlett-Packard Development Company, L.P.
```

11.5 Problem Submission

Avanti, AvantiFlex, and FreeAXP users with current support contracts can contact Migration Specialties directly for support. Users without support contacts can seek help at the FreeAXP Support Forum (http://vmshobbyist.org/forum/viewforum.php?forum_id=163).

When posting a problem report, please provide the following information:

1) Avanti configuration file (.cfg)
2) Avanti log file (.log)
3) Avanti crash dump file (.cra), if produced.
4) Host System Information
    a) Windows O/S
    b) Hardware information:
        i) CPU Type
        ii) Number of CPU's
        iii) Memory Installed

Glossary

VIRTUAL ALPHA

12 Glossary

Avanti	Commercial Alpha emulator
AvantiFlex	Commercial Alpha emulator
FreeAXP	Free Alpha emulator
GUI	Graphical User Interface
JIT	Just-In-Time compilation
NIC	Network Interface Card
NVRAM	Non-Volatile RAM
Shadow Code	Avanti pre-processes Alpha instructions, optimizing them to achieve better performance upon execution. These optimized instructions are referred to as shadow code. Shadow code is retained in memory until it is executed.
Shadow Memory	Holds shadow code.
SimH VAX	Free open source VAX emulator
SPD	Software Product Description: A document providing detailed product specifications.
TOY	Time-Of-Year

Avanti Emulator Files

VIRTUAL ALPHA

13 Avanti Emulator Files

Each Avanti emulator configuration has a set of files associated with it. By default, the files are stored in the same folder as the configuration file and use the configuration file name. If the emulator configuration is moved, these files should be moved with it.

File Extension	Example	Function	Created by...
.cfg	Alpha01.cfg	Configuration file. Defines the emulator's virtual hardware.	User
.log	Alpha01.log	Avanti og file. Records launch and runtime information during an emulator session.	Avanti at configuration launch. The file is replaced each time the configuration is launched. By default, logging is enabled and 6 versions of the log file are retained.
.nvr	Alpha01.nvr	NVRAM information storage. This file takes the place of non-volatile RAM. It stores boot configuration parameters.	Avanti at normal exit.
.toy	Alpha01.toy	Time-Of-Year information storage. The toy file stores the date and time offset for the virtual Alpha system	Avanti at normal exit.

Disk and CD-ROM image files are also a user-defined part of an Avanti configuration. If these files are moved, the associated Avanti configuration file must be updated.

Avanti Command Line Options

VIRTUAL ALPHA

14 Avanti Command Line Options

Avanti supports the following command line options. Options can be shortened to the least number of unique characters. I.e. /C = /CONFIGURATION_FILE

Command Line Option	Descrption
/CONFIGURATION_FILE="filename"	5.6 Launching Avanti Manually 54
/INSTALL	5.13 Installing Avanti as a Service 58
/UNINSTALL	5.13.2 Avanti Service Removal 59

- D -

- E -

- F -

- G -

- H -

- I -

- J -

- K -

- L -

- M -

- N -

- O -

- P -

- Q -

- R -

- S -

- T -

- U -

- V -

- W -

- X -