



HowTo VMware und FLI4L für DSL

Wie kann ich FLI4L in VMware laufen lassen?

Version vom 21. Juni 2002
Ergänzt am 22. Februar 2004

Zuerst einmal eine Warnung: FLI4L in VMware ist nicht so sicher wie auf einem eigenen Rechner.

Für Nutzung einer ISDN Karte hatte ich keine Möglichkeit zum Testen. Ob es analog zu meiner Vorgehensweise gilt, kann ich deshalb weder bestätigen noch verneinen. Aber insbesondere für die ISDN Karte gilt folgender Warnhinweis GANZ BESONDERS:

Es existiert im Client Rechner auch die Netzwerkkarte für die DSL Verbindung bzw. die ISDN Karte.

Bei Nutzung dieses HowTos geht ein großer Sicherheitsaspekt verloren.

Ein bösartiger Virus könnte u.U. die bestehende Netzwerk Karte zum DSL Router bzw (noch gefährlicher:) ISDN Karte erkennen und dies ausnutzen, trotz FLI4L!!!

Unser FLI4L kann seine Sicherheit nur dadurch erreichen, dass die Hardware zum Internet hin weitgehend abgeschirmt ist vor 0190-Dialern. Wenn die ISDN Karte allerdings im gleichen PC steckt, auf dem auch das (Windows) Betriebssystem läuft, dann bestünde die Möglichkeit, dass die ISDN Karte direkt von einem bösen Programm angesprochen wird, und FLI4L keine Möglichkeit hat, dies zu verhindern.

Nochmal die Warnung: meine Vorgehensweise ist nur als "Backup" Lösung (fallback) gedacht, falls der Router mal überraschend ausfallen sollte, z.B. durch Defekt von Festplatte, Prozessor, RAM oder Mainboard!

Zusätzlich kommen vielleicht noch diverse Testzwecke in Frage, in denen man sich der Hilfe von VMware bedienen könnte, um den originalen Router lauffähig zu halten und ein neues OPT erst mal in einer virtuellen Umgebung zu testen.

Ich konnte VMware nur auf einem Rechner während meines Praktikums nutzen. Mittlerweile, da dieses HowTo gelesen wird, wird mein Praktikum wohl schon vorbei sein und ich kann nicht versprechen, dass ich noch auf VMware zugreifen kann. Fragen hierzu versuche ich aber trotzdem gerne zu beantworten.

Mir stand VMware 3.0.0 zur Verfügung. Alle Hinweise in diesem HowTo beziehen sich auf diese Version. Ältere Versionen von VMware werden sehr wahrscheinlich noch nicht über gewisse Funktionen verfügen, da diese aktuelle Version von VMware einige Neuerungen bietet.

Ich habe die Lösung mit 3 Netzwerkkarten gewählt, weil ich es für anschaulicher hielt. Aufgrund vieler EMail, die mich aufgrund der ersten Version dieses HowTos erreichten, habe ich es auch mit nur einer einzigen physikalischen Netzwerkkarte im Host geschafft. Ich habe ursprünglich die Lösung über 3 Netzwerkkarten gewählt, weil hier anschaulich verdeutlicht werden kann, wie die beiden Netzwerkkarten aus dem Router (oder 2 völlig andere) in den Client eingebaut und dort von FLI4L verwendet werden.

Am Ende des HowTos habe ich einen Abschnitt "Nur eine Netzwerkkarte" erstellt. Dort findet der interessierte Leser eine Konfiguration, bei der nur eine einzige Netzwerkkarte im Host sein muss. Dies ist praktisch, wenn man seinen Router bei einem Bekannten vergessen hat und abends noch mal schnell ins Internet will, der (Windows) Rechner aber für die Einwahl über den FLI4L konfiguriert ist und man keine Einwahlsoftware oder -einstellungen durchführen will.

Ferner ist die Sicherheit über einen emulierten FLI4L immernoch deutlich höher, als ein direkter Anschluss an das DSL oder ISDN.

Aber: die Sicherheit ist immernoch deutlich niedriger als die von einer *echten* FLI4L Lösung auf einem eigenen Rechner.

Zuallererst installieren wir, falls noch nicht geschehen, die Software VMware.

benötigte Hardware

Wir benötigen im Clientrechner bei meiner Vorgehensweise 3 Netzwerkkarten.

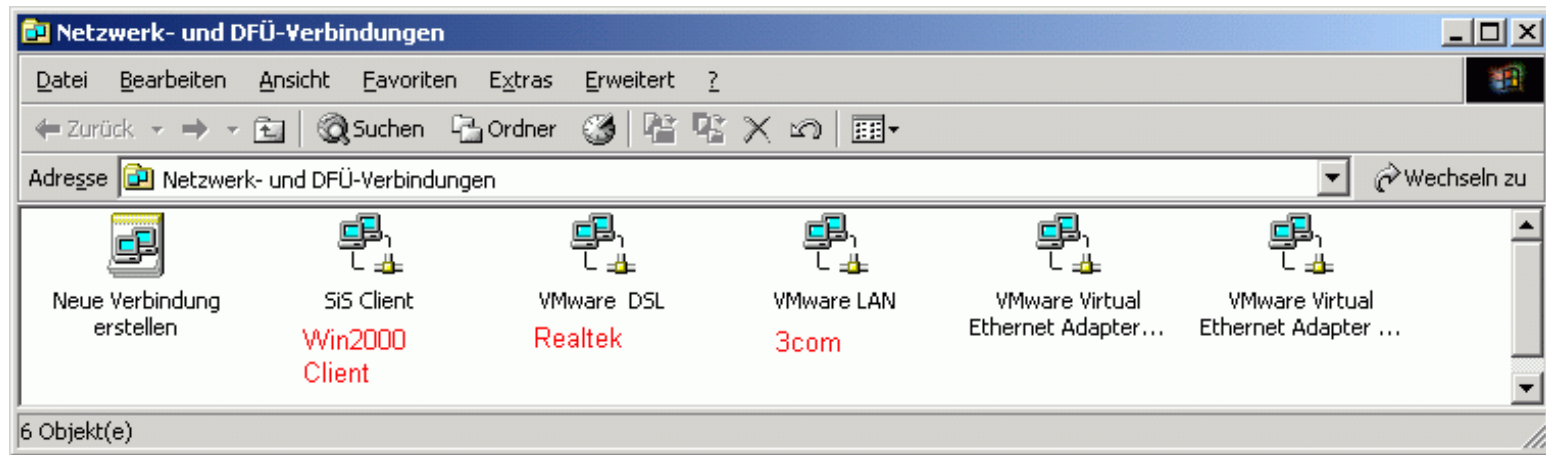
Eine Netzwerkkarte ist sowieso schon in unserem Client drin.

Wir fügen nur noch 2 NICs hinzu. Das können sein:

- die beiden NICs aus unserem Router
- eine NIC aus dem Router und eine neue
- oder aber 2 völlig andere (evtl. neue oder gebrauchte)

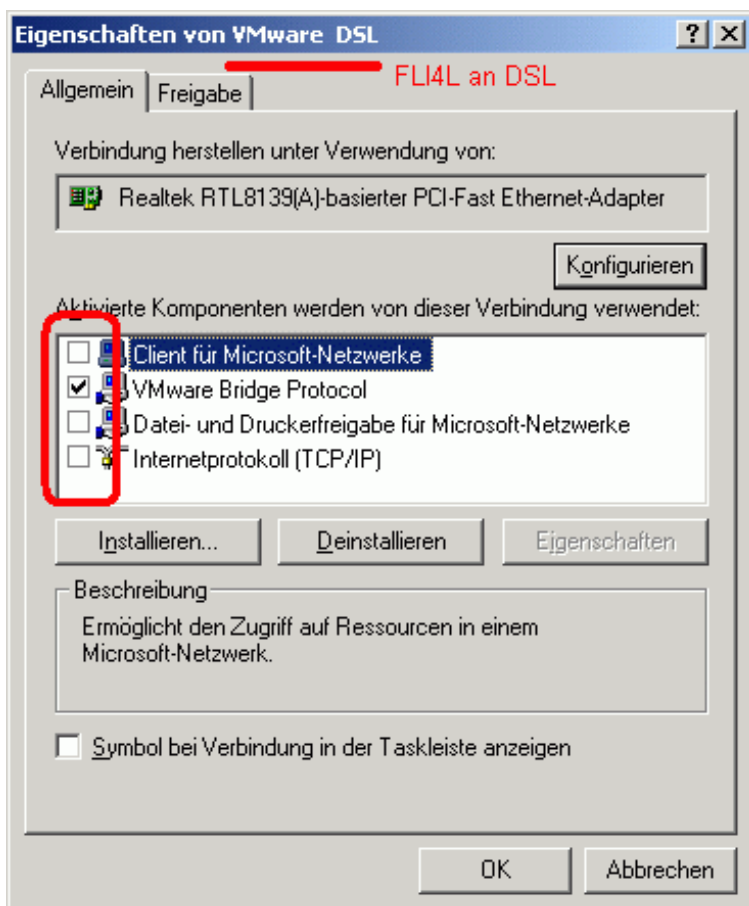
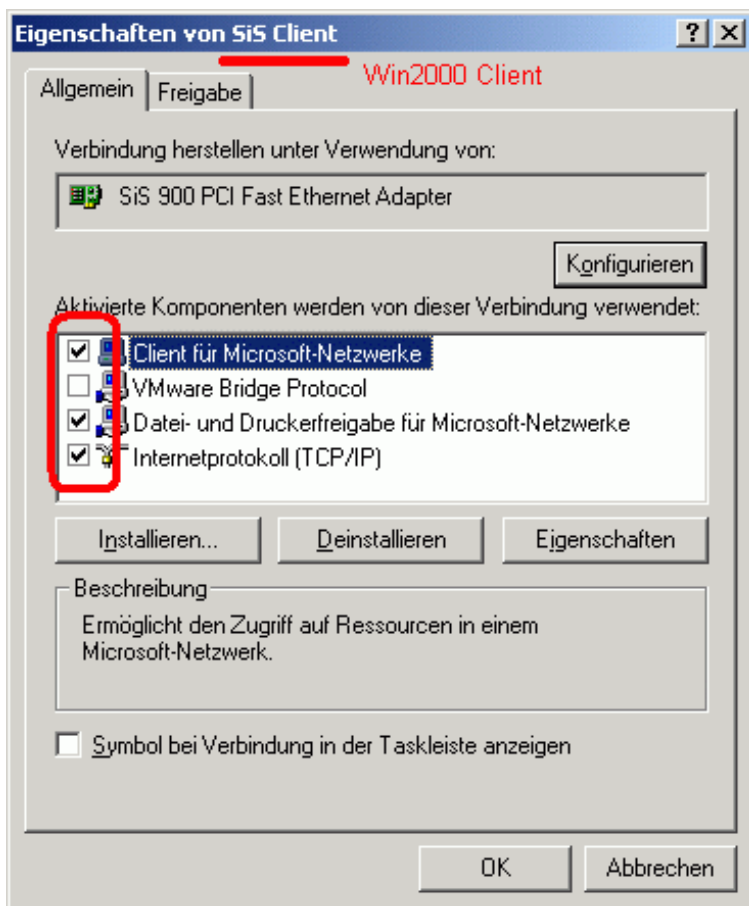
Wir haben dann in unserem Client 3 Netzwerkkarten, die die folgenden Funktionen übernehmen sollen:

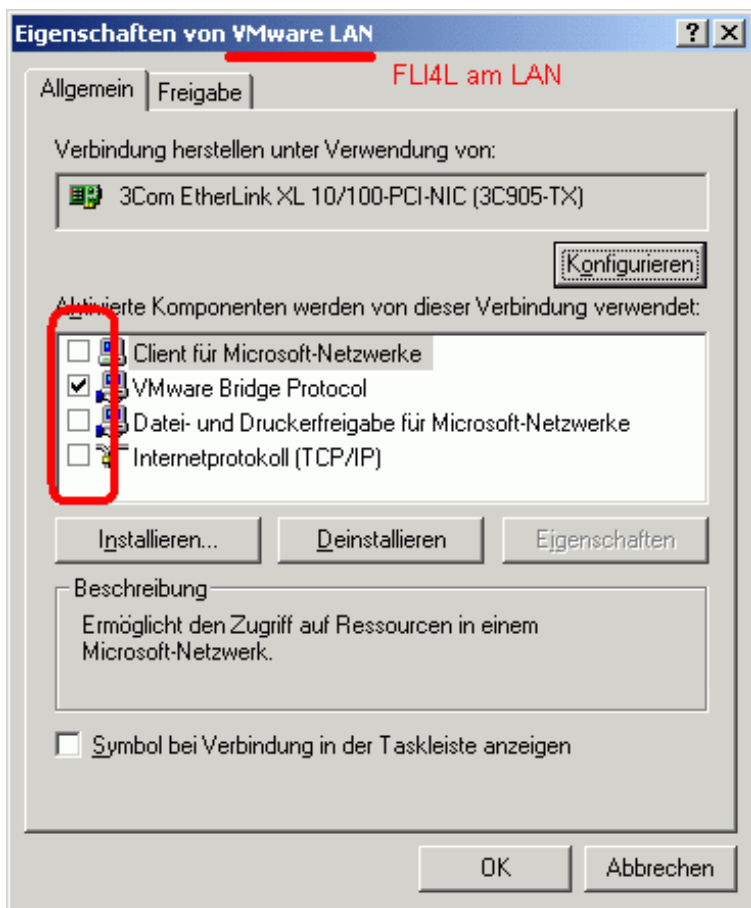
- 1 für DSL (neu)
- 1 für FLI ins Netzwerk (neu)
- 1 für den Client unter Windows (unsere bisherige)



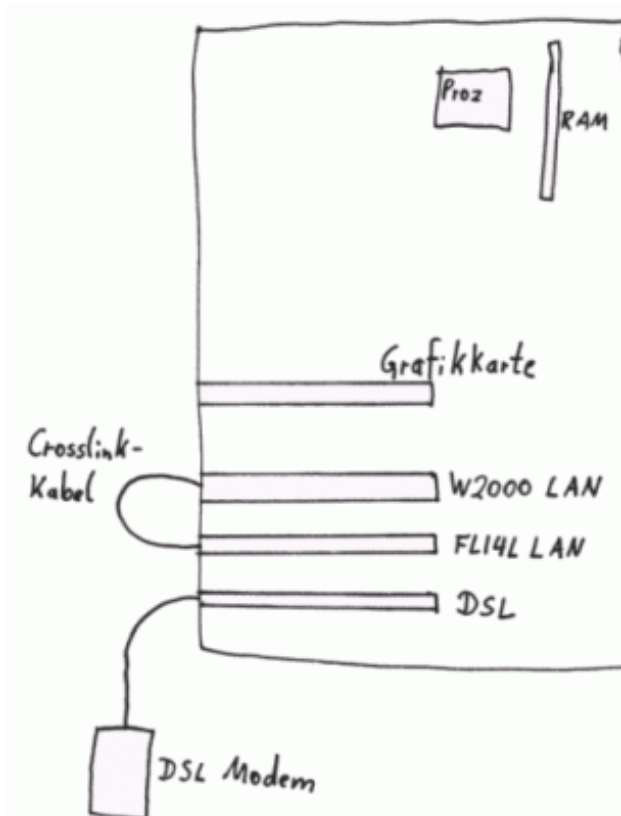
Vorbereitung der Hardware

Nun erfolgt die Protokollbindung wie hier:

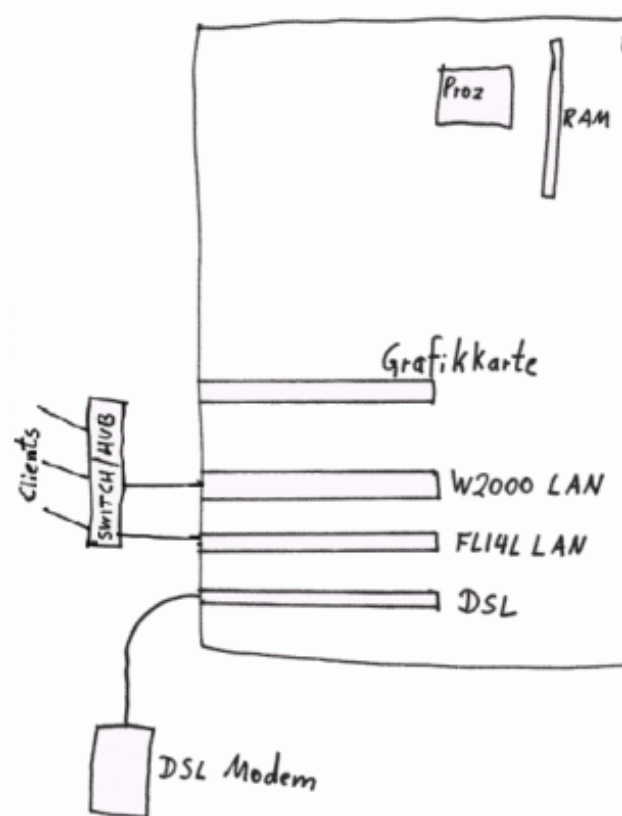




Die Verkabelung per Crosslink Kabel soll diese Zeichnung verdeutlichen:

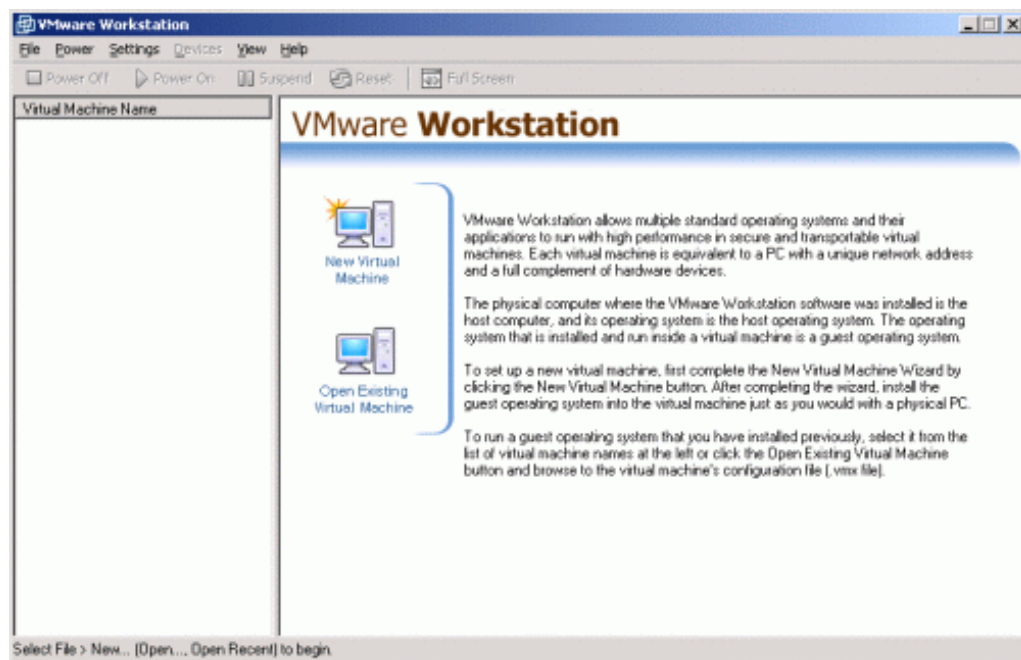


Alternativ kann man es auch über einen Switch oder Hub lösen:

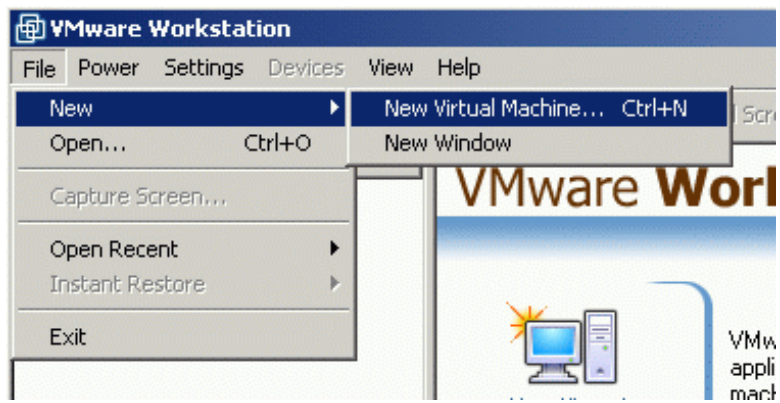


Konfiguration von VMware

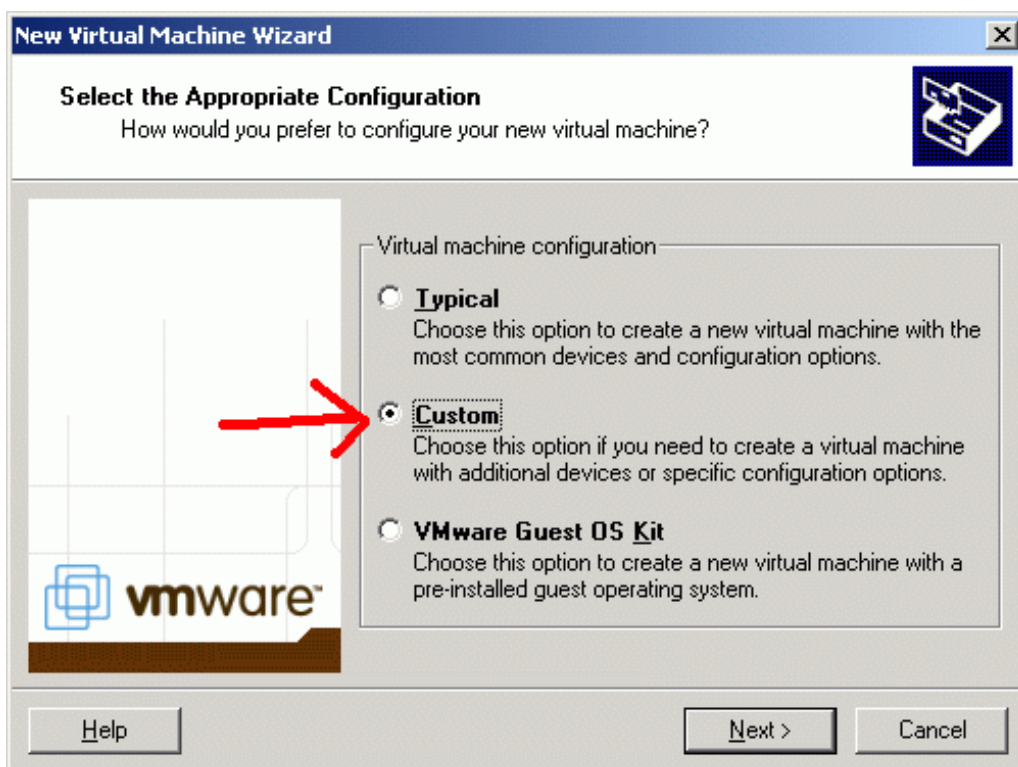
Wenn VMware komplett installiert ist und zum ersten Mal gestartet wird, bietet sich uns folgendes Bild:



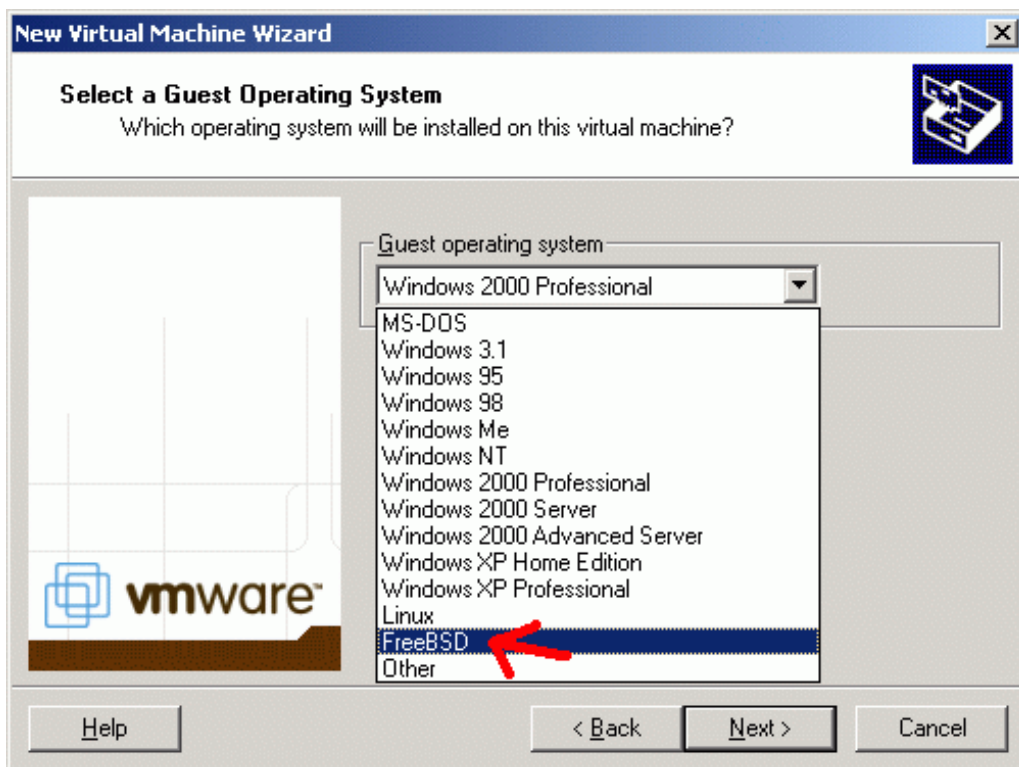
Wir wählen im Menü FILE den Punkt NEW, dort dann New Virtual Machine:



Im Wizard wählen wir die Funktion Custom:

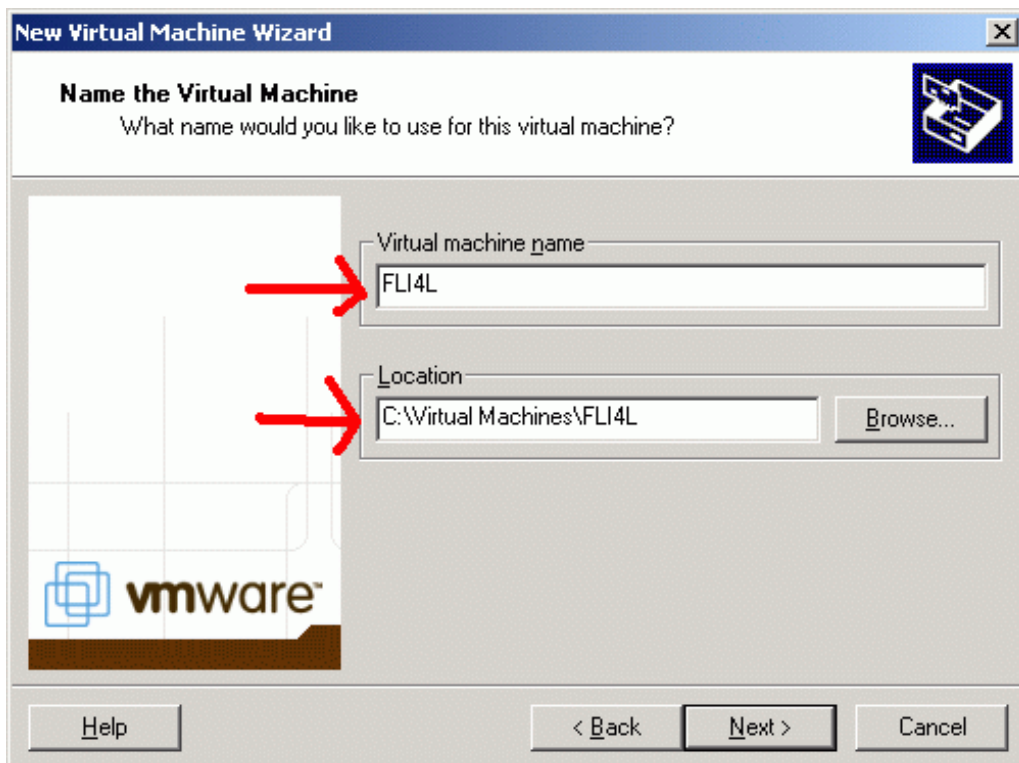


Als Betriebssystem wähle ich FreeBSD, ich bin mir aber nicht sicher, ob es nicht auch mit anderen Systemen ginge. Als ich Linux wählte, wurde mir eine SCSI Platte erstellt. Ich wollte aber eine IDE Platte als virtuelle Platte haben. Bei FreeBSD wurde mir standardmässig schon eine IDE Platte erstellt. Prinzipiell ist das Betriebssystem, das hier gewählt wird, relativ egal. Da wir mit FLI4L nur rudimentäre Funktionen der Hardware benutzen, benötigen wir keine Optimierung an ein Betriebssystem. Wie weiter unten erläutert hängt vom gewählten Betriebssystem stark die Variante der VMtools ab. Diese VMtools benötigen wir aber nicht. Sie bieten uns höhere Disk-Performance, höhere Bildschirmauflösung und besserer Maus Support. Wer unbedingt meint, dass er eine dieser 3 Vorteile benötigt, kann gerne die von VMware angebotenen Betriebssysteme durchprobieren. Ich für meinen Teil habe FreeBSD gewählt und es hat funktioniert. Aber hier bin ich gerne bereit, andere Meinungen anzunehmen, wenn mir jemand die Vorteile seiner Wahl darlegen will.



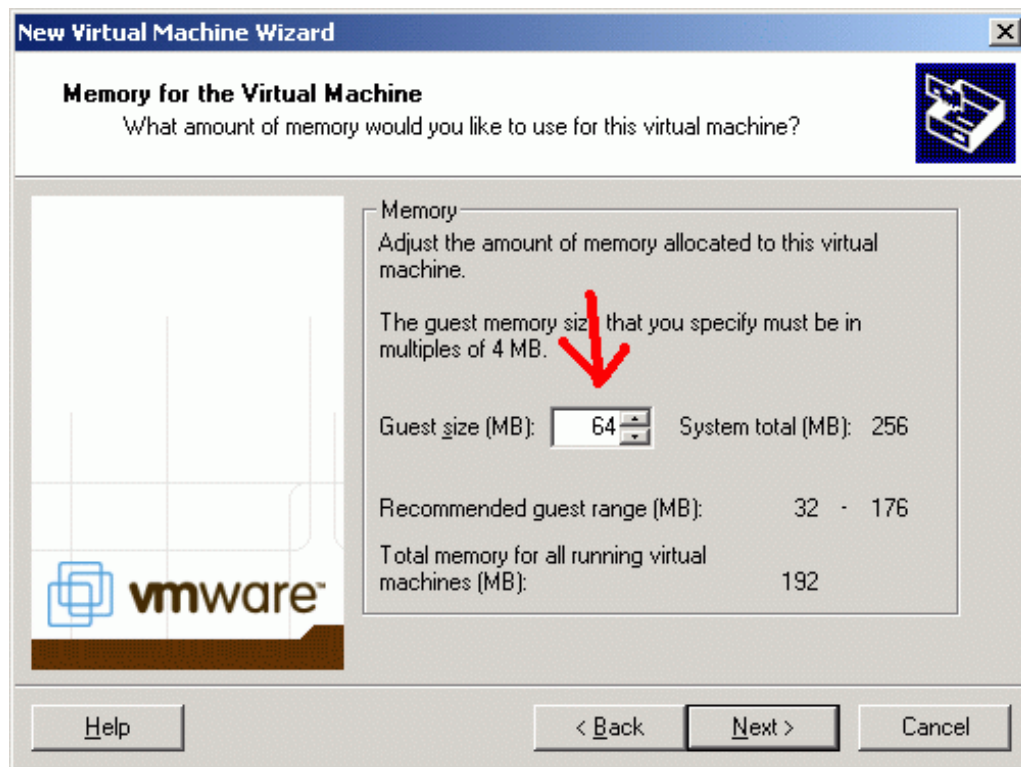
Als Virtual machine name geben wir einen treffenden Namen ein, z.B. FLI4L. Der Name ist egal, er dient nur zu unserer Information und besserer Übersicht.

Als Location wählen wir eine Partition und ein Verzeichnis unserer Wahl aus.



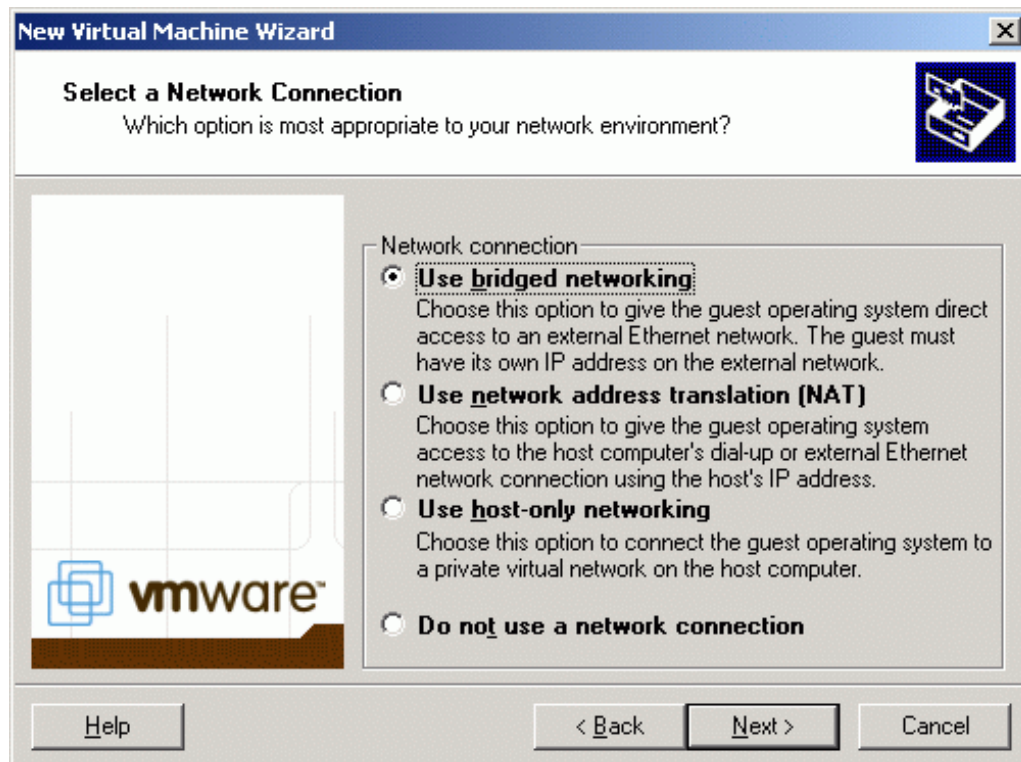
Die Menge an RAM können wir abhängig vom vorhandenen freien RAM auf unserem Client recht großzügig auf 64 MB einstellen. Selbstverständlich gingen auch 96 MB oder gar 128 MB. Aber dies dürfte für Routing-Zwecke stark überdimensioniert sein...

Wir wählen 32 MB oder 64 MB aus. Später können wir bei Bedarf diese Größe immernoch erhöhen oder verringern. Also brauchen wir uns jetzt damit nicht den Kopf zu zerbrechen.



Im nächsten Fenster wählen wir "Use bridged networking".

Es ist völlig egal, was genau das bedeutet, oder was wir da wählen. Wir ändern es gleich sowieso völlig anders ab. Aber wir müssen dieses Fenster irgendwie aktiv hinter uns bringen, also wählen wir gleich die erste Möglichkeit und dann NEXT.



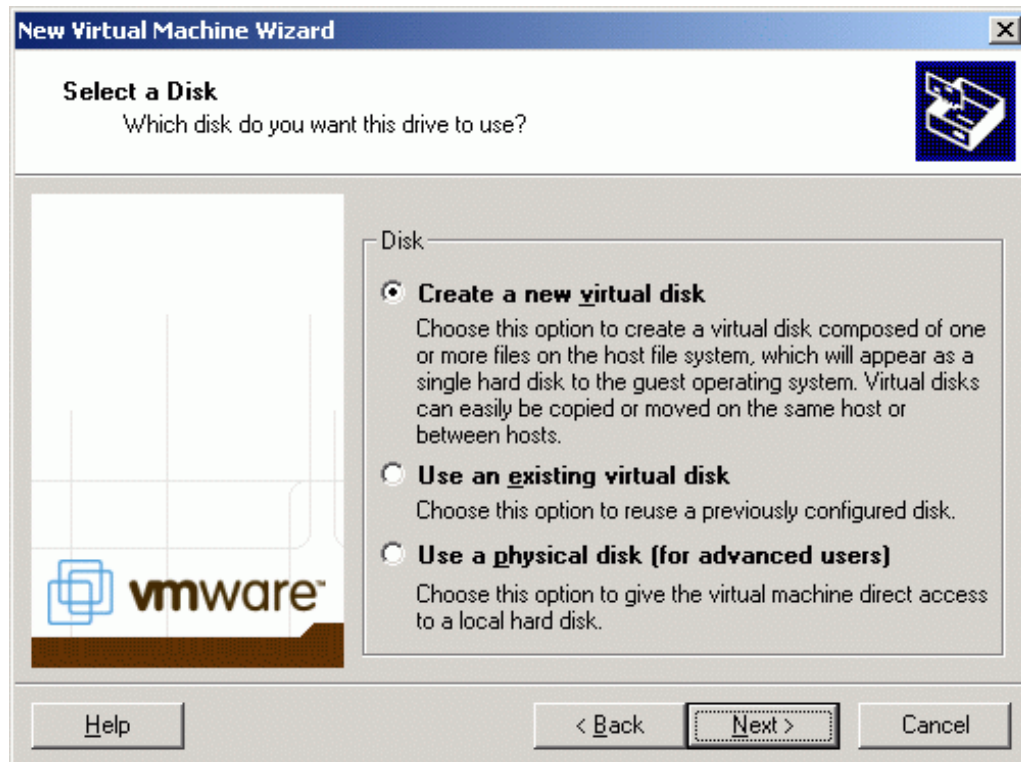
Hier werden wir nach der Art der Festplatte gefragt. Wir haben hier die Möglichkeit, eine neue "virtuelle" Festplatte zu erstellen. Dies ist ein einziges File auf unserer physikalische Festplatte. Hierin wird die Festplatte in VMware emuliert. Man kann es als Image der virtuellen Platte bezeichnen. Diese virtuelle Platte ist völlig abgeschirmt von der physikalischen Festplatte. Tools für den Bootsektor oder Partitionsprogramme konnten in VMware gestartet nichts auf meiner physikalischen Festplatte anrichten. Aber wie immer bei heiklen Aktionen gilt auch hier:

Mach ein Backup deiner Festplatte, bevor du ein Risiko eingehst!

Die zweite Wahl wäre eine bereits erstellte virtuelle Disk zu benutzen, für Leute, die evtl. früher schon mal eine erstellt, oder eine von einem anderen Rechner rüberkopiert haben.

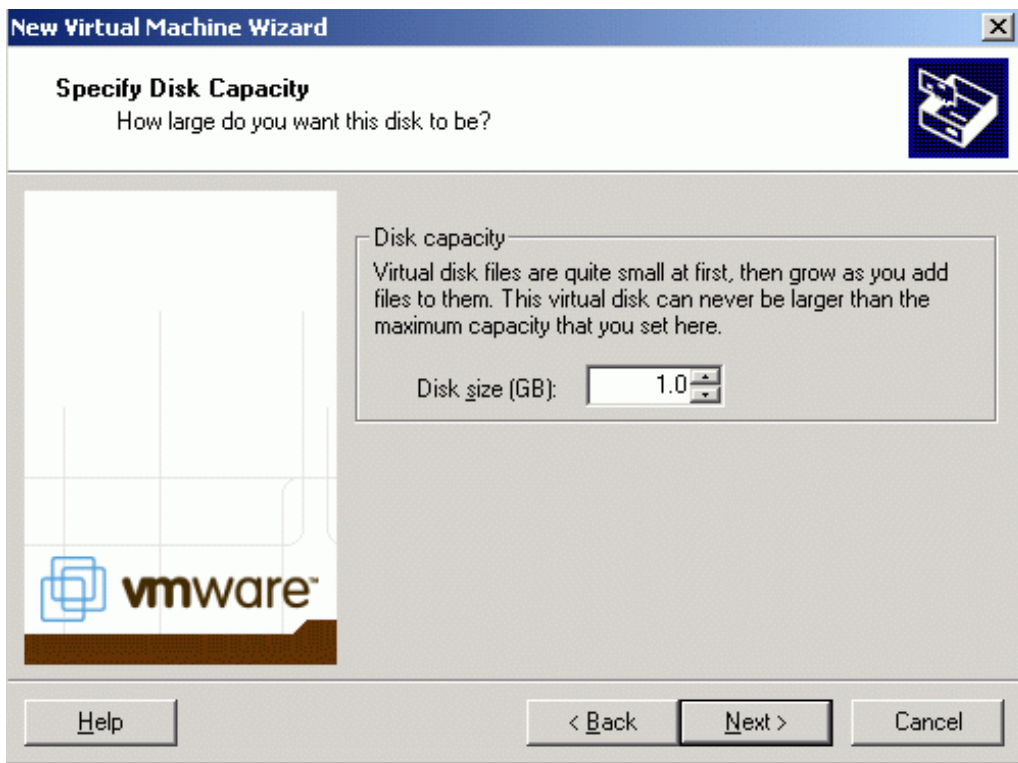
Die dritte Wahl ist für Profis. Hier kann man eine tatsächliche, echte Partition auf der Festplatte wählen. Diese Lösung empfiehlt sich aber nur für Profis, die genau wissen, was sie da tun und wie sie es tun!

Alle anderen wählen hier bitte die erste Möglichkeit:

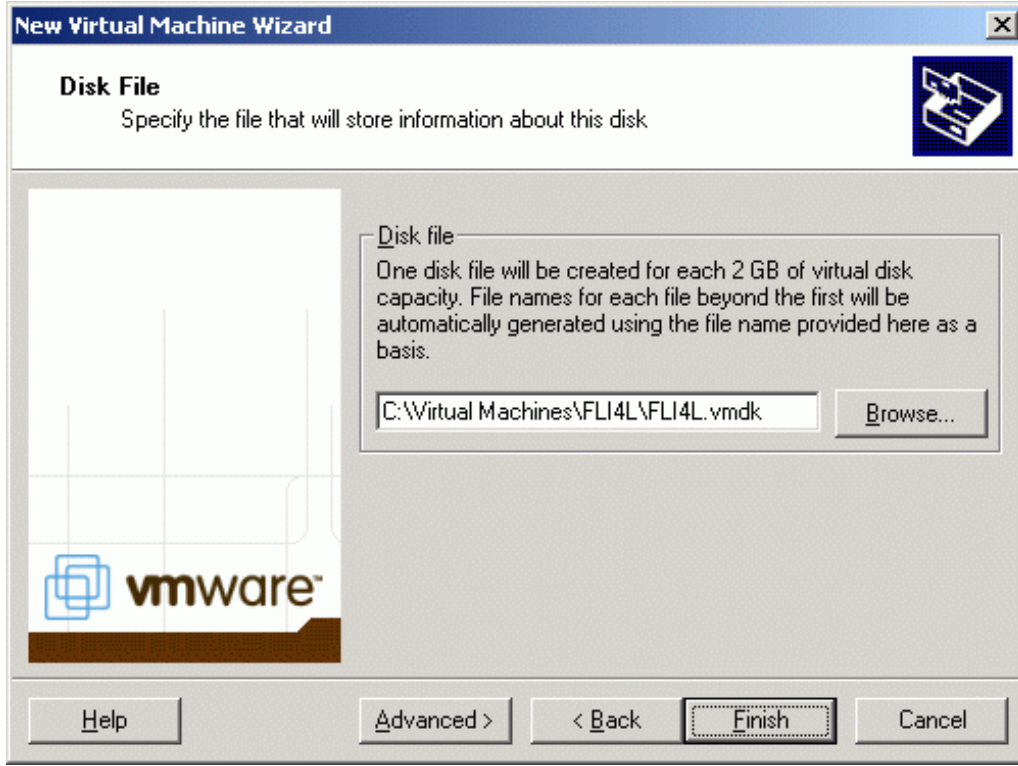


Als nächstes werden wir nach der Größe der virtuellen Festplatte gefragt. Je nach dem, was wir später mit dem Router vorhaben, wählen wir hier eine entsprechende Größe aus. Diese Festplattengröße ist allerdings nur die maximale Größe. Real verbrauchen wird unser File auf unserer "echten" Festplatte aber nur so viel Platz, wie wirklich Daten in dem File gespeichert sind. Wählen wir z.B. eine Partitionsgröße von 1 GB aus und speichern nur den kompletten FLI und ein paar OPT Pakete darin ab, dann wird das File lediglich ca. 8 MB groß. Halt eben nur so viel, wie Daten darin vorkommen. Es zählt nur der benötigte Speicherplatz.

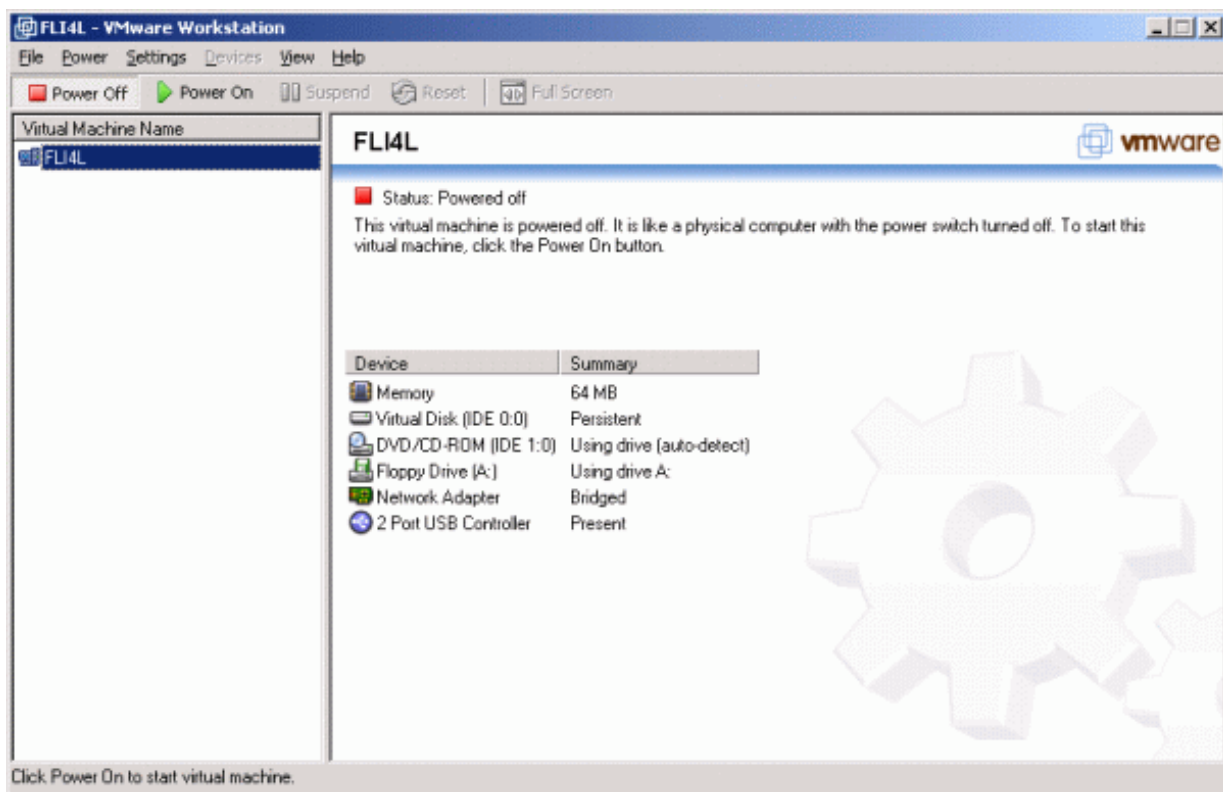
Hier sollte für den normalen Gebrauch 1 GB mehr als genügen:



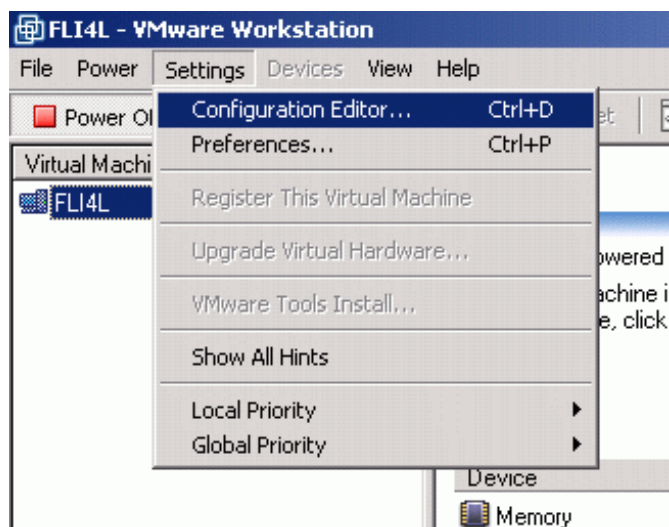
Im nächsten Fenster geben wir an, wohin das virtuelle Festplatten-Image geschrieben werden soll. Hier bietet sich das gleiche Verzeichnis wie vorhin an:



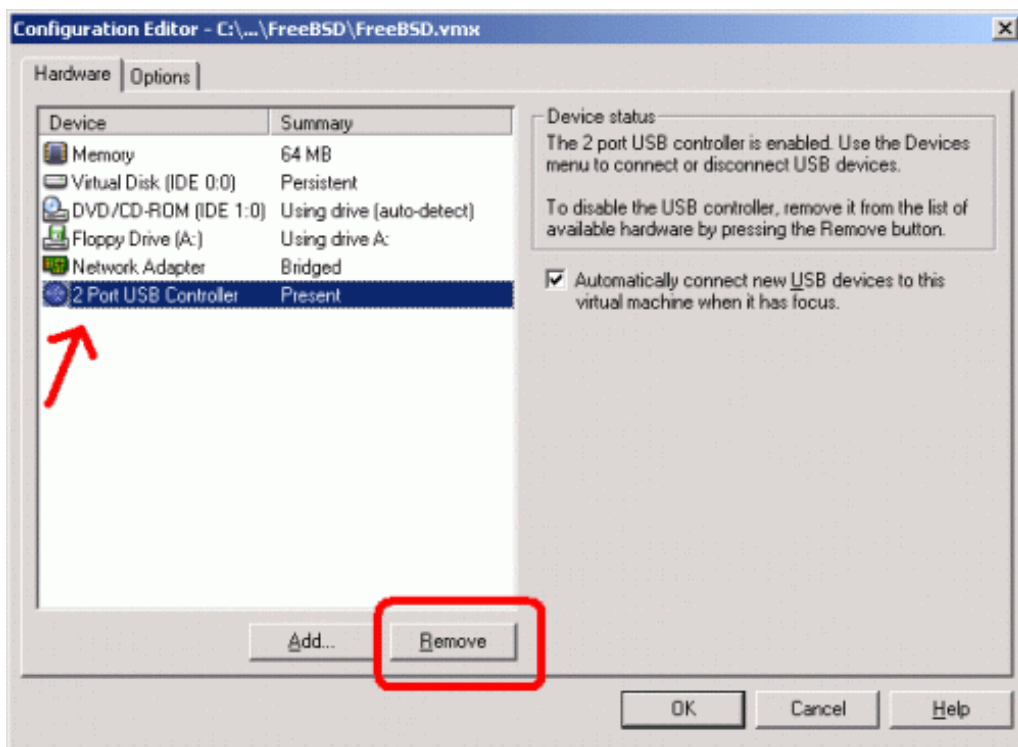
Nun finden wir uns im Hauptfenster von VMware wieder:



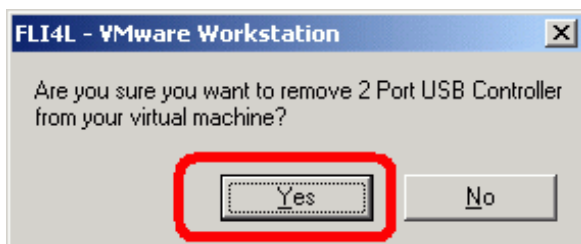
Jetzt gehen wir in den Configuration Editor



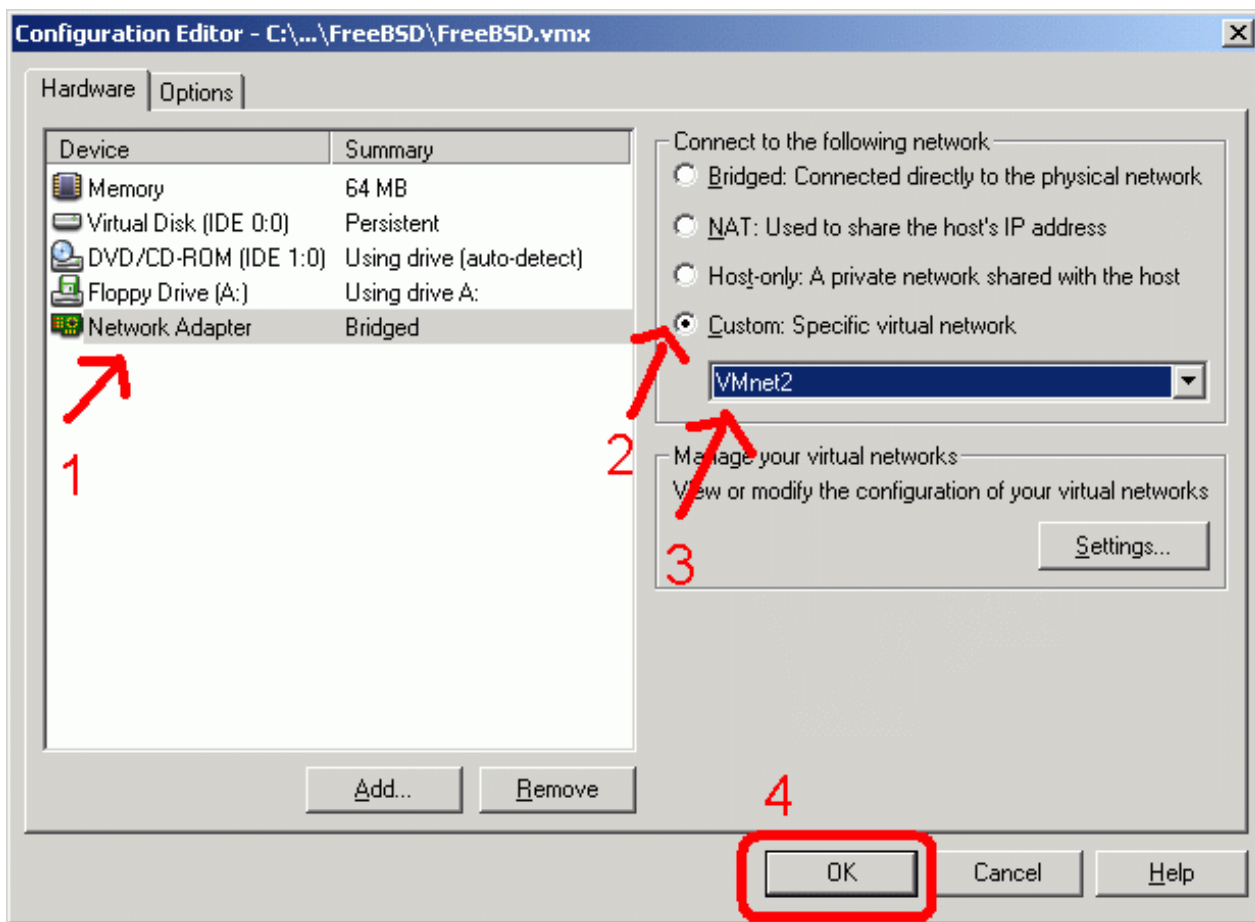
Wir wählen den USB Controller, falls er bei uns auftaucht. Denn in FLI benötigen wir kein USB (noch nicht). Sollten wir im Laufe unseres Lebens USB mit FLI4L benötigen und sollte es dann schon implementiert sein, kann man den USB dann immernoch nachträglich hinzufügen. Vorerst aber benötigen wir ihn nicht. Was ich nicht benötige, schalte ich ab oder entferne es.



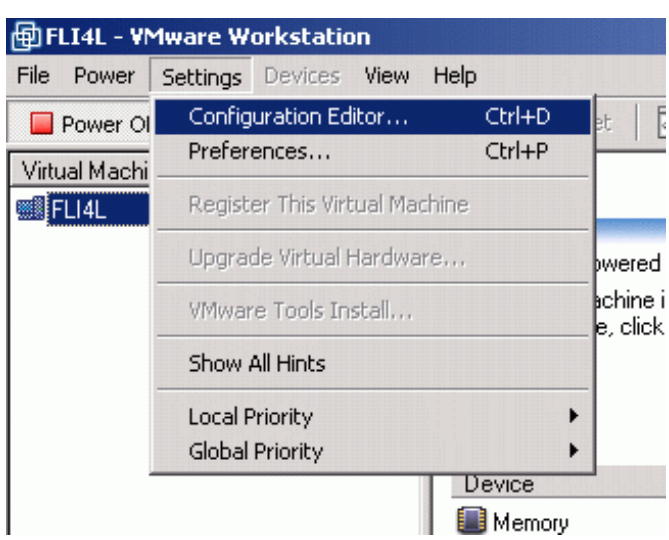
Wir bestätigen mit Yes



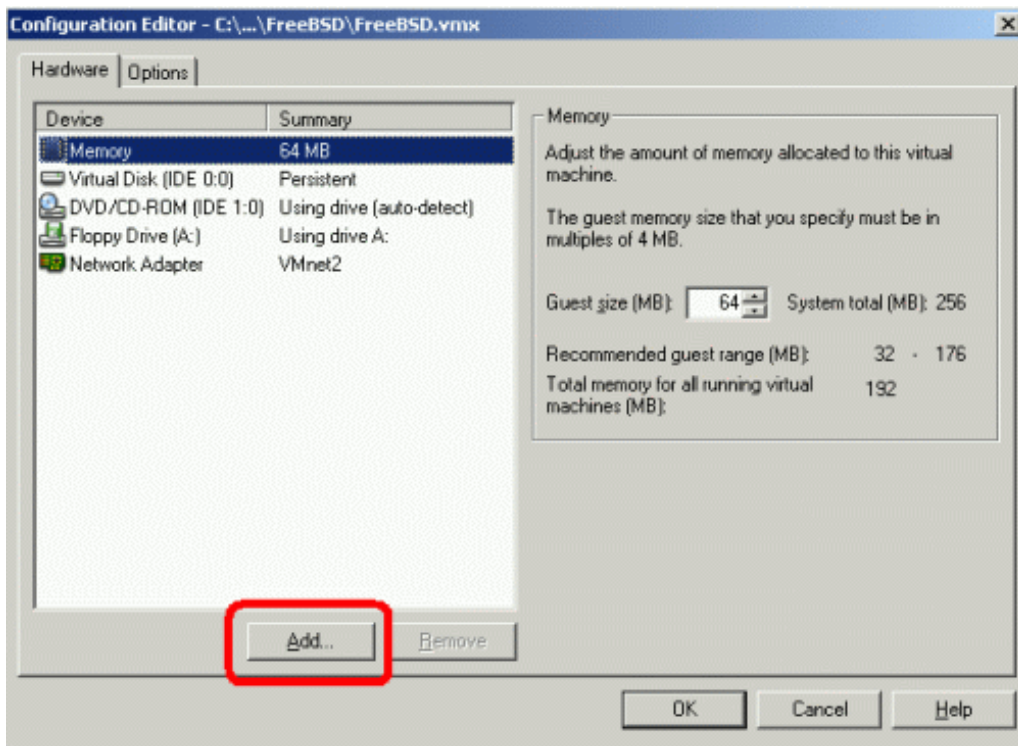
Nun wählen wir den Network Adapter aus (Schritt 1),
dann wählen wir bei Schritt 2 "Connect to the following network" den Punkt: "Custom" aus.
Nun als Schritt 3 wählen wir "VMnet2"
Als Schritt 4 bestätigen wir mit OK



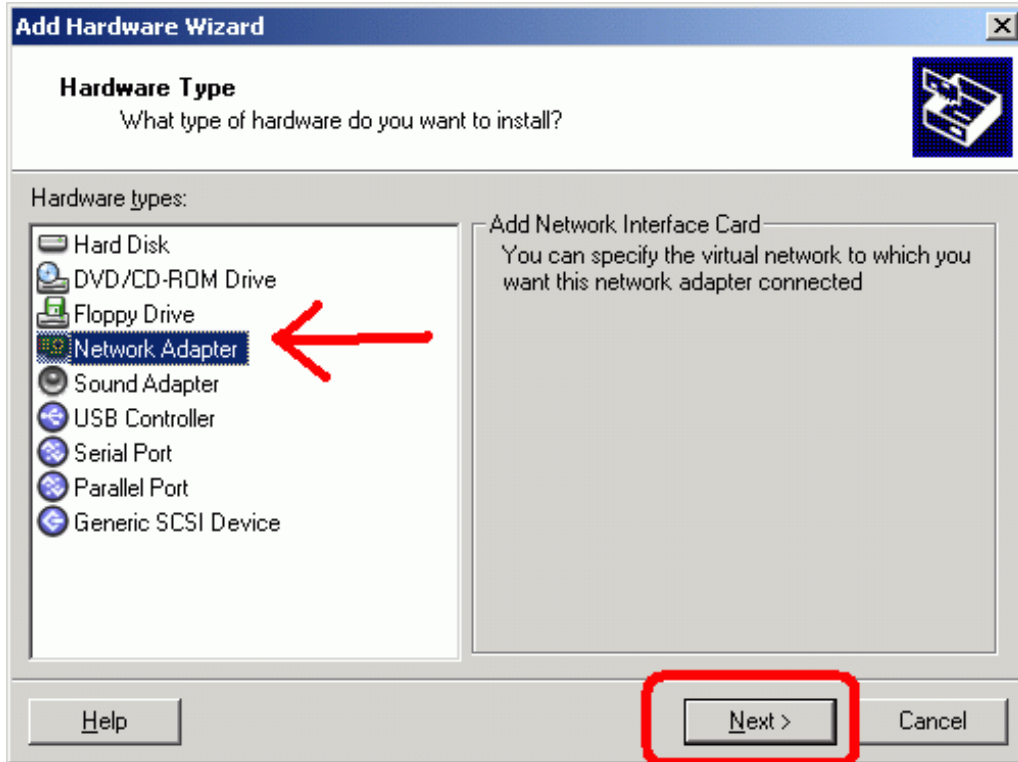
Wir gehen wieder in den Configuration Editor



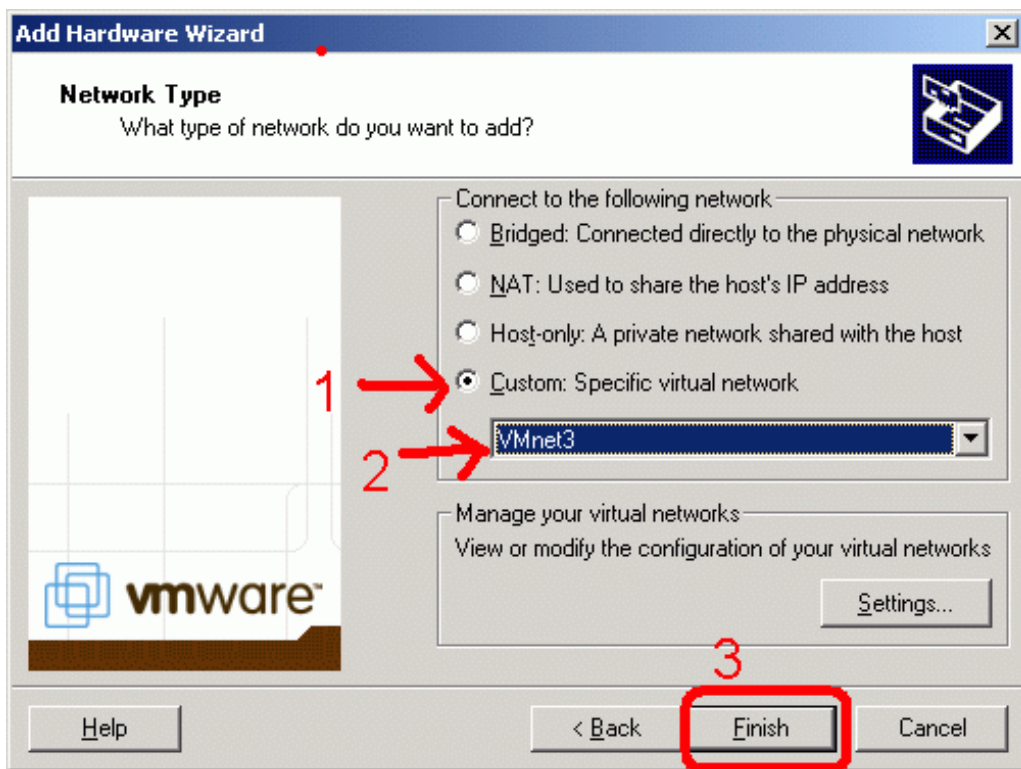
Wir fügen neue Hardware hinzu mit "Add"



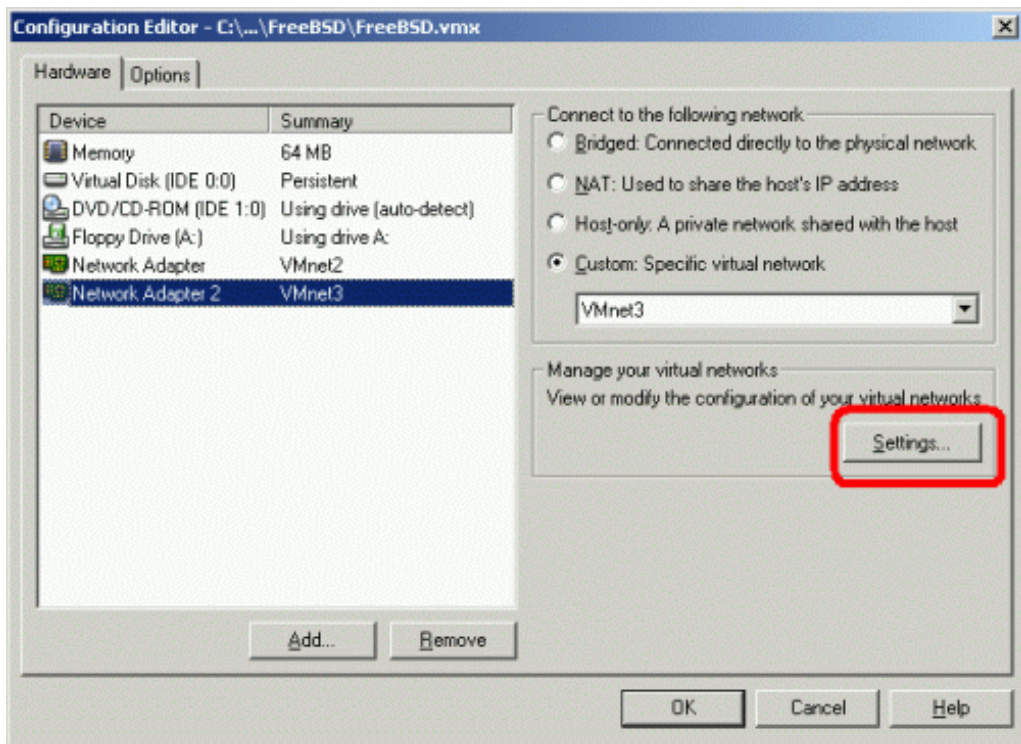
Hier wählen wir "Network Adapter" und machen weiter mit "Next"



Wir wählen wieder "Custom", dann VMnet3 und machen weiter mit "Finish"



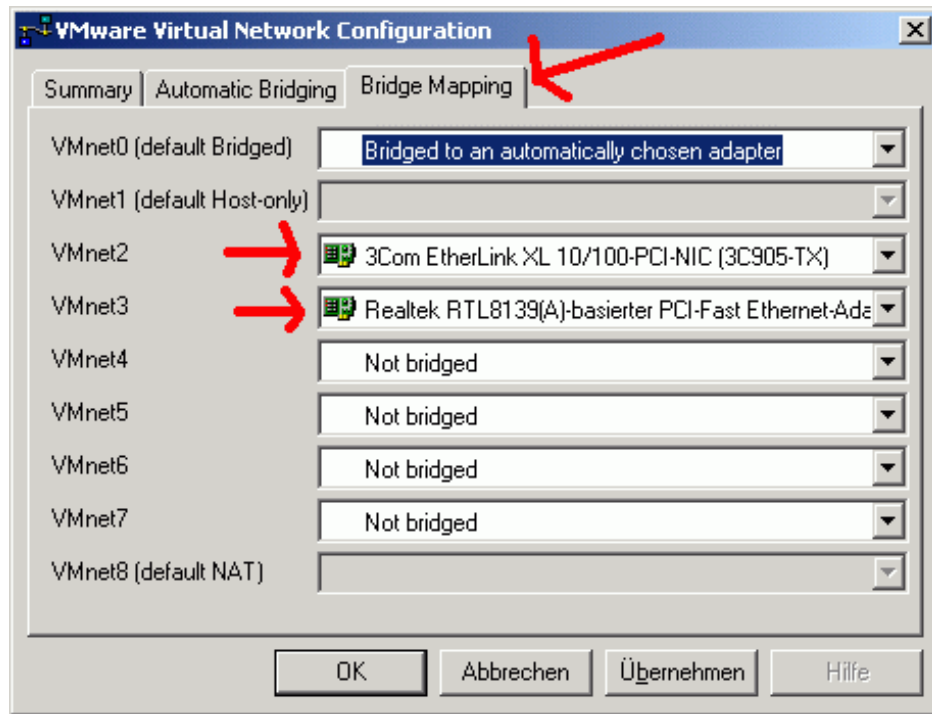
Wir sollten nun im Configuration Editor 2 Network Adapter sehen können. Weiter geht es mit Klick auf Settings



In der Virtual Network Configuration stellen wir nun auf dem Tab "Bridge Mapping" unsere beiden Netzwerkkarten für den FLI ein.

Ich habe meine Karte für das LAN auf VMnet2 gelegt.
Die Karte fürs DSL dann auf VMnet3

So sollte es auch in FLI erkannt werden. Die erste Karte (eth0) fürs LAN, die zweite (eth1) fürs DSL.



Konfiguration von FLI4L

Jetzt können wir die Konfiguration von FLI beginnen.
Viel ist nicht zu tun.

eth0 wird die interne Netzwerkkarte fürs LAN
eth1 wird die Netzwerkkarte für DSL

BASE.TXT

```
ETH_DRV_N='1'  
ETH_DRV_1='pcnet32'  
ETH_DRV_1_OPTION=""
```

DSL.TXT

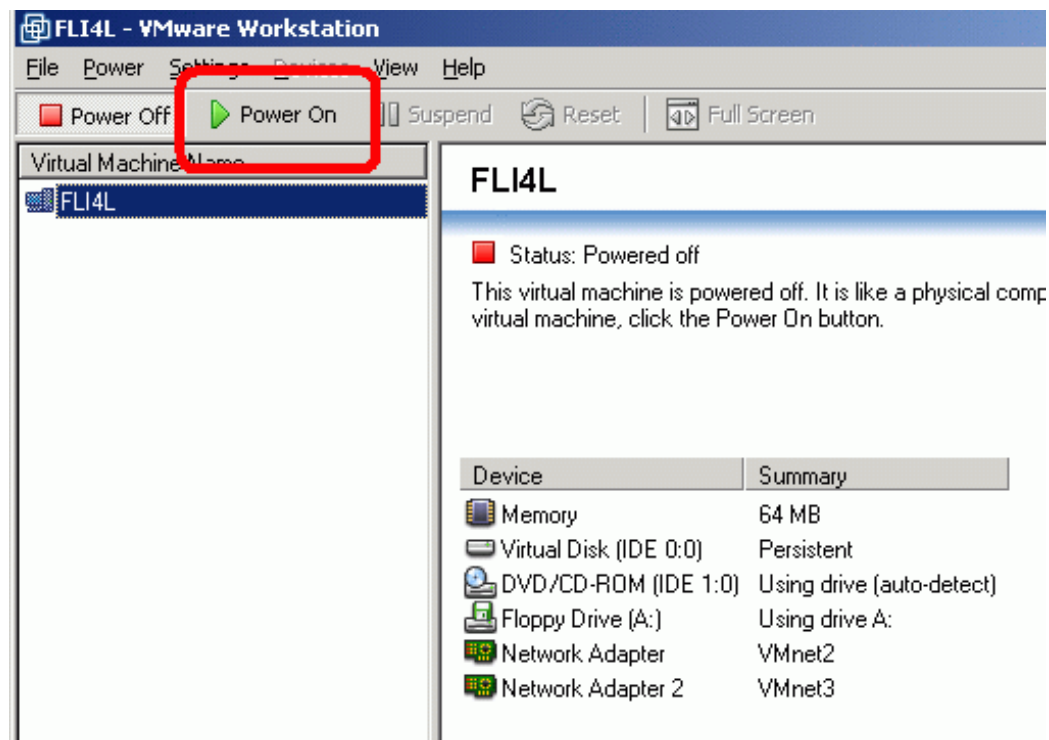
```
PPPOE_ETH='eth1'
```

Obwohl wir 2 Netzwerkkarten verschiedener Hersteller haben, stellen wir hier unabhängig davon in FLI die Netzwerkkarte **pcnet32** ein. Da unsere beiden Netzwerkkarten unter dem gleichen Treiber laufen, benötigen wir bei **ETH_DRV_N** auch nur 1. Egal welche Netzwerkkarte wir physikalisch im Rechner betreiben, wird dem FLI4L oder anderen Betriebssystemen von VMware immer eine pcnet32 vorgegault.

Nun noch das OPT HD und das OPT DSL hinzu, und die Diskette erstellen. Mehr OPTs sind zum jetzigen Zeitpunkt in den meisten Fällen unnötig. Näheres hierzu in den FLI4L ReadMe Dateien (u.a. HD Install Vorgang).

Start von VMware und FLI4L

Wir schliessen alle Netzkabel an die Karten an und starten in VMware unseren FLI durch Klick auf "Power On"



Alles weitere sollte wie in den jeweiligen HowTos für FLI4L beschrieben ablaufen.
Von nun an sollte FLI4L auf dem Client in einem Fenster lauffähig sein.

```
fli4l Version 2.0.4

Password:

Welcome to fli4l !

fli:/ # ping www.fli4l.de
PING antilope.in-berlin.de (192.109.42.66): 56 data bytes

64 bytes from 192.109.42.66: icmp_seq=0 ttl=247 time=21.2 ms
64 bytes from 192.109.42.66: icmp_seq=1 ttl=247 time=51.1 ms
64 bytes from 192.109.42.66: icmp_seq=2 ttl=247 time=52.3 ms
64 bytes from 192.109.42.66: icmp_seq=3 ttl=247 time=51.8 ms
64 bytes from 192.109.42.66: icmp_seq=4 ttl=247 time=52.7 ms

--- antilope.in-berlin.de ping statistics ---
5 packets transmitted, 5 packets received, 0% packet loss
round-trip min/avg/max = 21.2/45.8/52.7 ms
fli:/ # _
```

Die VMware Tools, von denen im unteren Rand des Fensters die Rede ist brauchen nicht installiert zu werden. Sie bringen nichts. Sie sind eigentlich hauptsächlich dazu da, eine bessere Leistung der Maus, eine höhere Bildschirm-Auflösung und verbesserte Disk-Performance im jeweiligen Betriebssystem zu ermöglichen.

Maus in FLI4L? :-) unnötig

Höhere Auflösung in FLI4L? :-) unnötig

Bessere Disk Performance? Eventuell, aber nicht unbedingt notwendig. Die Performance der Festplatte reicht schon in der Basiskonfiguration von VMware ohne Optimierungen dicke aus. So viele Daten werden nun ja auch nicht übertragen.

Das Kapitel "Nur eine Netzwerkkarte"

Oder: es geht alles viel, viel einfacher und rudimentärer

Man braucht keine Festplatteninstallation im FLI4L in VMware einzurichten. Selbstverständlich kann man auch wie gewohnt von Diskette booten, oder die Diskette in ein Image File schreiben und somit die Bootgeschwindigkeit deutlich erhöhen.

Helfen kann uns dabei u.a. das Tool WinImage, oder jedes andere Tool, das reine Images von Floppys erstellen kann. Dazu muss man wie gewohnt die FLI4L Diskette erstellen ("backen") und diese dann in dem entsprechenden Tool in ein Floppy-Image wandeln.

Unsere Konfiguration an der VMWare Einstellung ändert sich wie folgt:

Wir stellen die beiden Netzwerkkarten um auf den Modus Bridged, an stelle von der bisherigen Einstellung VMnet2 bzw. VMnet3.

Dazu klicken wir auf *Edit Virtual Machine Settings*

FLI4L

D:\VMware\FLI4L\FLI4L.vmx

Powered off

- ▶ Start this virtual machine
- 🔧 Edit virtual machine settings



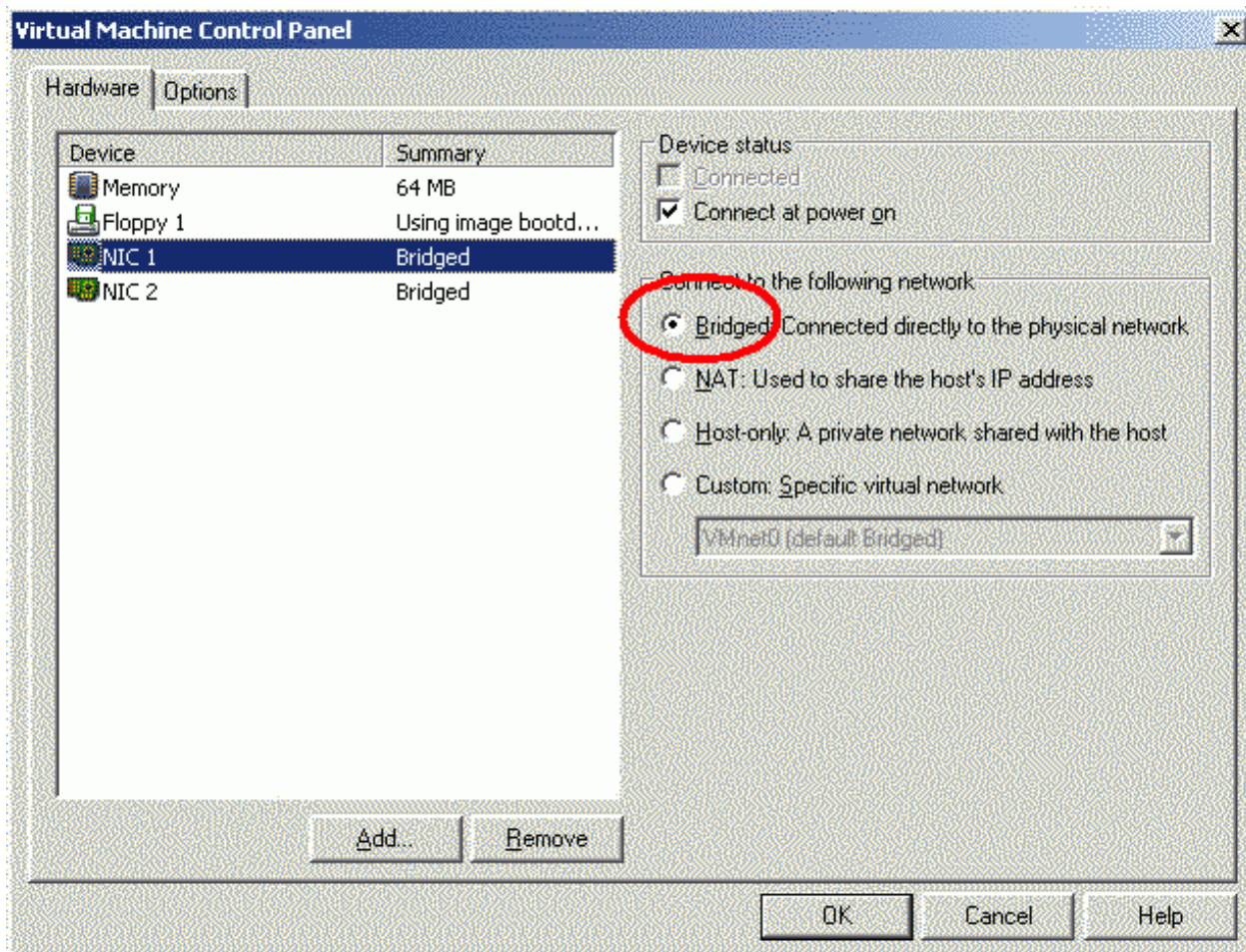
Notes

Type here to enter notes for this virtual machine

Summary

Memory	64 MB
Floppy 1	Using image bootdisk.flp
NIC 1	Bridged
NIC 2	Bridged

Dann wählen wir NIC1 aus und ändern sie in den Bridged Mode:



Mit NIC2 machen wir das gleiche.

An den Einstellungen in der base.txt sollte sich nichts ändern, von HD-Installation oder OPT BOOTCD mal abgesehen. Zudem könnten wir das CD-ROM und die Festplatte entfernen, wenn wir sie nicht mehr brauchen. Das habe ich auf dem Bild schon getan.

Jetzt müssen wir sicherstellen, dass der Host-PC keine Dienste des internen LAN mehr ausführt. Nun können wir das Netzwerkkabel vom DSL-Modem direkt in die Netzwerkbuchse der einzigen NIC des Host-PC stecken.

Mit einer einzigen Netzwerkkarte ist Routing nicht mehr so toll. Darum fällt das Routing ganzer Netze bei dieser Möglichkeit flach.

Wie schon anfangs gesagt, habe ich dies praktisch gefunden, als ich mal meinen Router bei einem Bekannten vergessen hatte und abends noch mal schnell ins Internet wollte, der (Windows) Rechner aber für die Einwahl des FLI4L konfiguriert war.

Eine Alternative, die das Routing ermöglicht wäre es, das DSL-Netzkabel an einen Switch zu hängen, den Host-PC mit dem virtuellen VMware und die PCs des Netzwerkes somit mit dem Signal zu versorgen.

Eine andere Lösung wäre es, eine weitere Netzwerkkarte in den Host-PC zu verbauen und an diese dann das weitere interne LAN anzustöpseln.

Dieses How-To ist momentan noch in der Erprobung. Ich konnte es bisher erst auf sehr wenigen Rechnern testen.

Für Verbesserungen, Optimierungen, Fehlermeldungen, Kommentare o.ä. bin ich zu erreichen unter [Email: hoescheler@t-online.de](mailto:hoescheler@t-online.de)

Ferner geht mein Dank auch an die vielen Hilfen der FLI4L Fangemeinde und des FLI4L Teams, ohne die ich niemals einen ausgereiften Softwarerouter in vollem Funktionsumfang hätte aufsetzen können.

bye

Gebhard Höscheler, Februar 2004

[Klicken Sie hier, um die Seite auszudrucken.](#)