

# Performance Tuning

## Tuning guide - setup the host for expected usage ...

Workstation and VMplayer can be tuned in different directions:

- best performance of a few VMs
- low impact on the host
- run many VMs and over-commit RAM
- factory defaults

The factory defaults have changed over the years and vary on Windows and Linux and 32 and 64 bit systems.

So before you complain about poor performance keep in mind that the factory defaults try to find a healthy compromise between the extremes.

If you are not happy with the defaults - maybe because you can not run as many VMs as you wanted to - or maybe because your most important VM is too slow - or .... you can change the factory defaults. This is a guide how to do this ...

---

The perceived performance of a VM depends on 3 big factors:

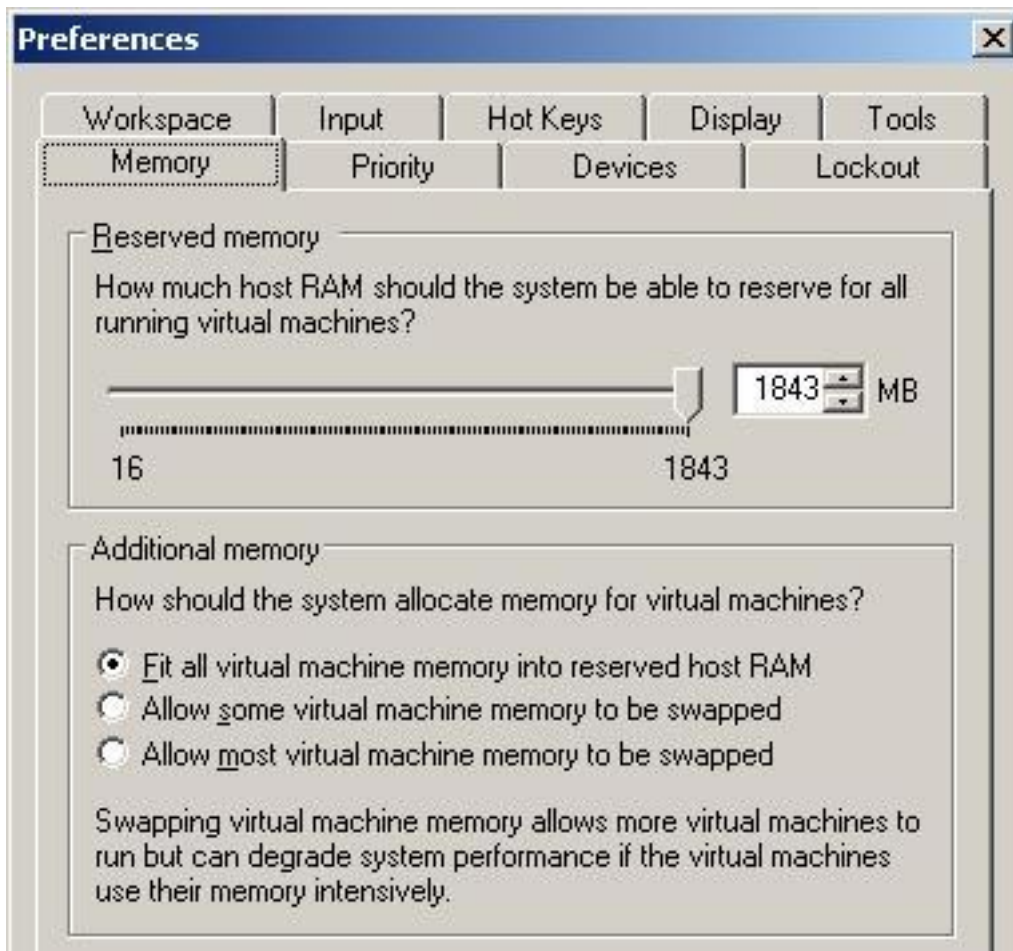
- write performance for vmdk-files
- amount of virtual CPUs
- behaviour and amount of the virtual RAM

The first 2 factors will vary from VM to VM so we can ignore them for now.

As different users may have different needs Workstation allows to set something like a hostwide "RAM-handling-policy"

This policy is defined in the file config.ini and can be set via the UI-option menu > edit > preferences > memory

# Performance Tuning



## How much RAM should be available to VMs ? - Reserved Memory

If you manually set the slider this will result in an entry like this in the config.ini file  
`prefvmx.allVMMemoryLimit = "1843"`

Do not set this manually unless the amount of memory available is definitely wrong. Usually VMware sets this value automatically and you do not have to set this.

If you have to set this manually use settings like 90% of the available RAM for normal operation.

If you want to cripple Workstation to make sure other, more important applications are not affected

use lower values.

The next parameter in the config.ini file is related

`prefvmx.useRecommendedLockedMemSize = "TRUE"`

is used for best performance of a few VMs

`prefvmx.useRecommendedLockedMemSize = "FALSE"`

is useful if you want to run many VMs with the price of a reduced performance.

---

## How should the system allocate memory for virtual machines ?

If you mark the checkbox "Fit all ..." you get this line in the config.ini:

`prefvmx.minVmMemPct = "100"`

# Performance Tuning

If you mark the checkbox "Allow some ..." you get this line in the config.ini:

```
prefvmx.minVmMemPct = "50"
```

If you mark the checkbox "Allow most ..." you get this line in the config.ini:

```
prefvmx.minVmMemPct = "25"
```

If want to heavily overcommit RAM - for example run a VM with 8 GB RAM while your host has only 4 GB of RAM

you can use even lower values like

```
prefvmx.minVmMemPct = "5"
```

The performance differences using this different settings are significant !

Use 100 to run a few VMs with best performance.

Use lower values if you want to run many VMs at the same time.

The factory defaults for `prefvmx.minVmMemPct` have changed over the years - so if you want to make sure you get the expected behaviour set this parameter yourself.

---

## Where should the system store the file used to backup virtual RAM ?

(This can be set in the config.ini manually but it is NOT available via the UI.)

While a VM is running VMware needs to create a file to backup this memory.

This file will be created when you start the VM and it will be destroyed if you power off the VM.

If you suspend a VM this file will be renamed and it will be stored in the directory of the VM.

The extension of this file is \*.vmem

The location where this file is stored has a big influence on the performance of the VM.

The fastest location for this file is inside the real RAM.

If this file is stored on a slow USB-stick obviously performance suffers a lot.

The location to this \*.vmem file can be changed - either per VM or host-wide - so that it affects all VMs you have.

So if you want to set the path for a single VM use the vmx-file.

If you want to set the behaviour for all VMs use the config.ini

Setting this in the vmx-file overrides settings in the config.ini

**WARNING: the syntax for this parameter has changed with the introduction of WS 7 so carefully read the next chapters.**

**Don't be surprised if old tweaks no longer have the expected effects !**

---

## Windows - 32 bit hosts

On Windows 32 bit hosts the path to the \*.vmem file is set by the parameter

```
mainmem.useNamedFile
```

# Performance Tuning

**mainmem.useNamedFile = "TRUE"**

creates a \*.vmem file with a random name inside the directory of the vmx-file.

This file will be destroyed when the VM is powered off or it will be renamed when the VM is suspended.

**mainmem.useNamedFile = "FALSE"**

creates a \*.vmem file and allocates it in real RAM plus pagefile.sys - if the host uses a pagefile

This file will be destroyed when the VM is powered off or it will be stored as \*.vmem file inside the directory of the vmx-file when the VM is suspended.

Obviously the best performance for a single VM will be achieved by storing the \*.vmem file in real RAM but due to rests of old code inside WS and Player the max amount of virtual RAM that can really be allocated in real RAM is 1600 Mbs.

So a VM configured with lets say 3 dims a 512 MB = 1536 MB should have the most responsive RAM possible with WS. Taskmanager should show about 1600 mbs of RAM-usage for the process vmware-vmx.exe.

If you assign more then 1600 the RAM-usage drops down to lower values

To run many VMs use "TRUE" - which is set by default

---

## Windows - 64 bit hosts

On Windows 64 bit hosts the path to the \*.vmem file is set by the parameter

**mainmem.useNamedFile**

**mainmem.useNamedFile = "TRUE"**

creates a \*.vmem file with a random name inside the directory of the vmx-file.

This file will be destroyed when the VM is powered off or it will be renamed when the VM is suspended.

**mainmem.useNamedFile = "FALSE"**

creates a \*.vmem file and allocates it in real RAM plus pagefile.sys - if the host uses a pagefile

This file will be destroyed when the VM is powered off or it will be stored as \*.vmem file inside the directory of the vmx-file when the VM is suspended.

Obviously the best performance for a single VM will be achieved by storing the \*.vmem file in real RAM.

To run many VMs use "TRUE" - which is set by default

---

## Workstation 5.0 to 6.5 on Linux

On Linux hosts using Workstation 5.0 to 6.5 the path to the \*.vmem file is set by the parameter

**mainmem.useNamedFile**

**mainmem.useNamedFile = "TRUE"**

# Performance Tuning

creates a \*.vmem file with a random name inside the directory of the vmx-file.  
This file will be destroyed when the VM is powered off or it will be renamed when the VM is suspended.

**mainmem.useNamedFile = "FALSE"**

creates a \*.vmem file in /tmp

This file will be destroyed when the VM is powered off or it will be stored as \*.vmem file inside the directory of the vmx-file when the VM is suspended.

Giving a suggestion here is not possible - decide yourself

- do I have enough space in /tmp ?
- which disk is faster ...

To run many VMs use "TRUE" - which is set by default

---

## Workstation 7 on Linux - 32 bit hosts

On Linux 32 bit hosts the path to the \*.vmem file is set by the parameter **mainmem.backing**  
The parameter **mainmem.useNamedFile** will be ignored.

**mainmem.backing = "named"**

creates a \*.vmem file with a random name inside the directory of the vmx-file.

This file will be destroyed when the VM is powered off or it will be renamed when the VM is suspended.

**mainmem.backing = "unnamed"**

creates a \*.vmem file in /tmp

This file will be destroyed when the VM is powered off or it will be stored as \*.vmem file inside the directory of the vmx-file when the VM is suspended.

Giving a suggestion here is not possible - decide yourself

- do I have enough space in /tmp ?
- which disk is faster ...

To run many VMs use "named"

---

## Workstation 7 on Linux - 64 bit hosts

On Linux 64 bit hosts the path to the \*.vmem file is set by the parameter **mainmem.backing**  
The parameter **mainmem.useNamedFile** will be ignored.

**mainmem.backing = "named"**

creates a \*.vmem file with a random name inside the directory of the vmx-file.

# Performance Tuning

This file will be destroyed when the VM is powered off or it will be renamed when the VM is suspended.

`mainmem.backing = "unnamed"`

creates a \*.vmem file in /tmp

This file will be destroyed when the VM is powered off or it will be stored as \*.vmem file inside the directory of the vmx-file when the VM is suspended.

`mainmem.backing = "swap"`

creates a \*.vmem file and allocates it in real RAM plus swap-area - if the host uses a swap-area

This file will be destroyed when the VM is powered off or it will be stored as \*.vmem file inside the directory of the vmx-file when the VM is suspended.

Obviously the best performance for a single VM will be achieved by storing the \*.vmem file in real RAM.

To run many VMs use "named"

---

## Setting a policy for all VMs ...

There are a few more parameters that can be used to fine-tune the performance of a single VMs but we have covered the basics now.

So if you plan to

- run a few VMs with best possible performance on a Windows host use

`prefvmx.minVmMemPct = "100"`

`prefvmx.useRecommendedLockedMemSize = "TRUE"`

`mainmem.useNamedFile= "FALSE"`

- run very many VMs with reduced performance on a Windows host use

`prefvmx.minVmMemPct = "25"`

`prefvmx.useRecommendedLockedMemSize = "FALSE"`

`mainmem.useNamedFile= "TRUE"`

- to restrict the amount of RAM available to VMs globally use

`prefvmx.allVMMemoryLimit = "1200"`

`prefvmx.useRecommendedLockedMemSize = "TRUE"`

- run a few VMs with best possible performance on a Linux 64 bit host use (WS 7 only)

`prefvmx.minVmMemPct = "100"`

`prefvmx.useRecommendedLockedMemSize = "TRUE"`

`mainmem.backing= "swap"`

---

Unique solution ID: #1075

Author: Ulli Hankeln

# Performance Tuning

Last update: 2010-10-29 05:40