
Parallels Server 4.0 for Mac Bare Metal Edition

User's Guide



*Parallels Holdings, Ltd.
c/o Parallels International GmbH.
Parallels International GmbH
Vordergasse 49
CH8200 Schaffhausen
Switzerland
Tel: + 49 (6151) 42996 - 0
Fax: + 49 (6151) 42996 - 255*

Copyright © 1999-2010 Parallels Holdings, Ltd. and its affiliates. All rights reserved.

Parallels, Coherence, Parallels Transporter, Parallels Compressor, Parallels Desktop, and Parallels Explorer are registered trademarks of Parallels Software International, Inc. Virtuozzo, Plesk, HSPcomplete, and corresponding logos are trademarks of Parallels Holdings, Ltd. The Parallels logo is a trademark of Parallels Holdings, Ltd.

This product is based on a technology that is the subject matter of a number of patent pending applications. Virtuozzo is a patented virtualization technology protected by U.S. patents 7,099,948; 7,076,633; 6,961,868 and having patents pending in the U.S.

Plesk and HSPcomplete are patented hosting technologies protected by U.S. patents 7,099,948; 7,076,633 and having patents pending in the U.S.

Distribution of this work or derivative of this work in any form is prohibited unless prior written permission is obtained from the copyright holder.

Apple, Bonjour, Finder, Mac, Macintosh, and Mac OS are trademarks of Apple Inc.

Microsoft, Windows, Microsoft Windows, MS-DOS, Windows NT, Windows 95, Windows 98, Windows 2000, Windows XP, Windows Server 2003, Windows Vista, Windows Server 2008, Windows 7, Microsoft SQL Server, Microsoft Desktop Engine (MSDE), and Microsoft Management Console are trademarks or registered trademarks of Microsoft Corporation.

Linux is a registered trademark of Linus Torvalds.

Red Hat is a registered trademark of Red Hat Software, Inc.

SUSE is a registered trademark of Novell, Inc.

Solaris is a registered trademark of Sun Microsystems, Inc.

X Window System is a registered trademark of X Consortium, Inc.

UNIX is a registered trademark of The Open Group.

IBM DB2 is a registered trademark of International Business Machines Corp.

SSH and Secure Shell are trademarks of SSH Communications Security, Inc.

MegaRAID is a registered trademark of American Megatrends, Inc.

PowerEdge is a trademark of Dell Computer Corporation.

eComStation is a trademark of Serenity Systems International.

FreeBSD is a registered trademark of the FreeBSD Foundation.

Intel, Pentium, Celeron, and Intel Core are trademarks or registered trademarks of Intel Corporation.

OS/2 Warp is a registered trademark of International Business Machines Corporation.

VMware is a registered trademark of VMware, Inc.

All other marks and names mentioned herein may be trademarks of their respective owners.

Contents

Introduction	5
About This Guide.....	5
Organization of This Guide	6
Documentation Conventions.....	6
Getting Help.....	8
Feedback	8
Parallels Server 4.0 for Mac Bare Metal Edition Basics	9
Basics of Hardware Virtualization	9
Parallels Virtual Machines	10
Virtual Machine Hardware.....	11
Virtual Machine Files.....	12
Support of Virtual and Real Media	13
Parallels Management Console	14
Resource Management	15
Understanding Licensing	15
Physical Server Availability Considerations	16
Operations on Virtual Machines	17
Creating a Virtual Machine	18
Supported Guest Operating Systems.....	19
Performing Initial Configuration.....	19
Configuring Network Settings	20
Setting the Password for a Virtual Machine	20
Setting Startup Parameters	20
Installing Parallels Tools.....	21
Starting, Stopping, and Querying Status of a Virtual Machine	22
Listing Virtual Machines.....	23
Storing Extended Information on a Virtual Machine	24
Copying a Virtual Machine Within the Server.....	25
Suspending a Virtual Machine	26
Pausing a Virtual Machine	26
Running Commands in a Virtual Machine.....	27
Deleting a Virtual Machine	27
Managing Virtual Machine Backups.....	28
Backups Overview	28
Using pct backup and pct restore.....	30
Using pbackup and prestore	33
Migrating Virtual Machines.....	36
General Migration Requirements.....	37
Migrating Virtual Machines Between Parallels Servers	38
Migrating a Physical Computer to a Virtual Machine	39
Managing Templates	41
Create a Template	41
Listing Templates	42
Deploying a Template.....	42
Managing Virtual Machine Disks	43
Changing the Disk Type	43
Increasing the Virtual Disk Capacity	44

Reducing the Virtual Disk Capacity	45
Compacting the Virtual Disk	46
Managing Virtual Machine Devices.....	46
Adding a New Device.....	47
Configuring Virtual Devices.....	51
Deleting a Device.....	52
Managing Snapshots	53
Creating a Snapshot	54
Listing Snapshots.....	55
Reverting to a Snapshot	56
Deleting a Snapshot	56
Making Screenshots	57
Managing Virtual Machine Resources	58

Managing Parallels Server for Mac Bare Metal Edition Network 60

Managing Network Adapters on the Parallels Server	60
Listing Adapters.....	61
Creating VLAN Adapter.....	62
Connecting an Adapter to a Virtual Network	63
Managing Virtual Networks.....	64
Creating a Virtual Network.....	65
Configuring Virtual Network Parameters	66
Listing Virtual Networks	67
Deleting a Virtual Network.....	68
Managing Adapters in Virtual Machines	68
Creating and Deleting Virtual Adapters.....	69
Configuring Virtual Adapter Parameters	70
Connecting Virtual Machines to Virtual Networks.....	71

Managing Licenses 72

Installing the License	73
Updating the Current License	74
Transferring the License to Another Server	75
Viewing the Current License.....	76
Viewing the License	77
License Statuses	78

Keeping Your System Up To Date 79

Updating Parallels Server for Mac Bare Metal Edition Software	80
Updating Parallels Server for Mac Bare Metal Edition Software	81
Updating in Graphical Mode	82
Updating in Command Line Mode	84
Updating Software In Virtual Machines	84

Glossary 85

Index 86

CHAPTER 1

Introduction

Parallels Server 4.0 for Mac Bare Metal Edition is a virtualization solution that allows you to create and run Parallels virtual machines on bare Apple hardware.

In This Chapter

About This Guide.....	5
Getting Help.....	8
Feedback	8

About This Guide

The *Parallels Server 4.0 for Mac Bare Metal Edition User's Guide* provides comprehensive information on Parallels Server 4.0 for Mac Bare Metal Edition - high-end virtualization software for bare metal servers. It covers the necessary theoretical conceptions as well as practical aspects of working with Parallels Server for Mac Bare Metal Edition. The guide will familiarize you with the way to create and administer virtual machines using the Parallels command line interface.

Note: The guide does not explain how to install and configure your Parallels Server 4.0 for Mac Bare Metal Edition system. For detailed information on these operations, see the *Parallels Server 4.0 for Mac Bare Metal Edition Installation Guide*.

The primary audience for this guide is anyone responsible for administering one or more systems running Parallels Server 4.0 for Mac Bare Metal Edition. We assume that you have some familiarity with how to work in the Linux command line.

Organization of This Guide

This guide is organized in the following way:

- **Chapter 1, Introduction**, gives an overview of the Parallels Server for Mac Bare Metal Edition product and this guide.
- **Chapter 2, Parallels Server for Mac Bare Metal Edition Basics**, explains the general principles of Parallels Server for Mac Bare Metal Edition operation.
- **Chapter 3, Operations on Virtual Machines**, covers those operations that you can perform on a virtual machine: creating and deleting virtual machines, starting and stopping them, backing up and restoring, etc. You will also learn how to migrate virtual machines between Parallels servers and migrate a physical server to a virtual machine.
- **Chapter 4, Managing Parallels Server for Mac Bare Metal Edition Network**, familiarizes you with the Parallels Server for Mac Bare Metal Edition network structure and explains how to manage networks in Parallels Server for Mac Bare Metal Edition systems.
- **Chapter 5, Managing Licenses**, provides detailed information on managing licenses in Parallels Server for Mac Bare Metal Edition.
- **Chapter 6 Keeping Your System Up To Date**, informs you of the ways to keep all the software components of a Parallels server up to date.

Documentation Conventions

Before you start using this guide, it is important to understand the documentation conventions used in it.

The table below presents the existing formatting conventions.

<u>Formatting convention</u>	<u>Type of Information</u>	<u>Example</u>
Special Bold	Items you must select, such as menu options, command buttons, or items in a list.	Go to the Resources tab.
	Titles of chapters, sections, and subsections.	Read the Basic Administration chapter.

<i>Italics</i>	Used to emphasize the importance of a point, to introduce a term or to designate a command-line placeholder, which is to be replaced with a real name or value.	These are the so-called <i>EZ templates</i> . To destroy a Container, type <code>vmctl destroy <i>ctid</i></code> .
Monospace	The names of commands, files, and directories.	Use <code>vmctl start</code> to start a Container.
Preformatted	On-screen computer output in your command-line sessions; source code in XML, C++, or other programming languages.	<pre>Saved parameters for Container 101</pre>
Monospace Bold	What you type, as contrasted with on-screen computer output.	<pre># rpm -V virtuo-release</pre>
Key+Key	Key combinations for which the user must press and hold down one key and then press another.	Ctrl+P, Alt+F4

Besides the formatting conventions, you should also know about the document organization convention applied to Parallels documents: chapters in all guides are divided into sections, which, in their turn, are subdivided into subsections. For example, **About This Guide** is a section, and **Documentation Conventions** is a subsection.

Getting Help

In addition to this guide, there are a number of other resources available for Parallels Server for Mac Bare Metal Edition which can help you use the product more effectively. These resources include:

Manuals:

- *Parallels Server 4.0 for Mac Bare Metal Edition Installation Guide*. This guide provides detailed information on installing Parallels Server for Mac Bare Metal Edition on your server, including the pre-requisites and the stages you shall pass.
- *Getting Started With Parallels Server 4.0 for Mac Bare Metal Edition*. This guide provides basic information on how to install Parallels Server for Mac Bare Metal Edition on your server, create new virtual machines, and perform main operations on them. Unlike the *Parallels Server 4.0 for Mac Bare Metal Edition Installation Guide*, it does not contain detailed description of all the operations needed to install and set Parallels Server for Mac Bare Metal Edition to work (e.g. installing Parallels Server for Mac Bare Metal Edition in text mode).
- *Parallels Command Line Reference Guide*. This guide is a complete reference on all Parallels Server for Mac Bare Metal Edition configuration files and command line utilities.

Help systems:

- *Getting Started with Parallels Management Console*. This help system provides information on how to start working in Parallels Management Console. You will learn how to install this application on your computer, connect to a server running Parallels Server for Mac Bare Metal Edition, and perform the basic operations on your virtual machines.
- *Parallels Management Console User's Guide*. This help system provides detailed information on Parallels Management Console - a graphical user interface tool for managing physical servers and their virtual machines.

Feedback

If you spot a typo in this guide, or if you have thought of a way to make this guide better, you can share your comments and suggestions with us by completing the feedback form at the Parallels documentation feedback page (<http://www.parallels.com/en/support/usersdoc/>).

CHAPTER 2

Parallels Server 4.0 for Mac Bare Metal Edition Basics

This chapter provides a brief description of Parallels Server 4.0 for Mac Bare Metal Edition, Parallels virtual machines, their specifications and underlying technologies.

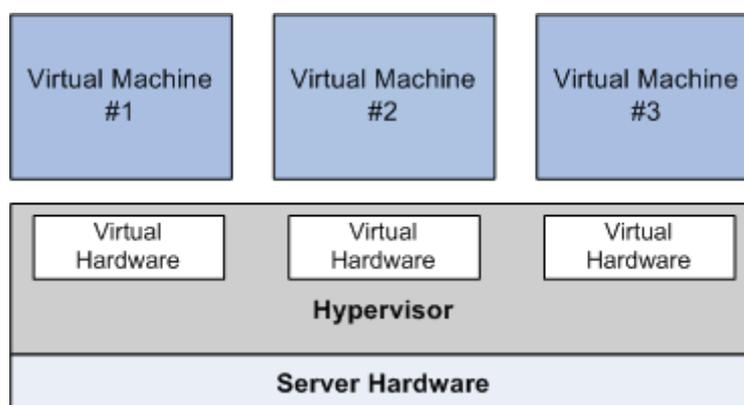
In This Chapter

Basics of Hardware Virtualization	9
Parallels Virtual Machines	10
Virtual Machine Hardware.....	11
Virtual Machine Files.....	12
Support of Virtual and Real Media	13
Parallels Management Console	14
Resource Management	15
Understanding Licensing	15
Physical Server Availability Considerations.....	16

Basics of Hardware Virtualization

The basis of Parallels Server for Mac Bare Metal Edition is the hardware virtualization layer (or *hypervisor*). This layer is loaded directly on the bare server and acts as an intermediary between the server hardware and virtual machines. To allocate hardware and resources to virtual machines, Parallels Server for Mac Bare Metal Edition virtualizes all hardware on the server. Once virtualized, hardware and resources can be easily assigned to virtual machines. Based on the virtual hardware, a virtual machine runs its own complete copies of an operating system and applications.

The following diagram shows the basic architecture of hardware virtualization.



Hardware virtualization provides many benefits the main of which are listed below:

- Create multiple virtual machines with different operating systems on a single physical computer.
- Manage several physical servers at a time using Parallels Management Console, an integrated GUI-based multi-server and cross-platform management tool.
- Run several guest operating systems and their applications simultaneously on a single physical computer without rebooting.
- Consolidate and virtualize the computing environment, reduce hardware costs, lower operating expenses, and increase productivity.
- Use open APIs and SDK to extend management integration with in-house and third-party applications.

Parallels Virtual Machines

From the point of view of applications and virtual machine users, each virtual machine is an independent system with an independent set of virtual hardware. This independence is provided by the Parallels Server for Mac Bare Metal Edition hardware virtualization layer. The main features of the virtualization layer are the following:

- A virtual machine looks like a normal computer. It has its own virtual hardware, and software applications can run in virtual machines without any modifications or adjustment.
- A user can easily change the virtual machine configuration (e.g. add a new virtual disk or increase memory).
- Virtual machines are fully isolated from each other (file system, processes, `sysctl` variables) and Parallels Server for Mac Bare Metal Edition.
- Install any of the supported operating systems in the virtual machine. The guest operating system and its applications are isolated inside a virtual machine and share physical hardware resources with other virtual machines.

Intel and AMD Virtualization Technology Support

Parallels Server for Mac Bare Metal Edition provides support for Intel and AMD virtualization technologies comprising a set of processor enhancements and improving the work of virtualization solutions. Utilizing these technologies, Parallels Server for Mac Bare Metal Edition can offload some workload to the system hardware, which results in the "near native" performance of guest operating systems.

Virtual Machine Hardware

A Parallels virtual machine works like a stand-alone computer with the following hardware:

CPU	Up to 12-core Intel/AMD CPU (Intel Celeron or AMD Duron for legacy OS compatibility)
Motherboard	Intel i965 chipset-based motherboard
RAM	Up to 64 GB of main memory
Video Adapter	VGA and SVGA with VESA 3.0 compatible video adapter
Video RAM	Up to 256 MB of video memory
Floppy Disk Drive	1.44 MB floppy disk drive mapped to an image file or to a physical floppy drive
IDE Devices <ul style="list-style-type: none"> ▪ Hard Disk ▪ CD/DVD-ROM Drive 	Up to 4 IDE devices Hard disk drive mapped to an image file (up to 2 TB each) CD/DVD-ROM drive mapped to a physical drive or to an image file
SCSI Devices <ul style="list-style-type: none"> ▪ Hard Disk ▪ Generic SCSI Device 	Up to 15 SCSI devices Hard disk drive mapped to an image file (up to 2 TB each) Generic SCSI device
Network Interfaces	Up to 16 network interfaces, including Ethernet virtual network cards compatible with RTL8029
Serial (COM) Ports	Up to 4 serial (COM) ports mapped to a socket or to an output file
Parallel (LPT) Ports	Up to 3 parallel (LPT) ports mapped to an output file, to a real port, or to a printer
Sound Card	AC'97-compatible sound card, sound recording support
Keyboard	Generic PC keyboard
Mouse	PS/2 wheel mouse

Virtual Machine Files

A virtual machine has at least two files: a configuration file (PVS file) and a hard disk image file (HDD file). It can also have additional files: a file for each additional virtual hard disk and output files for virtual ports. By default, the virtual machines files are stored in the `/var/parallels` directory on the Parallels server.

The list of files related to a virtual machine is given in the table below:

File Name	Description
<code>.pvm</code>	A bundle that contains the virtual machine files.
<code>.pvs</code>	A virtual machine configuration file. It defines the hardware and resources configuration of the virtual machine. The configuration file is automatically generated during the virtual machine creation.
<code>.sav</code>	A dump file created when you suspend the virtual machine. This file contains the state of the virtual machine and its applications at the moment the suspend was invoked.
<code>.mem</code>	A file containing the memory dump for the suspended virtual machine. For a running virtual machine, it is a temporary virtual memory file.
<code>.hdd</code>	A file representing a virtual hard disk. When you create a virtual machine, you can create it with a new virtual hard disk or use an existing one. A virtual machine can have several hard disks.
<code>.iso</code>	An image file of a CD or DVD disc. Virtual machines treat ISO images as real CD/DVD discs.
<code>.txt</code>	Output files for serial and parallel ports. The output <code>.txt</code> files are generated when a serial or parallel port connected to an output file is added to the virtual machine configuration.

Support of Virtual and Real Media

This section lists the types of disks that can be used by Parallels virtual machines and provides the information about basic operations you can perform on these disks.

Supported Types of Hard Disks

Parallels virtual machines can use only virtual hard disks image files as their hard disks.

Virtual Hard Disks

The capacity of a virtual hard disk can be set from 100 MB to 2 TB.

Virtual hard disks can be of either *plain* or *expanding* format. When you create a virtual machine in **Express Windows** or **Typical** mode (in the New Virtual Machine wizard), the disk is created in the *expanding* format.

- | | |
|------------------|---|
| plain | A plain virtual hard disk image file has a fixed size. The size is determined when the disk is created. Plain disks can be created with the help of New Virtual Machine wizard (the Custom mode.) |
| expanding | An expanding virtual hard disk image file is small initially. Its size grows as you add applications and data to the virtual hard disk in the guest OS. |

Split disks

A virtual disk of either format can be a single-piece disk or a split disk. A split disk is cut into 2 GB pieces and is stored as a single .hdd file.

CD/DVD Discs and Their Images

Parallels Server for Mac Bare Metal Edition can access real CD/DVD discs and images of CD/DVD discs.

Parallels Server for Mac Bare Metal Edition has no limitations on using multi-session CD/DVD discs. A virtual machine can play back audio CDs without any limitations on copy-protected discs.

If your server has a recordable optical drive, you can use it to burn CD or DVD discs in a virtual machine.

Parallels Server for Mac Bare Metal Edition supports CD/DVD disc images in ISO, CUE, and CCD formats.

Floppy Disks and Floppy Disk Images

Parallels Server for Mac Bare Metal Edition can use two types of floppy disks:

- Real diskettes inserted into a real floppy disk drive that is connected to the virtual machine.
- Floppy disk image files having the .fdd extension and connected to the virtual machine.

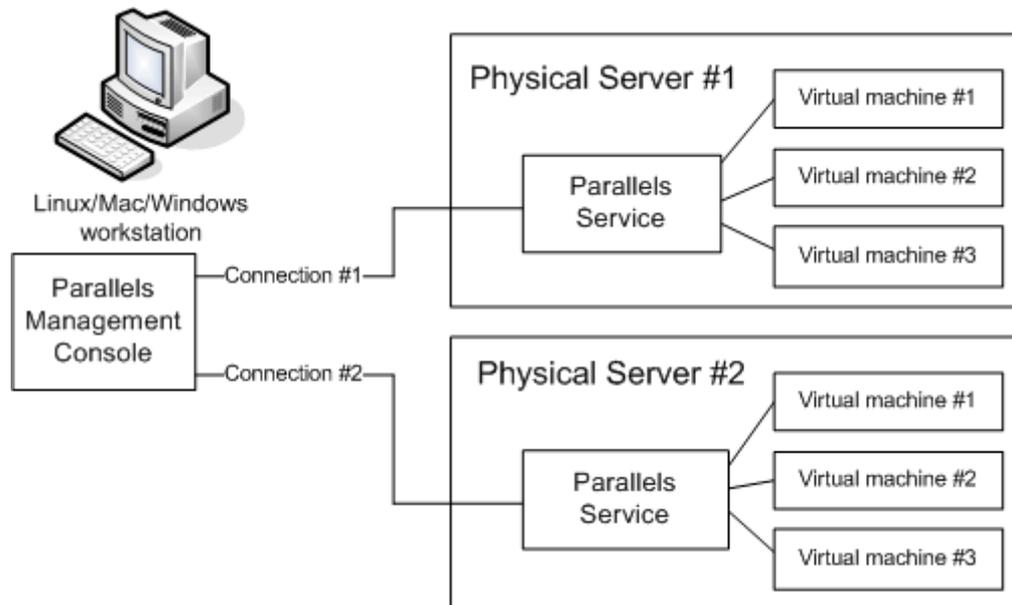
Parallels Server for Mac Bare Metal Edition treats floppy disk images like real diskettes. Parallels Server for Mac Bare Metal Edition supports floppy disk image files that have the `.fdd` extension and are 1.44 MB in size.

With Parallels Server for Mac Bare Metal Edition, you can also create an image of a blank floppy using the Floppy Disk pane of the **Virtual Machine Configuration** dialog.

Note: Parallels Server for Mac Bare Metal Edition cannot create images of real diskettes.

Parallels Management Console

Parallels Management Console is a remote tool with a graphical user interface (GUI) for managing your physical servers with Parallels Server for Mac Bare Metal Edition and virtual machines residing on them. This tool uses a typical client-server architecture.



The client application with the graphical user interface is installed on a computer running one of the supported Linux, Mac, or Windows operating systems. Once the client application is up and running, it can connect to the Parallels Server for Mac Bare Metal Edition software on a physical server. The client application can control multiple physical servers simultaneously (e.g. *Physical Server #1* and *Physical Server #2* as shown in the picture above). After the connection to the required physical server has been established, you can start managing this server and its virtual machines using the intuitive and comfortable GUI.

Resource Management

Parallels Server for Mac Bare Metal Edition resource management controls the amount of resources available to virtual machines. The controlled resources include such parameters as CPU power, disk space, a set of memory-related parameters. Resource management allows you to:

- effectively share available physical server resources among virtual machines
- guarantee Quality-of-Service in accordance with a service level agreement (SLA)
- provide performance and resource isolation and protect from denial-of-service attacks
- simultaneously assign and control resources for a number of virtual machines
- collect usage information for system health monitoring

Resource management is much more important for Parallels Server for Mac Bare Metal Edition than for a standalone server since server resource utilization in such a system is considerably higher than that in a typical system.

Understanding Licensing

To start using the Parallels Server for Mac Bare Metal Edition software, you need a special license - *Parallels Server for Mac Bare Metal Edition license*. You must install this license on your server after or when installing Parallels Server for Mac Bare Metal Edition on it. Every physical server hosting virtual machines must have its own license. Licenses are issued by Parallels and define a number of parameters in respect of your physical server. The main licensed parameters are listed below:

- The number of CPUs which can be installed on the physical server. Keep in mind that each of the Dual Core and Hyperthreading processors is regarded as one CPU.
- The license expiration date. Any license can be time-limited or permanent.

Parallels Server for Mac Bare Metal Edition licenses have a start date, and if they are time-limited, can also have an expiration date specified in them. You must set up your system clock correctly; otherwise, the license validation may fail.

- The number of virtual machines the physical server will be able to host.
- The platform and architecture with which the Parallels Server for Mac Bare Metal Edition software is compatible.

Physical Server Availability Considerations

The availability of a physical server running Parallels Server for Mac Bare Metal Edition is more critical than the availability of a typical PC server. Since it runs multiple virtual machines providing a number of critical services, physical server outage might be very costly. It can be as disastrous as the simultaneous outage of a number of servers running critical services.

To increase physical server availability, we suggest that you follow the recommendations below:

- Use a RAID storage for critical virtual machines. Do prefer hardware RAIDs, but software mirroring RAIDs might suit too as a last resort.

Note: The current version of Parallels Server for Mac Bare Metal Edition does not support the installation on an Xserve with the Apple RAID Card.

- Do not run any software on the server itself. Create special virtual machines where you can host necessary services such as BIND, FTPD, HTTPD, and so on. On the server, you need only the SSH daemon. Preferably, it should accept connections from a pre-defined set of IP addresses only.
- Do not create users on the server itself. You can create as many users as you need in any virtual machine. Remember: compromising the server means compromising all virtual machines as well.

CHAPTER 3

Operations on Virtual Machines

This chapter describes how to perform day-to-day operations on your virtual machines.

Note: We assume that you have successfully installed, configured, and deployed your Parallels Server for Mac Bare Metal Edition system. If you have not, refer to the *Parallels Server for Mac Bare Metal Edition Installation Guide* providing detailed information on these operations.

In This Chapter

Creating a Virtual Machine	18
Performing Initial Configuration.....	19
Starting, Stopping, and Querying Status of a Virtual Machine.....	22
Listing Virtual Machines.....	23
Storing Extended Information on a Virtual Machine	24
Copying a Virtual Machine Within the Server.....	25
Suspending a Virtual Machine	26
Pausing a Virtual Machine	26
Running Commands in a Virtual Machine.....	27
Deleting a Virtual Machine	27
Managing Virtual Machine Backups.....	28
Migrating Virtual Machines	36
Managing Templates.....	41
Managing Virtual Machine Disks	43
Managing Virtual Machine Devices	46
Managing Snapshots	53
Making Screenshots	57
Managing Virtual Machine Resources.....	58

Creating a Virtual Machine

To create a new virtual machine in Parallels Server for Mac Bare Metal Edition, do the following:

- 1 Create a virtual machine configuration. To create a virtual machine configuration, you can use either the `pctl create` command or Parallels Management Console.
- 2 Install an operating system in the virtual machine. This operation can be performed using Parallels Management Console only.

The example below shows you how to create a new virtual machine configuration using `pctl create`:

```
# pctl create MyVM --distribution win-2008 --location /vz/VMs
Creating the virtual machine...
Generate the VM configuration for win-2008.
The VM has been successfully created.
```

This will create a virtual machine with the name of `MyVM`, adjust its configuration for installing the Windows Server 2008 operating system in it, and place all virtual-machine-related files in the `/vz/VMs` directory. Now you can use Parallels Management Console to install Windows Server 2008 OS in this virtual machine. For information on how you can do it, see the *Parallels Management Console User's Guide* or *Getting Started With Parallels Management Console* guide.

Note: For more information on options you can pass to `pctl create` when creating virtual machines, see the *Parallels Command Line Reference Guide*.

Supported Guest Operating Systems

Listed below are the operating systems that you can run in your virtual machines:

Mac

Mac OS X 10.5 Server Leopard

Mac OS X 10.6 Server Snow Leopard

Windows

Windows 7 (x32, x64)

Windows Server 2008 R2 (x32, x64)

Windows Server 2003 R2 (x32, x64)

Windows Vista with Service Pack 1 and 2 (x32, x64)

Windows XP with Service Pack 2 and 3 (x32, x64)

Windows 2000 with Service Pack 4 (x32, x64)

Linux

Red Hat Enterprise Linux 5.3 (x32, x64)

Red Hat Enterprise Linux 4.7 (x32, x64)

Fedora 11

Fedora 10

CentOS 5.3 (x32, x64)

CentOS 4.7 (x32, x64)

SUSE Linux Enterprise Server 10 (x32, x64)

Debian GNU/Linux 5.0 (x32, x64)

Debian GNU/Linux 4.0 (x32, x64)

Ubuntu Linux 9.04 Server (x32, x64)

Ubuntu Linux 8.10 Server (x32, x64)

BSD

FreeBSD 7 (x32, x64)

FreeBSD 6 (x32, x64)

Performing Initial Configuration

Before starting your newly created virtual machine, you first need to configure it. This section describes the configuration steps for virtual machines.

Configuring Network Settings

To be accessible from the network, a virtual machine must be assigned a correct IP address. DNS servers must also be configured. The session below illustrates setting the main network parameters for a virtual machine having the name of MyVM:

- To set the IP address of 10.0.186.1, run this command:

```
# pct1 set MyVM --device-set net0 --ipadd 10.0.186.1
```

(net0 is the network card to assign the IP address to.)

- To set the DNS server IP address of 192.168.1.165, run this command:

```
# pct1 set MyVM --nameserver 192.168.1.165
```

Note: You can configure the network settings only inside virtual machines that have Parallels Tools installed.

Setting the Password for a Virtual Machine

In Parallels Server for Mac Bare Metal Edition, you can use the `--userpasswd` option of the `pctl set` command to create new accounts in your virtual machines directly from the Parallels server. The created account can then be used to log in to the virtual machine. The easiest way of doing it is to run this command:

```
# pct1 set MyVM --userpasswd user1:2wsx123qwe
```

This command creates the `user1` account in the MyVM virtual machine and sets the `2wsx123qwe` password for it. Now you can log in to the MyVM virtual machine as `user1` and administer it in the same way you would administer a standalone server: install additional software, add users, set up services, and so on.

The `pctl set` command can also be used to change passwords for existing accounts in your virtual machines. For example, to change the password for `user1` in the MyVM virtual machine to `0pi65jh9`, run this command:

```
# pct1 set MyVM --userpasswd user1:0pi65jh9
```

Note: You can use manage user accounts only inside virtual machines that have Parallels Tools installed.

Setting Startup Parameters

The `pctl set` command allows you to define the `onboot` startup parameter for virtual machines. Setting this parameter to `yes` makes your virtual machine automatically boot at the physical server startup. For example, to enable the MyVM virtual machine to start automatically on your server boot, you can execute the following commands:

```
# pct1 set MyVM --onboot yes
```

Notice that the `onboot` parameter will have effect only on the next server startup.

Installing Parallels Tools

If you are creating a Parallels virtual machine, you are also recommended to install Parallels Tools. Parallels Tools are a set of special utilities that help you use your virtual machines in the most comfortable and efficient way. With Parallels Tools, you can move the mouse seamlessly outside the guest OS window without pressing any key, change the virtual machine's screen resolution by simply resizing its window, synchronize your virtual machine's time and date settings with the time setting of the host computer, and share clipboard of your computer with the virtual machine's clipboard.

Parallels Tools are available for the following guest operating systems:

Windows

- Windows 2000
- Windows Server 2003
- Windows XP
- Windows Vista
- Windows Server 2008

Linux

Any supported Linux guest operating systems that have the following packages installed:

- `x.org 6.7` and later
- `glibc2.4` and later

Mac

Any supported Mac OS X guest operating systems.

Installing Parallels Tools

To install Parallels Tools in a virtual machine, use the `pctl installtools` command. For example, to install these tools in the `MyVM` virtual machine, you can run this command:

```
# pctl installtools MyVM
```

For more information on Parallels Tools, refer to the *Parallels Management Console User's Guide*.

Starting, Stopping, and Querying Status of a Virtual Machine

After a Parallels virtual machine has been created, it can be managed like an ordinary computer.

Starting a Virtual Machine

You can use the `pctl start` command to start your virtual machines. For example, to start a virtual machine with the name of `MyVM`, you can run this command:

```
# pctl start MyVM
Starting the VM ...
```

Stopping a Virtual Machine

The `pctl stop` command is used to stop your virtual machines. For example, to stop a virtual machine with the name of `MyVM`, you can run this command:

```
# pctl stop MyVM
Stopping the VM ...
```

Checking the Status of a Virtual Machine

You can use the `pctl list` command to check the status of a virtual machine:

```
# pctl list MyVM
stopped 10.12.12.121 MyVM
```

You can also get more detailed information on a virtual machine by specifying the `-i` option after `pctl list`.

Restarting a Virtual Machine

Sometimes, you may need to restart a virtual machine. To do this, use the `pctl reset` command:

```
# pctl reset MyVM
```

Listing Virtual Machines

To get an overview of the virtual machines existing on the physical server and to get additional information about them - their IP addresses, hostnames, current resource consumption, and so on - use the `pctl list` command. In the most general case, you may get a list of all virtual machines by issuing the following command:

```
# pctl list -a
                ID      NPROC  STATUS  IP_ADDR  HOSTNAME
{b8cb6d99-1af1-453d-a302-2fddd8f86769}  -   stopped  10.10.100.1  VM_1
```

The `-a` option tells the `pctl list` command to output both running and stopped virtual machines. By default, only running virtual machines are shown. The default columns inform you of the virtual machines IDs, the number of running processes inside virtual machines, their status, IP addresses, and hostnames. This output may be customized as desired by using `pctl list` command line options. For example:

```
# pctl list -a -o name,ctid
NAME                ID
My_Vm      {b8cb6d99-1af1-453d-a302-2fddd8f86769}
```

This command displays only the names and IDs of the virtual machines existing on the physical server. The full list of the `pctl list` command options for virtual machines is available in the *Parallels Command Line Reference Guide*.

Storing Extended Information on a Virtual Machine

Sometimes, it may be difficult to remember the information on certain virtual machines. The probability of this increases together with the number of virtual machines and with the time elapsed since their creation. Parallels Server for Mac Bare Metal Edition allows you to set the description of any virtual machine on the physical server and view it later on, if required. The description can be any text containing any virtual machine-related information. For example, you can include the following in the virtual machine description:

- the owner of the virtual machine
- the purpose of the virtual machine
- the summary description of the virtual machine

Let us assume that you are asked to create a virtual machine for a Mr. Johnson who is going to use it for hosting the MySQL server. So, you create the `MyVM` virtual machine and, after that, execute the following command on the physical server:

```
# pct1 set MyVM --description "MyVM
> owner - Mr. Johnson
> purpose - hosting the MySQL server" -
The VM has been successfully configured.
```

This command saves the following information related to the virtual machine: its name, owner, and the purpose of its creation. At any time, you can display this information by issuing the following command:

```
# pct1 list -o description MyVM
MyVM
owner - Mr. Johnson
purpose - hosting the MySQL server
```

When working with virtual machine descriptions, keep in mind the following:

- You can use any symbols you like in the virtual machine description (new lines, dashes, underscores, spaces, etc.).
- If the virtual machine description contains one or more spaces or line breaks (as in the example above), it must be put in single or double quotes.
- As distinct from a virtual machine name and ID, a description cannot be used for performing virtual machine-related operations (e.g. for starting or stopping a virtual machine) and is meant for reference purposes only.

Copying a Virtual Machine Within the Server

Parallels Server for Mac Bare Metal Edition allows you to create a complete copy of a particular virtual machine (in respect of all the virtual machine data and resources parameters), or a *clone*. This saves your time because you do not have to think of setting up the virtual machine configuration parameters and the like. Moreover, you can create a number of virtual machine clones at a sitting.

In Parallels Server for Mac Bare Metal Edition-based systems, you can use the `pctl clone` command to copy a virtual machine within the given physical server. For example, you can create a clone of the `MyVM` virtual machine and assign the `Cloned_VM` name to it as follows:

```
# pctl clone MyVM --name ClonedVM
Clone the MyVM VM to the VM ClonedVM...
The VM has been successfully cloned.
```

Notice that you can create clones of stopped virtual machines only.

Checking the Cloned Virtual Machine

To check that your virtual machine has been successfully moved, run this command:

```
# pctl list -a
STATUS      IP_ADDR      NAME
stopped     10.0.10.115  MyVM
stopped     10.0.10.115  ClonedVM
```

As you can see from the example above, the clone the `MyVM` virtual machine (`ClonedVM`) has been successfully created. However, before starting to use this clone, you should assign a different IP address to it because the current IP address is identical to that of `MyVM`. Refer to [Performing Initial Configuration](#) (p. 19) to learn how you can do it.

Configuring the Default Directories

When cloning a virtual machine, you can also override the `/vz/dest_VM_Name.pvm` directory used by default to store the files of a cloned virtual machine (where `dest_VM_Name` denotes the name of the resulting virtual machine). For example, for the `ClonedVM` virtual machine, this directory is `/vz/ClonedVM.pvm`. To store the files of the `ClonedVM` virtual machine in a different directory, you can run the following command:

```
# pctl clone MyVM --name ClonedVM --location /vz/VM_directory
```

In this case all virtual machine files will be placed to the `/vz/VM_directory` directory. Notice that the specified directory must exist on the server; otherwise, the command will fail.

Suspending a Virtual Machine

Parallels Server for Mac Bare Metal Edition allows you to suspend a running virtual machine on the physical server by saving its current state to a special file. Later on, you can resume the virtual machine and get it in the same state the virtual machine was at the time of its suspending. Suspending your virtual machines may prove useful, for example, if you need to restart the physical server, but do not want to:

- quit the applications currently running in the virtual machine
- spend much time on shutting down the guest operating system and then starting it again

You can use the `pctl suspend` command to save the current state of a virtual machine. For example, you can issue the following command to suspend the `MyVM` virtual machine:

```
# pctl suspend MyVM
Suspending the VM...
The VM has been successfully suspended.
```

At any time, you can resume the `MyVM` virtual machine by executing the following command:

```
# pctl resume MyVM
Resuming the VM...
The VM has been successfully resumed
```

Once the restoration process is complete, any applications that were running in the `MyVM` virtual machine at the time of its suspending will be running again and the information content will be the same as it was when the virtual machine was suspended.

Pausing a Virtual Machine

Pausing a running virtual machine releases the resources, such as RAM and CPU, currently used by this virtual machine. The released resources can then be used by the Parallels server or other running virtual machines.

To pause a virtual machine, you can use the `pctl pause` command. For example, the following command pauses the `My_VM` virtual machine:

```
# pctl pause My_VM
Pause the VM...
The VM has been successfully paused.
```

You can check that the virtual machine has been successfully paused by using the `pctl list -a` command:

```
# pctl list -a
STATUS  IP_ADDR      NAME
running 10.10.10.101 101
paused  10.10.10.201 My_VM
```

The command output shows that the `My_VM` virtual machine is paused at the moment. To continue running this virtual machine, execute this command:

```
# pctl start My_VM
Starting the VM...
The VM has been successfully started.
```

Running Commands in a Virtual Machine

Parallels Server for Mac Bare Metal Edition allows you to execute arbitrary commands inside virtual machines by running them on the physical server, i.e. without the need to log in to the respective virtual machine. For example, this can be useful in these cases:

- If you do not know the virtual machine login information, but need to run some diagnosis commands to verify that it is operational.
- If network access is absent for a virtual machine.

In both these cases, you can use the `pctl exec` command to run a command inside the respective virtual machine. The session below illustrates the situation when you run the stopped SSH daemon inside a Linux virtual machine with the name of `My_Linux`:

```
# pctl exec My_Linux /etc/init.d/sshd status
sshd is stopped
# pctl exec My_Linux /etc/init.d/sshd start
Starting sshd:[OK]
# pctl exec My_Linux /etc/init.d/sshd status
sshd (pid 26187) is running...
```

Notes:

1. You can use the `pctl exec` command only inside virtual machines that have Parallels Tools installed.
 2. The `pctl exec` command is executed inside a virtual machine from the `/` directory rather than from the `/root` one.
-

Deleting a Virtual Machine

You can delete a virtual machine that is not needed anymore using the `pctl delete` command. Notice that you cannot delete a running or mounted virtual machine. For example, you can run these commands to delete the `MyVM` virtual machine:

```
# pctl delete MyVM
Deleting the VM...
VM is currently running
# pctl stop MyVM
Stopping the VM...
VM was stopped
# pctl delete MyVM
Deleting the VM...
VM was deleted
```

Managing Virtual Machine Backups

A regular backing up of the existing virtual machines is essential for any physical server reliability. In Parallels Server for Mac Bare Metal Edition, you can use the following utilities to back up and restore your virtual machines:

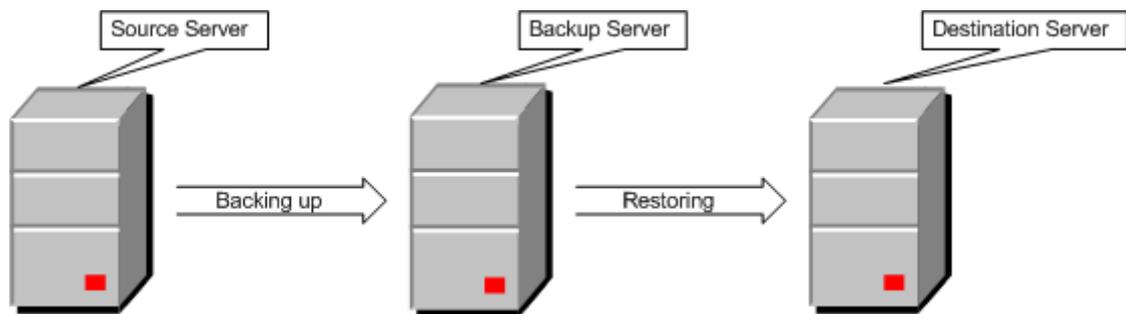
- `pctl`
- `pbackup`
- `prestore`

Detailed information on these utilities is provided in the following subsections.

Backups Overview

Parallels Server for Mac Bare Metal Edition backup utilities deal with three kinds of servers:

- *Source Server*. This is the server where virtual machines are hosted during their backing up.
- *Backup Server*. This is the server where virtual machine backups are stored.
- *Destination Server*. This is the server where virtual machine backups are restored.



These servers are singled out by their functionality only. In reality, one and the same physical server can perform two or even three functions. Usually, the Source and Destination Servers are represented by one and the same server because you will likely want the virtual machines you back up to be restored to their original server. However, setting up a dedicated Backup Server is recommended.

Creating Consistent Backups of Virtual Machines

Parallels Server for Mac Bare Metal Edition allows you to back up both running and stopped virtual machines. However, to create a consistent backup of a running virtual machine, it must meet the following requirements:

- Have Parallels Tools installed.
- Run one of the following operating systems:

Windows operating systems

- Windows Server 2003
- Windows Server 2008
- Windows Vista
- Windows 7

Linux operating systems

- Suse, version 9.0 and higher
- RHEL, version 4.0 and higher
- CentOS, version 4.0 and higher
- Fedora Core, version 3 and higher
- Debian, version 3.1 and higher
- Ubuntu, version 4.10 and higher

Using `pctl backup` and `pctl restore`

This section describes how to perform the basic backup-related operations using the `pctl` utility.

Creating a Virtual Machine Backup

You can use the `pctl backup` command to back up your virtual machines. This command is executed on the Source Server and can store the created virtual machine backup on both the Source and Backup Servers. When creating a backup on the Source Server, you only need to specify the name of the virtual machine to back up. For example, you can execute the following command to back up the `MyVM` virtual machine and store its backup archive on the Source Server:

```
# pctl backup MyVM
Backing up the VM MyVM
Operation progress 100%
The virtual machine has been successfully backed up with backup ID {746dba2a-3b10-4ced-9dd6-76a2b1c14a69}
```

The command output informs you that the virtual machine backup has been successfully created and assigned ID `746dba2a-3b10-4ced-9dd6-76a2b1c14a69`. You can use this ID when managing the backup archive (e.g. remove the backup).

At the same time, you can run the following command to back up the `MyVM` virtual machine and store its backup archive on the Backup Server with the IP address of `129.129.10.10`:

```
# pctl backup MyVM -s root:1qaz2wsx@129.129.10.10
```

`root:1qaz2wsx` before the Destination Server IP address denotes the root credentials used to log in to this server. If you do not specify these credentials, you will be asked to do so during the command execution.

All newly created backups are placed to the directory specified as the value of the `BACKUP_DIR` parameter in the `/etc/vzbackup.conf` configuration file. By default, this directory is `/vz/backups/ID` (where `ID` is the ID of the respective virtual machine).

Note: For more information on the options you can pass to `pctl backup`, refer to the *Parallels Command Line Reference Guide*.

Listing the Existing Backups

You can use the `pctl backup-list` command to view the backups existing on the physical server. For example:

```
# pctl backup-list
Node          Date      Type      ID          Backup_ID
{c1dee22f-8667-4870-9e11-278f1398eab0} {209d54a0-e3b8-4a03-9ca8-d4cc7a2a27ca}
test.com     2009-06-30 10:19:32   f          411566405
```

This command lists the backups existing on the Source Server. If you want to list the backups on the Backup Server, you need to specify the IP address of this server.

The command output shows that currently only two backups exist on the Source Server. These backups were created for a virtual machine with the ID of `c1dee22f-8667-4870-9e11-278f1398eab0`. The information on the backups is presented in the following table:

Column Name	Description
ID	The ID uniquely identifying the virtual machine.
Backup ID	The ID assigned to the backup archive. You need to specify this ID when performing any backup-related operations.
Node	The hostname of the physical server storing the backup archive.
Date	The date and time when the backup archive was created.
Type	The backup type. Currently, you can create two types of backups: <ul style="list-style-type: none"> ▪ A full backup indicated by <code>f</code>. ▪ An incremental backup indicated by <code>i</code> and containing only the files changed since the previous full or incremental backup. This is the default backup type.
Size	The size of the backup archive, in bytes.

Removing a Virtual Machine Backup

At any time, you can remove a backup that you do not need any more using the `pctl backup-delete` command. To do this, you need to specify the ID of the backup to remove and the ID of the respective virtual machine. If you do not know these IDs, use the `pctl backup-list` and check the ID and Backup ID columns. For example:

```
# pctl backup-list
Node          Date      Type      ID          Backup_ID
{c1dee22f-8667-4870-9e11-278f1398eab0} {209d54a0-e3b8-4a03-9ca8-d4cc7a2a27ca}
test.com     2009-06-30 10:19:32   f          411566405
# pctl backup-delete c1dee22f-8667-4870-9e11-278f1398eab0 -t 209d54a0-e3b8-4a03-9ca8-d4cc7a2a27ca
Delete the VM backup
The VM backup has been successfully removed.
```

You can also specify the virtual machine name instead of its ID:

```
# pctl backup-delete MyVM -t 209d54a0-e3b8-4a03-9ca8-d4cc7a2a27ca
```

If you have several backups of a particular virtual machine and want to delete them all at once, indicate only the virtual machine name or ID:

```
# pctl backup-delete MyVM
```

This command removes all backups of the `MyVM` virtual machine from the local Backup Server. To remove backups stored remotely, you also need to specify the IP address of the remote Server:

```
# pctl backup-delete MyVM -s root:lqaz2wsx@129.129.10.10
```

Restore a Virtual Machine

To restore a backup of a virtual machine, you can use the `pctl restore` command. This command supports restoring backups to the Source Server only. For example, to restore a backup of the `MyVM` virtual machine stored on the Backup Server with the IP address of `10.10.100.1`, you can run this command on the Source Node:

```
# pctl restore MyVM -s root:lqaz2wsx@10.10.100.1
```

If you have two or more backups of the MyVM virtual machine, the latest backup is restored. If you want to restore a particular virtual machine backup, you need to specify the ID of this backup. You can use the `pctl backup-list` command to list the existing backups and the IDs assigned to them:

```
# pctl backup-list -s root:lqaz2wsx@10.10.100.1
```

Node	Date	Type	ID	Size	Backup_ID
{c1dee22f-8667-4870-9e11-278f1398eab0}				{209d54a0-e3b8-4a03-9ca8-d4cc7a2a27ca}	
test.com	2009-06-30 10:19:32	i		11566405	
{c1dee22f-8667-4870-9e11-278f1398eab0}				{24a3011c-092e-4f21-bb3b-29ccfe967e92}	
test.com	2009-05-21 11:12:35	f		356798701	

You can now indicate the desired ID after the `-t` option to tell `pctl backup` to restore this particular backup. For example, to restore the backup for the virtual machine with the ID of `c1dee22f-8667-4870-9e11-278f1398eab0` that was created on the 21st of May, you can execute this command:

```
# pctl restore -t {c1dee22f-8667-4870-9e11-278f1398eab0} -s  
root:lqaz2wsx@10.10.100.1
```

Using pbackup and prestore

Along with `pctl`, you can use the following utilities to create and manage backups of your virtual machines:

- `pbackup`. This utility is used to create backups of individual virtual machines or entire Parallels servers.
- `prestore`. This utility is used to manage the existing backups of virtual machines.

Backing Up Virtual Machines

The `pbackup` utility is run on the Backup Server connecting via SSH to the Parallels server and backing up one or more virtual machines on this server. The created backup archive is then placed to the directory on the Backup Server defined in the `/etc/vzbackup.conf` global backup configuration file. By default, this directory is `/vz/backups`. Later on, the virtual machine backups can be restored from this directory.

Assuming that you are going to back up the entire Parallels server (i.e. all virtual machines on this server) with the `test.com` hostname, you can run the following command on the Backup Server:

```
# pbackup test.com
```

During the command execution, you will be asked to provide the `test.com` credentials. After doing so, the command will back up all virtual machines on the `test.com` and put all backed up virtual machines to the Source Server.

To save the backed up virtual machines also on the Backup Server, you should additionally specify the `-n` option. This option is used to indicate the IP address or hostname of the Backup Server and its credentials:

```
# pbackup -n root:7ujn6yhb@192.168.10.199 test.com
```

If you wish to back up not all, but specific virtual machines from the specified server, use the `-e` or `-x` switches (to include or exclude the specified virtual machines, respectively). For example:

```
# pbackup -n root:7ujn6yhb@192.168.10.199 test.com -e MyVM
```

In this session, only the `MyVM` virtual machine residing on the Source Server with the `test.com` hostname will be included in the backup, and their backups will be stored on the Backup Server.

For the full list of configuration parameters and command line options for `pbackup`, refer to the *Parallels Command Line Reference Guide*.

Restoring Backups

To restore any individual virtual machines or entire Parallels servers, you may want to view first the information about them. This can be done using the `prestore -l` command:

```
# prestore -l -n test.com test.com
root@test.com's password:
...
Backups for node test.com:
Node      Date              Type      ID Backup_ID
          Size
```

```
{cd91b90b-469d-42c6-acf4-fefee09cfa61} {4ef87485-ec3b-4594-896b-c7ccb859b5}
test.com 2009-07-16 17:15:47 f 92617398
```

The command output shows that currently only two backups exist for the `test.com` server on the Backup Server. If you omit the `-n test.com` option, the command will list all virtual machine backups for the `test.com` server stored on the `test.com` server.

The information on the backups is presented in the following table:

Column Name	Description
ID	The ID uniquely identifying the virtual machine.
Backup ID	The ID assigned to the backup archive. You need to specify this ID when performing any backup-related operations.
Node	The hostname of the Source Server.
Date	The date and time when the backup archive was created.
Type	The backup type. Currently, you can create two types of backups: <ul style="list-style-type: none"> ▪ A full backup indicated by <code>f</code>. ▪ An incremental backup indicated by <code>i</code> and containing only the files changed since the previous full or incremental backup. This is the default backup type.
Size	The size of the backup archive, in bytes.

To restore the `{cd91b90b-469d-42c6-acf4-fefee09cfa61}` virtual machine, run this command:

```
# prestore -n test.com -e 101 {cd91b90b-469d-42c6-acf4-fefee09cfa61}
```

This command will restore the virtual machine to their Source Server.

Notes:

1. The current version of Parallels Server for Mac Bare Metal Edition supports restoring virtual machines to the Source Server only.
 2. The `prestore` utility can also manage (list, restore, etc.) backups created using the `pctl` backup command. However, you are recommended to use the same utility (either `pctl` or `prestore`) during the life cycle of a particular backup.
 3. For the full list of command line options for `prestore`, refer to the *Parallels Command Line Reference Guide*.
-

Configuring Per-Server Backup Parameters

A number of default parameters in the global backup configuration file can be adjusted for a particular physical server to be backed up. To do this:

- 1 Create a new configuration file named `server.conf`.
- 2 Put the file to the backup directory. This directory is defined by the `BACKUP_DIR` parameter in the `/etc/vzbackup.conf` global backup configuration file and is set to `/vz/backups` by default.

The `server.conf` file should contain those parameters that you want to rewrite for a given Parallels server. For a complete list of those backup parameters that can be configured using per-server and configuration files, refer to the *Parallels Command Line Reference Guide*.

Configuring Passwordless Access to the Source Node

You need to provide the Source Server credentials each time you execute the `pbackup` and `prestore` commands. However, you can allow these utilities to log in to the Source Server without having to enter the `root` password. To do this, you should provide each Source Server with authorized public SSH RSA keys:

- 1 Log in to the Backup Server as `root`, and generate a pair of SSH keys - public and private:

```
# ssh-keygen -t rsa
Generating public/private rsa key pair.
Enter file in which to save the key (/root/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /root/.ssh/id_rsa.
Your public key has been saved in /root/.ssh/id_rsa.pub.
The key fingerprint is:
c6:19:a8:2c:67:31:15:e6:30:23:2b:8a:b0:63:77:8f root@dhcp-130.parallels.com
```

Note that you must leave an empty passphrase in the above procedure. The private key is saved by default in `/root/.ssh/id_rsa`, and the public key is saved in `/root/.ssh/id_rsa.pub`.

- 2 Transfer your public key to the `/root/.ssh` directory on each Source Server (use some intermediary name for the file not to overwrite the corresponding file on the Source Server):

```
# scp /root/.ssh/id_rsa.pub root@dhcp-129.parallels.com:/root/.ssh/temp_name
The authenticity of host 'dhcp-129.parallels.com (192.168.1.129)' can't be
established.
RSA key fingerprint is 01:fc:b6:e9:26:40:1f:1a:41:5f:7a:fb:cf:14:51.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added 'dhcp-129.parallels.com,192.168.1.129' (RSA) to the
list of known hosts.
root@dhcp-129.parallels.com's password:
id_rsa.pub      100% |*****|          235      00:00
```

- 3 Add the contents of the transferred file to the `authorized_keys` file in this very directory on the Source Node. To do this, log in to the Source Server, change to the `/root/.ssh` directory, and issue the following command:

```
# cat temp_name >> authorized_keys
```

Now the `pbackup/prestore` utilities should be able to log in to your Source Nodes as `root` without having to provide the `root` password.

Migrating Virtual Machines

The Parallels physical server is the system with higher availability requirements in comparison with a typical system. If you are running your company mail server, file server, and web server in different virtual machines on one and the same physical server, then shutting it down for hardware upgrade will make all these services unavailable at once. To facilitate hardware upgrades and load balancing between several Parallels servers, the Parallels Server for Mac Bare Metal Edition software provides you with the ability to migrate virtual machines from one physical server to another.

Parallels Server for Mac Bare Metal Edition is shipped with a special utility - `pmigrate` - allowing you to perform different types of migration. Using this utility, you can migrate

- virtual machines from one physical server to another
- a physical server to a virtual machine

All these operations are described in the following subsections.

General Migration Requirements

Before deciding on the type of migration to perform, make sure that the source computer (i.e. the physical computer that you will migrate or that stores the virtual machine before its migration) and the destination computer (i.e. the computer that runs Parallels Server for Mac Bare Metal Edition and that will host the resulting virtual machine) meet the requirements below.

Requirements for the Source Computer

The source computer can be a physical computer or a virtual machine. The software requirements for source computers are given in the following table:

<u>Operating System</u>	<u>Physical Computer</u>	<u>Virtual Machine</u>
Windows		
Windows 7 (x32, x64)	+	+
Windows Server 2003 (x32, x64)	+	+
Windows Server 2008 (x32, x64)	+	+
Windows 2000 Server (x32)	+	+
Windows XP (x32, x64)	+	+
Windows Vista (x32, x64)	+	+
Linux		
Red Hat Enterprise Linux 5 (x32, x64)	+	+
Red Hat Enterprise Linux 4 (x32, x64)	+	+
CentOS 5 (x32, x64)	+	+
CentOS 4 (x32, x64)	+	+
Fedora 11 (x32, x64)	+	+
Fedora 10 (x32, x64)	+	+
SUSE Linux Enterprise Server 10 (x32, x64)	+	+
Debian GNU/Linux 5 (x32, x64)	+	+
Debian GNU/Linux 4 (x32, x64)	+	+
Ubuntu Linux 9.04 Server (x32, x64)	+	+
Ubuntu Linux 8.10 Server (x32, x64)	+	+

Requirements for the destination Server

The destination server must meet the following requirements:

- Has enough hard disk space to store the resulting virtual machine.
- Has enough memory and CPU power to run the resulting virtual machine.
- Has a stable network connection with the source server.

Migrating Virtual Machines Between Parallels Servers

The standard migration procedure allows you to move stopped, paused, suspended, and running virtual machines. Migrating a stopped, paused, or suspended virtual machine includes copying all virtual machine-related files from one Parallels server to another and does not differ from copying a number of files from one server to another over the network. In its turn, the migration procedure of a running virtual machine is a bit more complicated and can be described as follows:

- 1 After initiating the migration process, all virtual machine data are copied to the destination server. During this time, the virtual machine on the source server continues running.
- 2 The virtual machine on the source server is stopped.
- 3 The virtual machine data copied to the destination server are compared with those on the source server, and if any files were changed during the first migration step, they are copied to the destination server again and rewrite the outdated versions.
- 4 The virtual machine on the destination server is started.

There is a short downtime needed to stop the virtual machine on the source server, copy the virtual machine data changes to the destination server, and start the virtual machine on the destination server. However, this time is very short and almost unnoticeable to users.

The following session shows you how to move the MyVM virtual machine from the local Parallels server to the destination server `ts7.test.com`:

```
# pmigrate v MyVM v ts7.test.com/MyVM
Migrate the VM MyVM to test.com
root@ts7.test.com's password:
Operation progress 100%
The VM has been successfully migrated.
```

You can also use `pmigrate` to migrate the MyVM virtual machine from a remote Parallels server to the local one:

```
# pmigrate v ts7.test.com/MyVM v localhost
root@ts7.test.com's password:
Migrate the VM MyVM to localhost
Operation progress 100%
The VM has been successfully migrated.
```

This command move the MyVM virtual machine from the `ts7.test.com` server to the local server.

Note: For more information on options that you can pass to `pmigrate`, refer to the *Parallels Command Line Reference*.

Migrating a Physical Computer to a Virtual Machine

You can also use the `pmigrate` utility to move a stand-alone physical computer to a virtual machine. The migration process includes copying the whole contents of the physical computer (i.e. all its files, directories, quota limits, configuration settings, and so on) to a virtual machine on the Parallels server. After migrating the computer, you will have its exact copy in a virtual machine including the operating system, the IP addresses assigned, the amount of available disk space and memory, etc.

Moving a physical computer to a virtual machine involves completing the following steps:

- 1 Installing the Parallels agent on the physical computer you want to migrate. This step is required only if you are migrating the physical computer to a virtual machine.
- 2 Migrating the physical computer by running the `pmigrate` utility on the server.

Installing the Agent

If you are planning to migrate a physical computer to a virtual machine, you must first install the Parallels agent on this computer. This agent collects essential system data on the physical computer and transfers it to the `pmigrate` utility on the Parallels server. To install the Parallels agent, do the following:

- 1 Make sure that your physical computer meets the necessary requirements for installing the Parallels agent. See *Requirements for Migrating to Virtual Machines* (p. 40) for details.
- 2 Log in to the physical computer as a user with administrative rights.
- 3 Copy the Parallels agent installation file to the physical computer. The installation file is located in the `/usr/share/pmigrate/tools` directory on the Parallels server:
 - `parallels-transporter-agent-XXXX.run`. Use this file to install the Parallels agent on computers running a Linux operating system.
 - `ParallelsTransporterAgent-parallels-XXXX.exe`. Use this file to install the Parallels agent on computers running a Windows operating system.
- 4 Execute the installation file on the physical computer.
- 5 Follow the instructions of the wizard to install the Parallels agent.
- 6 Restart the source computer to complete the installation.

Note: The Parallels agent is automatically launched after the restart, so you do not need to start it manually.

Migrating the Server

Once the physical computer is up and running, you can migrate to a virtual machine on the Parallels server. For example, you can move a physical computer to a virtual machine by running the following command on the destination server:

```
# pmigrate h root:1qsde34rt@192.168.1.130 v MyVM
```

where

- `h` denotes that you are migrating a physical computer.

- `root:lqsde34rt@192.168.1.130` is the IP address and credentials of the physical computer to be migrated.

You can omit the credentials in the command above. In this case you will be asked to provide them during the command execution.

- `v` indicates that the physical computer is to be moved to a virtual machine.
- `MyVM` is the name of the resulting virtual machine on the Parallels server.

Once the command is complete, you will find the resulting virtual machine in the `/var/parallels` directory on the Parallels server.

Requirements for Migrating to Virtual Machines

Any physical computer that you plan to migrate to a virtual machine must have the Parallels agent installed. The agent can be installed on computers meeting the following requirements.

Hardware Requirements

- 700 (or higher) MHz x86 or x64 processor (Intel or AMD).
- 256 MB or more RAM.
- 50 MB of hard disk space for installing the Parallels agent package.
- Ethernet or WiFi network adapter.

Software Requirements

For software requirements, see the table in [General Migration Requirements](#) (p. 37).

Additional Requirements for Migrating Servers with Parallels Server for Mac Bare Metal Edition

If you plan to migrate a server running the PParallels Server for Mac Bare Metal Edition software, you should first make sure that the `snapapi26` and `snumbd26` modules are not loaded on the server. You can use the following command to check this:

```
# lsmod | grep snapapi26
# lsmod | grep snumbd26
```

If any of these modules are loaded, unload them by running the `rmmod` command. Only after they are unloaded, proceed with migrating the server.

Notes:

1. Migrating Windows dynamic volumes and Linux logical volumes (LVM) is not supported.
 2. You may also try to migrate servers with unsupported file systems. However, in this case all disk sectors are copied successively, and you may experience problems with using the resulting virtual machine.
-

Managing Templates

A template in Parallels Server for Mac Bare Metal Edition is a pre-configured virtual machine that can be easily and quickly deployed into a fully functional virtual machine. Like any normal virtual machine, a template contains hardware (virtual disks, peripheral devices) and the operating system. It can also have additional software installed. In fact, the only main difference between a virtual machine and a template is that the latter cannot be started.

In Parallels Server for Mac Bare Metal Edition, you can perform the following operations on templates:

- create a new template
- list the existing templates
- create a virtual machine from a template

These operations are described in the following subsections in detail.

Create a Template

In Parallels Server for Mac Bare Metal Edition, you can create a virtual machine template using the `pctl clone` utility. Making a template may prove useful if you need to create several virtual machines with the same configuration. In this case, your steps can be as follows:

- 1 You create a virtual machine with the required configuration.
- 2 You make a template on the basis of the created virtual machine.
- 3 You use the template to create as many virtual machines as necessary.

Let us assume that you want to create a template of the `My_VM` virtual machine. To do this, you can run the following command:

```
# pctl clone My_VM --name template1 --template
Clone the My_VM VM to VM template template1...
Operation progress 98%
The VM has been successfully cloned.
```

This command clones the `My_VM` virtual machine and saves it as the `template1` template. After the template has been successfully created, you can use it for creating new virtual machines.

Listing Templates

Sometimes, you may need to get an overview of the virtual machine templates available on your Parallels server. For example, this may be necessary if you plan to create a virtual machine from a specific template, but do not remember its exact name. In this case, you can use the `pctl list` command to list all templates on the Parallels server and find the necessary one:

```
# pctl list -t
{4ad11c28-9f0e-4086-84ea-9c0487644026} win-2003      template1
{64bd8fea-6047-45bb-a144-7d4bba49c849} rhel        template3
{6d3c9d6f-921a-484d-9772-bc7096f68df1} win-2003      template2
```

In this example, 3 virtual machine templates exist on the Parallels server. The information on these templates is presented in the form of a table with the following columns (from left to right): the template ID, the operating system contained in the template, and the template name.

Deploying a Template

Though a template has the same components as a virtual machine (hardware, software, etc.), it cannot be started. To run a template as a virtual machine, you need first to convert the template. By converting a template, you create a virtual machine with the configuration identical to that of the template.

To convert a template into a virtual machine, use the `--ostemplate` option of the `pctl create` command. For example, to convert the `template1` template to a virtual machine with the `Converted_VM` name, you can run this command:

```
# pctl create Converted_VM --ostemplate template1
Creating the VM on the basis of the template1 template...
Clone the template1 VM to the VM Converted_VM...
Operation progress 99%
The VM has been successfully cloned.
```

To check that the `Converted_VM` virtual machine has been successfully created, use the `pctl list -a` command:

```
# pctl list -a
STATUS      IP_ADDR      NAME
running     10.12.12.101 111
stopped     10.12.12.34  Converted_VM
running     10.30.17.149 Windows XP
```

The template itself is left intact and can be used for creating other virtual machines:

```
# pctl list -t
{4ad11c28-9f0e-4086-84ea-9c0487644026} win-2003      template1
{64bd8fea-6047-45bb-a144-7d4bba49c849} rhel        template2
```

Managing Virtual Machine Disks

In Parallels Server for Mac Bare Metal Edition, you can manage virtual machine disks as follows:

- change the type of your virtual disks
- increase the capacity of your virtual disks
- reduce the capacity of your virtual disks
- reduce the size occupied by your virtual disks on the physical hard drive

All these operations are described in the following subsections in detail.

Changing the Disk Type

A virtual disk can be one of the two types:

- `plain`. A plain virtual hard disk has a fixed size from the moment of its creation.
- `expanding`. An expanding virtual hard disk is small initially. Its size grows as you add applications and data to it.

To change the type of a virtual disk in Parallels Server for Mac Bare Metal Edition, you can use the `pctl set` command. Let us assume that the current type of the `hdd0` virtual disk in the `MyVM` virtual machine is `plain` and you want to change it to `expanding`. In this case, you can execute the following command:

```
# pctl set MyVM --device-set hdd0 --type expand
```

To change the disk type back to `plain`, run this command:

```
# pctl set MyVM --device-set hdd0 --type plain
```

Increasing the Virtual Disk Capacity

If you find that the capacity of your virtual machine's hard disk does not fit your needs anymore, you can increase it using the `prl_disk_tool` utility. For example, you can execute the following command to set the capacity for the `MyVM-0.hdd` disk to 80 GB:

```
# prl_disk_tool resize --size 80000 --hdd /vz/MyVM.pvm/MyVM-0.hdd/  
Operation progress 100%
```

This command adds additional disk space as unallocated space. You can use standard means (e.g. the Disk Management tool in Windows-based virtual machines) to allocate this space by creating a new partition or expanding an existing one.

At the same time, you can use the `--resize_partition` option to automatically add additional space to the last partition on the virtual disk:

```
# prl_disk_tool resize --size 80000 --hdd /vz/MyVM.pvm/MyVM-0.hdd/ --  
resize_partition  
Operation progress 100%
```

When increasing the disk capacity, keep in mind the following:

- You cannot increase the capacity of a virtual disk if the virtual machine using this disk is running.
- The virtual machine using the virtual disk you want to configure must not have any snapshots. Otherwise, the operation will fail:

```
# prl_disk_tool resize --size 68000 --hdd /vz/MyVM.pvm/MyVM-0.hdd/  
This disk has one or more snapshots and cannot be resized.  
You need to delete snapshots using the pctl tool before resizing the disk.
```

In this case, you should delete all existing snapshots and run the command again. To learn how to delete virtual machine snapshots, refer to [Deleting a Snapshot](#) (p. 56).

- The capacity of an expanding virtual disk shown from inside the virtual machine and the size the virtual disk occupies on the server's physical disk may differ.

Reducing the Virtual Disk Capacity

Parallels Server for Mac Bare Metal Edition provides a possibility to reduce the size of an expanding virtual disk by setting the limit the disk cannot exceed. In general, the process of reducing a virtual disk includes these steps:

- 1 Finding out the minimum capacity to which the disk can be reduced.
- 2 Running the `prl_disk_tool resize` command to reduce the disk.

Checking the Minimum Disk Capacity

Before reducing a virtual disk, you may wish to see the minimum capacity to which it can be reduced. To do this, use the `prl_disk_tool resize --info` command. For example, you can run the following command to get detailed information on the `MyVM-0.hdd` disk:

```
# prl_disk_tool resize --info --hdd /vz/MyVM.pvm/MyVM-0.hdd
Operation progress 100 %
Disk information:
  Size:                65537M
  Minimum:             2338M
  Minimum without resizing the last partition: 65523M
```

The information on the virtual disk is presented in the form of the following table:

Column Name	Description
Size	The virtual disk disk capacity, in megabytes, as it is seen from inside the virtual machine.
Minimum	The virtual disk capacity, in megabytes, after resizing the disk using the <code>prl_disk_tool</code> utility with the <code>--resize_partition</code> option.
Minimum without resizing the last partition	The virtual disk capacity, in megabytes, after resizing the disk using the <code>prl_disk_tool</code> utility without the <code>--resize_partition</code> option.

Reducing the Disk Size

Once you know the minimum capacity of the virtual disk, you can start reducing it. For example, to reduce the `MyVM-0.hdd` disk to 30 GB, you can execute the following command:

```
# prl_disk_tool resize --size 30G --hdd /vz/MyVM.pvm/MyVM-0.hdd --
resize_partition
Operation progress 100 %
```

When reducing the disk capacity, keep in mind the following:

- You cannot reduce the capacity of a virtual disk if the virtual machine using this disk is running.
- The virtual machine using the virtual disk you want to configure must not have any snapshots. Otherwise, you will be informed of this fact:

```
# prl_disk_tool resize --size 68000 --hdd /vz/MyVM.pvm/MyVM-0.hdd/
This disk has one or more snapshots and cannot be resized.
You need to delete snapshots using the pctl tool before resizing the disk.
```

In this case, you should delete all existing snapshots and run the command again. To learn how to delete virtual machine's snapshots, refer to [Deleting a Snapshot](#) (p. 56).

- The capacity of an expanding virtual disk shown from inside the virtual machine and the size the virtual disk occupies on the server's physical disk may differ.

Compacting the Virtual Disk

In Parallels Server for Mac Bare Metal Edition, you can decrease the space your virtual machines occupy on the Parallels server's disk drive by compacting their virtual disks. Compacting virtual disks allows you to save your server's disk space and host more virtual machines on the server.

Note: Plain disk cannot be compacted.

To compact a virtual disk, you can use the `prl_disk_tool compact` command. For example, to compact the `MyVM-0.hdd` disk, you can run this command:

```
# prl_disk_tool compact --hdd /vz/MyVM.pvm/MyVM-0.hdd/  
Operation progress 100 %
```

To check the space that was freed by compacting the virtual disk, you can use standard Linux utilities (e.g. the `df` utility).

Managing Virtual Machine Devices

Parallels Server for Mac Bare Metal Edition allows you to manage the following virtual machine devices:

- hard disk drives
- CD/DVD-ROM drives
- floppy disk drives
- network adapters
- serial and parallels ports
- sound cards
- USB controllers

The main operations you can perform on these devices are:

- adding a new device to the virtual machine
- configuring the device properties
- removing a device from the virtual machine

Adding a New Device

This section provides information on adding new devices to your virtual machines. You can add new virtual devices to your virtual machine using the `pctl set` command. The options responsible for adding particular devices are listed in the following table:

Option Name	Description
<code>hdd</code>	Adds a new hard disk drive to the virtual machine. You can either connect an existing image to the virtual machine or create a new one.
<code>cdrom</code>	Adds a new CD/DVD-ROM drive to the virtual machine.
<code>net</code>	Adds a new network adapter to the virtual machine.
<code>fdd</code>	Adds a new floppy disk drive to the virtual machine.
<code>serial</code>	Adds a new serial port to the virtual machine.
<code>parallel</code>	Adds a new parallel port to the virtual machine.
<code>sound</code>	Adds a new sound device to the virtual machine.
<code>usb</code>	Adds a new USB controller to the virtual machine.

For example, you can execute the following command to add a new virtual disk to the `MyVM` virtual machine:

```
# pctl set MyVM --device-add hdd
Creating hdd1 (+) scsi:0 image='/var/parallels/MyVM.pvm/harddisk1.hdd
Create the expanding disk, 32768Mb...
The VM has been successfully configured.
```

This command creates a new virtual disk with the following default parameters:

- name: `hdd1`
- disk type: SCSI
- image file name and location: `/var/parallels/MyVM.pvm/harddisk1.hdd`
- disk format: expanding
- disk capacity: 32768 MB

You can redefine some of these parameters by specifying specific options during the command execution. For example, to create an IDE virtual disk that will have the capacity of 64 GB, you can run this command:

```
# pctl set MyVM --device-add hdd --size 64000 --iface ide
Creating hdd1 (+) ide:1 image='/var/parallels/MyVM.pvm/harddisk1.hdd
Create the expanding disk, 64000Mb...
The VM has been successfully configured.
```

The virtual disk has been added to your virtual machine. However, before starting to use it, you must initialize the disk. Refer to the next subsection for information on how you can do it.

When managing devices, keep in mind the following:

- Detailed information on all options that can be passed to `pctl set` when creating a new virtual machine device is provided in the *Parallels Command Line Reference Guide*.
- You can connect up to 4 IDE devices and up to 15 SCSI devices (virtual disks or CD/DVD-ROM drives) to a virtual machine.

- If you want to use an existing image file as a virtual CD/DVD-ROM drive, keep in mind that Parallels Server for Mac Bare Metal Edition supports `.iso`, `.cue`, `.ccd` and `.dmg` (non-compressed and non-encrypted) image files.
- A virtual machine can have only one floppy disk drive.
- A virtual machine can have up to 16 virtual network adapters.
- A virtual machine can have up to four serial ports.
- A virtual machine can have up to three parallel ports.
- Any virtual machine can have only one sound device.
- A virtual machine can have only one USB controller.

Initializing the Newly Added Disk

After you added a new blank virtual hard disk to the virtual machine configuration, it will be invisible to the operating system installed inside the virtual machine until the moment you initialize it.

Initializing the New Virtual Hard Disk in Windows

To initialize the new virtual hard disk in a Windows guest OS, you will need the Disk Management utility available through:

- In Windows Vista: Start > Control Panel > System and Maintenance > Administrative Tools > Create and Format Hard Disk Partitions > Disk Management.
- In Windows XP: Start > Control Panel > Administrative Tools > Computer Management > Storage > Disk Management.

When you open the Disk Management utility, it automatically detects that a new hard disk was added to the configuration and launches **Initialize and Convert Disk Wizard**:

- 1 In the introduction window, click **Next**.
- 2 In the **Select Disks to Initialize** window, select the newly added disk and click **Next**.
- 3 In the **Select Disks to Convert** window, select the newly added disk and click **Finish**.

The added disk will appear as a new disk in the Disk Management utility window, but its memory space will be unallocated. To allocate the disk memory, right-click this disk name in the Disk Management utility window and select **New Simple Volume** in Windows Vista or **New Volume** in Windows XP. The **New Simple Volume Wizard**/**New Volume Wizard** window will appear. Follow the steps of the wizard and create a new volume in the newly added disk.

After that your disk will become visible in **Computer/My Computer** and you will be able to use it as a data disk inside your virtual machine.

Initializing the New Virtual Hard Disk in Linux

Initializing the new virtual hard disk in a Linux guest OS comprises two steps: allocating the virtual hard disk space and mounting this disk in the guest OS.

To allocate the space, you will need to create a new partition on this virtual hard disk, using the `fdisk` utility.

Note: To use the `fdisk` utility, you need the `root` privileges.

- 1 Launch Terminal.
- 2 To list the IDE disk devices present in your virtual machine configuration, enter:

```
fdisk /dev/hd*
```

Note: If you added a SCSI disk to the virtual machine configuration, use the `fdisk /dev/sd*` command instead.

- 3 By default, the second virtual hard disk appears as `/dev/hdc` in your Linux virtual machine. To work with this device, enter:

```
fdisk /dev/hdc
```

Note: If this is a SCSI disk, use the `fdisk /dev/sdc` command instead.

4 To get extensive information about the disk, enter:

```
p
```

5 To create a new partition, enter:

```
n
```

6 To create the primary partition, enter:

```
p
```

7 Specify the partition number. By default, it is 1.

8 Specify the first cylinder. If you want to create a single partition on this hard disk, use the default value.

9 Specify the last cylinder. If you want to create a single partition on this hard disk, use the default value.

10 To create a partition with the specified settings, enter:

```
w
```

When you allocated the space on the newly added virtual hard disk, you should format it by entering the following command in the terminal:

```
mkfs -t <FileSystem> /dev/hdc1
```

Note: *<FileSystem>* stands for the file system you want to use on this disk. It is recommended to use `ext3` or `ext2`.

When the added virtual hard disk is formatted, you can mount it in the guest OS.

1 To create a mount point for the new virtual hard disk, enter:

```
mkdir /mnt/hdc1
```

Note: You can specify a different mount point.

2 To mount the new virtual hard disk to the specified mount point, enter:

```
mount /dev/hdc1 /mnt/hdc1
```

When you mounted the virtual hard disk, you can use its space in your virtual machine.

Configuring Virtual Devices

In Parallels Server for Mac Bare Metal Edition, you can use the `--device-set` option of the `pctl set` command to configure the parameters of an existing virtual device. As a rule, the process of configuring the device properties includes two steps:

- 1 Finding out the name of the device you want to configure.
- 2 Running the `pctl set` command to configure the necessary device properties.

Finding Out the Device Name

To configure a virtual device, you need to specify its name when running the `pctl set` command. If you do not know the device name, you can use the `pctl list` command to learn it. For example, to obtain the list of virtual devices in the `MyVM` virtual machine, run this command:

```
# pctl list --info MyVM
...
Hardware:
  cpu 2 VT-x accl=high mode=32
  memory 256Mb
  video 46Mb
  fdd0 (+) real='/dev/fd0' state=disconnected
  hdd0 (+) ide:0 image='/var/parallels/MyVM.pvm/harddisk.hdd' 27000Mb
  hdd1 (+) scsi:0 image='/var/parallels/MyVM.pvm/harddisk1.hdd' 32768Mb
  cdrom0 (+) ide:1 real='Default CD/DVD-ROM'
  parallelo (+) real='/dev/lp0'
  usb (+)
  net0 (+) type=bridged iface='eth1' mac=001C4201CED0
...
```

All virtual devices currently available to the virtual machine are listed under `Hardware`. In our case the `MyVM` virtual machine has the following devices: 2 CPUs, main memory, video memory, a floppy disk drive, 2 hard disk drives, a CD/DVD-ROM drive, a parallel port, a USB controller, and a network card.

Configuring a Virtual Device

Once you know the virtual device name, you can configure its properties. For example, you can execute the following command to configure the current type of the virtual disk `hdd1` in the `MyVM` virtual machine from SCSI to IDE:

```
# pctl set MyVM --device-set hdd1 --iface ide
The VM has been successfully configured.
```

To check that the virtual disk type has been successfully changed, use the `pctl list --info` command:

```
# pctl list --info MyVM
...
  hdd0 (+) ide:0 image='/var/parallels/MyVM.pvm/harddisk.hdd' 27000Mb
  hdd1 (+) ide:2 image='/var/parallels/MyVM.pvm/harddisk1.hdd' 32768Mb
...
```

Deleting a Device

You can delete a virtual device that you do not need any more in your virtual machine using the `--device-del` option of the `pctl set` command. The options responsible for removing particular devices are listed in the following table:

Option Name	Description
<code>hdd</code>	Deletes the specified hard disk drive from the virtual machine.
<code>cdrom</code>	Deletes the specified CD/DVD-ROM drive from the virtual machine.
<code>net</code>	Deletes the specified network adapter from the virtual machine.
<code>fdd</code>	Deletes the floppy disk drive from the virtual machine.
<code>serial</code>	Deletes the specified serial port from the virtual machine.
<code>parallel</code>	Deletes the specified parallel port from the virtual machine.
<code>sound</code>	Deletes the sound device from the virtual machine.
<code>usb</code>	Deletes the USB controller from the virtual machine.

As a rule deleting a virtual device involves performing two operations:

- 1 Finding out the name of the device to be deleted.
- 2 Deleting the device from the virtual machine.

Finding Out the Device Name

To remove a virtual device, you need to specify its name when running the `pctl set` command. If you do not know the device name, you can use the `pctl list` command to learn it. For example, to obtain the list of virtual devices in the `MyVM` virtual machine, run this command:

```
# pctl list --info MyVM
...
Hardware:
  cpu 2 VT-x accl=high mode=32
  memory 256Mb
  video 46Mb
  fdd0 (+) real='/dev/fd0' state=disconnected
  hdd0 (+) ide:0 image='/var/parallels/MyVM.pvm/harddisk.hdd' 27Mb
  hdd1 (+) scsi:0 image='/var/parallels/MyVM.pvm/harddisk1.hdd' 32768Mb
  cdrom0 (+) ide:1 real='Default CD/DVD-ROM'
  parallel0 (+) real='/dev/lp0'
  usb (+)
  net0 (+) type=bridged iface='eth1' mac=001C4201CED0
...
```

All virtual devices currently available to the virtual machine are listed under `Hardware`. In our case the `MyVM` virtual machine has the following devices: 2 CPUs, main memory, video memory, a floppy disk drive, 2 hard disk drives, a CD/DVD-ROM drive, a parallel port, a USB controller, and a network card.

Deleting a Virtual Device

Once you know the virtual device name, you can remove it from your virtual machine. For example, you can execute the following command to remove the virtual disk `hdd1` from the `MyVM` virtual machine:

```
# pct1 set MyVM --device-del hdd1
Remove the hdd1 device.
The VM has been successfully configured.
```

When deleting virtual machine devices, keep in mind the following:

- If you do not want to permanently delete a virtual device, you can temporarily disconnect it from the virtual machine using the `--disable` option.
- Detailed information on all options that can be used with `pctl set` when deleting a device is given in the `Parallels Command Line Reference Guide`.

Managing Snapshots

In `Parallels Server for Mac Bare Metal Edition`, you can save the current state of a virtual machine by creating a snapshot. You can then continue working in your virtual machine and return to the saved state any time you wish. For example, you can make use of snapshots in the following cases:

- You are going to configure an application with a lot of settings. In this case, you may first wish to play with settings before applying them to your application. So, you create a snapshot before starting to experiment with the application settings.
- You are involved in a large development project. In this case, you may wish to mark milestones in the development process by creating a snapshot after each milestone. If anything goes wrong, you can easily revert to the previous milestone and start the development anew.

In `Parallels Server for Mac Bare Metal Edition`, you can manage snapshots as follows:

- create a new snapshot of a virtual machine
- list the existing snapshots of a particular virtual machine
- revert to a snapshot
- remove a snapshot

All these operations are described in the following subsections in detail.

Creating a Snapshot

To create a snapshot of a virtual machine in Parallels Server for Mac Bare Metal Edition, you can use the `pctl snapshot` command. For example, you can execute the following command to create a snapshot of the `MyVM` virtual machine:

```
# pctl snapshot MyVM
Creating the snapshot...
The snapshot with ID {12w32198-3e30-936e-a0bbc104bd20} has been successfully
created.
```

A newly created snapshot is saved to the `/vz/VM_Name.pvm/Snapshots/Snapshot_ID.pvs` file where `VM_Name` denotes the name of the corresponding virtual machine and `Snapshot_ID` is a random ID assigned to the snapshot. In the command above, the snapshot is assigned the ID of `{12w32198-3e30-936e-a0bbc104bd20}` and saved to the `/vz/MyVM/Snapshots/{12w32198-3e30-936e-a0bbc104bd20}.pvs` file.

```
# ls /vz/MyVM.pvm/Snapshots/
{063615fa-f2a0-4c14-92d4-4c935df15840}.pvc
```

The ID assigned to the snapshot can be used to manage this snapshot (e.g. get detailed information on the snapshot or delete it).

When creating a snapshot, you can also set a name for it and provide its description:

```
# pctl snapshot MyVM -n Clean_System -d "This snapshot was created right after
installing the Windows XP operating system"
Creating the snapshot...
The snapshot with ID {0i8798uy-1eo0-786d-nn9ic106b9ik} has been successfully
created.
```

You can then view the set name and description in the `/vz/MyVM/Snapshots.xml` file or in Parallels Management Console.

When working with snapshots, keep in mind the following:

- If a virtual machine name contains spaces, use quotation marks to specify the name in `pctl` commands (e.g. "Windows XP").
- Before creating a snapshot, it is recommended that you complete all operations of installing, downloading, or writing to external devices. You should also complete or cancel any transactions performed via the virtual machine in external databases.

Creating Branches

The branches are created when you do the following:

- 1 Create several sequential snapshots.
- 2 Revert to an intermediate snapshot.
- 3 Make some changes to the virtual machine.
- 4 Save the virtual machine state by creating a new snapshot.

In this case, the newly created snapshot will start a new branch using the intermediate snapshot from **Step 2** as the baseline.

Listing Snapshots

To list all snapshots of a particular virtual machine, use the `pctl snapshot-list` command:

```
# pctl snapshot-list MyVM
PARENT_SNAPSHOT_ID          SNAPSHOT_ID
                             {989f3415-3e30-4494-936e-a0bbc104bd20}
{989f3415-3e30-4494-936e-a0bbc104bd20} *{063615fa-f2a0-4c14-92d4-4c935df15840}
```

This command shows that currently two snapshots exist for the MyVM virtual machine. The snapshot with ID `{063615fa-f2a0-4c14-92d4-4c935df15840}` is based on the snapshot with ID `{989f3415-3e30-4494-936e-a0bbc104bd20}`, i.e. the latter acts as the parent for the snapshot with ID `{063615fa-f2a0-4c14-92d4-4c935df15840}`. The `*` sign before `{063615fa-f2a0-4c14-92d4-4c935df15840}` denotes that this is the current snapshot for the given virtual machine.

You can also view the relationship between snapshots by specifying the `-t` option:

```
# pctl snapshot-list MyVM -t
_{989f3415-3e30-4494-936e-a0bbc104bd20}_{063615fa-f2a0-4c14-92d4-4c935df15840}
\_{712305b0-3742-4ecc-9ef1-9f1e345d0ab8}
```

The command output shows you that currently 2 branches exist for the MyVM virtual machine. The snapshot with ID `{989f3415-3e30-4494-936e-a0bbc104bd20}` is the baseline used as a starting point for these branches.

You can get detailed information on a particular snapshot using the `-i` option and specifying the snapshot ID:

```
# pctl snapshot-list MyVM -i {063615fa-f2a0-4c14-92d4-4c935df15840}
ID: {063615fa-f2a0-4c14-92d4-4c935df15840}
Name: Clean_System
Date: 2009-07-22 22:39:06
Current: yes
State: power_off
Description: <![CDATA[This snapshot was created right after installing Windows XP operating system]]>
```

The `pctl snapshot-list` command displays the following information about snapshots:

Field	Description
ID	The ID assigned to the snapshot.
Name	The name assigned to the snapshot.
Date	The date and time when the snapshot was created.
Current	Denotes whether this is the current snapshot of the virtual machine.
State	The state the virtual machine was in at the time you took the snapshot.
Description	The description set for the snapshot.

Reverting to a Snapshot

You can use the `pctl snapshot-switch` command to revert to a snapshot. When you revert to a snapshot, the current state of the virtual machine is discarded, and all changes made to the system since the previous snapshot are lost. So, before returning to a specific snapshot, you may first wish to save these states by creating a new snapshot. Refer to the [Creating a Snapshot](#) subsection (p. 54) for information on how you can do it.

The `pctl snapshot-switch` command requires the virtual machine name and the snapshot ID to be specified as arguments:

```
pctl snapshot-switch "Windows XP" --id {cedbc4eb-dee7-42e2-9674-89d1d7331a2d}
Switch to the snapshot...
The VM has been successfully switched.
```

This command restores the snapshot with ID `{cedbc4eb-dee7-42e2-9674-89d1d7331a2d}` for the `Windows XP` virtual machine.

Deleting a Snapshot

In Parallels Server for Mac Bare Metal Edition, you can use the `pctl snapshot-delete` command to delete those snapshots that you do not need any more. Assuming that you want to delete the snapshot with ID `{903c12ea-f6e6-437a-a2f0-a1d02eed4f7e}` for the `MyVM` virtual machine, you can run this command:

```
# pctl snapshot-delete MyVM --id {903c12ea-f6e6-437a-a2f0-a1d02eed4f7e}
Deleting the snapshot...
The snapshot has been successfully deleted.
```

When you delete a parent snapshot, its children are not deleted, and the information the parent snapshot contains is merged into them.

For example, the following session demonstrates the process of deleting the snapshot with ID `{903c12ea-f6e6-437a-a2f0-a1d02eed4f7e}` acting as a parent for another snapshot:

```
# pctl snapshot-list MyVM
PARENT_SNAPSHOT_ID          SNAPSHOT_ID
{989f3415-3e30-4494-936e-a0bbc104bd20} {989f3415-3e30-4494-936e-a0bbc104bd20}
{063615fa-f2a0-4c14-92d4-4c935df15840} {063615fa-f2a0-4c14-92d4-4c935df15840}
{063615fa-f2a0-4c14-92d4-4c935df15840} *{58c9941e-f232-4273-892a-82e836536889}
# pctl snapshot-delete MyVM --id {903c12ea-f6e6-437a-a2f0-a1d02eed4f7e}
Deleting the snapshot...
The snapshot has been successfully deleted.
# pctl snapshot-list MyVM
PARENT_SNAPSHOT_ID          SNAPSHOT_ID
{063615fa-f2a0-4c14-92d4-4c935df15840} {063615fa-f2a0-4c14-92d4-4c935df15840}
{063615fa-f2a0-4c14-92d4-4c935df15840} *{58c9941e-f232-4273-892a-82e836536889}
```

Making Screenshots

In Parallels Server for Mac Bare Metal Edition, you can use the `pctl capture` command to capture an image (or screenshot) of your virtual machine screen. You can take screenshots of running virtual machines only. The session below demonstrates how to take a screenshot of the `My_VM` virtual machine screen and save it to the `/usr/screenshots/image1.png` file:

1 Make sure that the virtual machine is running:

```
# pctl list
STATUS  IP_ADDR      NAME
running 10.10.10.101 101
running 10.10.10.201 My_VM
```

2 Take the virtual machine screenshot:

```
# pctl capture My_VM --file /usr/screenshots/image1.png
Capture the VM screen...
The VM screen has been saved to /usr/screenshots/image1.png
```

3 Check that the `image1.png` file has been successfully created:

```
# ls /usr/screenshots/
image1.png
```

Managing Virtual Machine Resources

Parallels Server for Mac Bare Metal Edition allows you to manage the following resources of your virtual machines:

- main memory
- number of CPUs
- video memory

The procedure of managing these resources is described below in this section.

Configuring Main Memory

To configure the amount of memory that will be available to the virtual machine, use the `--memsize` option of the `pctl set` command. The following session shows how to change the amount of memory for the `MyVM` virtual machine from 512 MB to 756 MB and to check that the new value has been successfully set:

```
# pctl list -i MyVM | grep memory
memory 512Mb
# pctl set MyVM --memsize 756
Set the memsize parameter to 756Mb
The VM has been successfully configured.
# pctl list -i MyVM | grep memory
memory 756Mb
```

You can configure the memory size for both running and stopped virtual machines.

Configuring the Number of CPUs

If the Parallels server has more than one physical processor installed, you can control the number of CPUs which will be used to handle the processes running inside your virtual machines. By default, a virtual machine is allowed to consume the CPU time of one processor only. However, you can modify the number of physical CPUs which will be simultaneously available to a virtual machine using the `--cpus` option of the `pctl set` command. For example, if your server has 4 physical processors installed, you can set the processes inside the `MyVM` virtual machine to be run on 2 CPUs by issuing the following command:

```
# pctl set MyVM --cpus 2
Set cpus(2): 2
The VM has been successfully configured.
```

Note: The maximum allowable number of virtual CPUs depends on the number of physical CPU cores available on the Parallels server. For example, if you have a Core 2 Duo physical processor, the maximum allowable number of virtual CPUs will be 2.

You can check if the number of CPUs has been successfully changed by running this command:

```
# pctl list -i MyVM | grep cpu
cpu 2 VT-x accl=high mode=32
```

Configuring Video Memory

To set the amount of video memory to be available to the virtual machine's video card, use the `--videosize` option of the `pctl set` command. Assuming that the current video memory size of the `MyVM` virtual machine is set to 32 MB, you can increase it to 64 MB by running the following command:

```
# pctl set MyVM --videosize 64
Set the --videosize parameter to 64Mb.
The VM has been successfully configured.
```

To check that the new value has been successfully set, use this command:

```
# pctl list -i MyVM | grep video
video 64Mb
```

CHAPTER 4

Managing Parallels Server for Mac Bare Metal Edition Network

The given chapter familiarizes you with the Parallels Server for Mac Bare Metal Edition network structure, enumerates Parallels networking components, and explains how to manage these components in your working environments. In particular, it provides the following information:

- How you can manage network adapters on the Parallels server.
- What Virtual Networks are and how you can manage them on the Parallels server.
- How to create virtual network adapters inside your virtual machines and configure their parameters.
- How to connect virtual machines to different networks.

In This Chapter

Managing Network Adapters on the Parallels Server	60
Managing Virtual Networks.....	64
Managing Adapters in Virtual Machines	68

Managing Network Adapters on the Parallels Server

Network adapters installed on the Parallels server are used to provide virtual machines with access to each other and to external networks. During the installation, Parallels Server for Mac Bare Metal Edition registers all physical and VLAN network adapters available on the server. In addition to that, it creates a number of virtual network adapters on the server. Once Parallels Server for Mac Bare Metal Edition has been successfully installed, you can perform the following operations on network adapters:

- List the adapters currently available on the server.
- Create new VLAN adapters on the server.
- Connect adapters to Virtual Networks on the server.

Note: For more information on Virtual Networks, refer to *Managing Virtual Networks* (p. 64).

These operations are described in the following subsections in detail.

Listing Adapters

You can view the physical, virtual, and VLAN network adapters existing on your Parallels server using the `vznetcfg` utility. For example, you can execute the following command to list the available adapters:

```
# vznetcfg if list
Name      Type      Network ID  Addresses
eth0      nic       Bridged     10.30.18.41/16,dhcp
br2       bridge   Bridged
br1       bridge   Host-Only
br0       bridge   Shared
vnic1     vnic     Host-Only   10.37.131.2/24
vnic0     vnic     Shared      10.37.130.2/24
```

The information on adapters is presented in the table having the following columns:

Column Name	Description
Name	The adapter name.
Type	The type of the network adapter. It can be one of the following: <ul style="list-style-type: none"> ▪ <code>nic</code> denotes a physical adapter installed on the Parallels server. ▪ <code>vlan</code> stands for a VLAN adapter available on the Parallels server. ▪ <code>vnic</code> denotes a virtual network adapter available on the Parallels server. By default, 2 virtual adapters are created during the Parallels Server for Mac Bare Metal Edition installation: <code>vnic0</code> and <code>vnic1</code>. Besides, a new virtual adapter is automatically created on the server when you create a new Virtual Network. ▪ <code>bridge</code> is a virtual bridge automatically created for each Virtual Network on the Parallels server.
Network ID	The ID of the Virtual Network where the network adapter is connected. Detailed information on Virtual Networks is provided in <i>Managing Virtual Networks</i> (p. 64).
Addresses	The IP address and subnet mask assigned to the network adapter. <code>dhcp</code> denotes that the adapter gets its network parameters from a DHCP server.

Creating VLAN Adapter

Parallels Server for Mac Bare Metal Edition allows you to create new VLAN adapters on the Parallels server. You can use these adapters later on to connect your virtual machines to any of the available Virtual Networks (for more information on Virtual Networks, turn to [Managing Virtual Networks](#) (p. 64). VLAN adapters can be made using the `vznetcfg vlan add` command. To create a new VLAN adapter, you should specify the VLAN ID - an arbitrary integer number which will uniquely identify the virtual LAN among other VLANs on the server - and the physical network adapter on the server to which the VLAN is to be bound. For example, you can execute the following command to make a new VLAN adapter, associate it with a VLAN having the ID of 5 (i.e. with VLAN 5), and attach the VLAN adapter to the `eth0` physical adapter on the server:

```
# vznetcfg vlan add eth0 5
```

To check that the VLAN adapter has been successfully created, execute the following command:

```
# vznetcfg if list
Name      Type      Network ID  Addresses
eth0      nic       192.168.0.150/22,dhcp
eth0.5    vlan
```

VLAN adapters can be easily identified by the `vlan` designation shown in the `Type` column of the command output. As you can see, only one VLAN adapter currently exists on the server. It is assigned the name of `eth0.5`. This name is generated automatically on the basis of the specified VLAN ID and the name of the physical adapter to which the VLAN adapter is tied.

At any time, you can delete the `eth0.5` VLAN adapter and thus destroy VLAN 5 by issuing the following command:

```
# vznetcfg vlan del eth0.5
# vznetcfg if list
Name      Type      Network ID  Addresses
eth0      nic       192.168.0.150/22,dhcp
```

Managing Virtual Networks

A Virtual Network acts as a binding interface between a virtual network adapter inside a virtual machine and the corresponding network adapter on the Parallels server, which allows you to include your virtual machines in different networks. In Parallels Server for Mac Bare Metal Edition, you can manage Virtual Networks as follows:

- Create a new Virtual Network and remove an existing one.
- Configure the parameters of an existing Virtual Network.
- List the existing Virtual Networks.
- Delete a Virtual Network that you do not need any more.

These operations are described in the following subsections in detail.

Creating a Virtual Network

Virtual Networks serve as binding interfaces between the virtual network adapters inside virtual machines and the physical, VLAN, and virtual network adapters on the Parallels server. Using Virtual Networks, you can connect your virtual machines to different networks.

By default, Parallels Server for Mac Bare Metal Edition creates the following Virtual Networks:

- *Bridged.* This Virtual Network is connected to one of the physical adapters on the Parallels server (as a rule, `eth0`) and provides virtual machines included in this Virtual Network with access to the network behind this physical adapter.
- *Shared.* This Virtual Network is connected to the `vnic0` virtual adapter on the Parallels server and allows virtual machines included in this Virtual Network to use the current network connections of your Parallels server.
- *Host-only.* This Virtual Network is connected to the `vnic1` virtual adapter on the Parallels server and allows a virtual machine included in this Virtual Network to access only the Parallels server and the other virtual machines on this network.

You can also create your own Virtual Networks using the `prlsrvctl` or `vznetcfg` utility. For example, to make a new Virtual Network with the name of `vznetwork1`, you can issue one of the following commands:

```
# vznetcfg net new vznetwork1
```

or

```
# prlsrvctl net add vznetwork1
```

By default, both commands create host-only Virtual Networks. However, you can change their types using the `prlsrvctl` utility (see [Configuring Virtual Network Parameters](#) (p. 66) for details).

In the current version of Parallels Server for Mac Bare Metal Edition, you can create

- 1 shared Virtual Network (it is automatically created on the Parallels server during the Parallels Server for Mac Bare Metal Edition installation).
- 5 host-only Virtual Networks (1 host-only Virtual Network is automatically created on the Parallels server during the Parallels Server for Mac Bare Metal Edition installation).
- One or more bridged Virtual Networks. The number Virtual Networks depends on the number of physical and VLAN adapters available on the Parallels server. One Virtual Network can be connected to only one physical or VLAN adapter.

Viewing Bridges

Each Virtual Network is associated with some bridge which is automatically made on the Parallels server during the Virtual Network creation and serves as the basis for the Virtual Network functioning. To find out what bridge is associated with what Virtual Network, you can run the following command:

```
# vznetcfg if list
Name      Type      Network ID  Addresses
eth0      nic       Bridged     10.30.18.41/16,dhcp
br3       bridge   vznetwork1
br2       bridge   Bridged
br1       bridge   Host-Only
br0       bridge   Shared
vnic1     vnic     Host-Only   10.37.131.2/24
```

vnic0	vnic	Shared	10.37.130.2/24
vnic2	vnic	vznetwork1	10.37.132.2/24

The bridges existing on the Parallels server are listed in the Name column and can be easily identified by the `br` prefix. For example, you can see that the `br3` bridge is currently associated with the `vznetwork1` Virtual Network.

Configuring Virtual Network Parameters

Parallels Server for Mac Bare Metal Edition allows you to configure the following parameters for a Virtual Network:

- the name assigned to the Virtual Network
- the networking mode in which the Virtual Network is operating
- the description of the Virtual Network

All these operations can be performed using the `prlsrvctl` utility. Let us assume that you want to configure the `vznetwork1` Virtual Network. This Virtual Network is currently configured as a host-only network, attached to the `vnic2` adapter, and has the following description: This is a host-only Virtual Network. To change these parameters, you can execute the following command:

```
# prlsrvctl net set vznetwork1 -n psbm_network1 -t bridged --ifname eth1 -d  
"This is a bridged Virtual Network"
```

This command sets the following parameters for the `vznetwork1` Virtual Network:

- Changes the Virtual Network name to `psbm_network1`.
- Changes the Virtual Network type to `bridged`. To do this, `prlsrvctl` detaches the Virtual Network from the `vnic2` adapter and connects it to the `eth1` physical adapter on the Parallels server.
- Changes the Virtual Network description to the following: This is a shared Virtual Network. You can view this description in Parallels Management Console.

For more information on the `prlsrvctl` utility, refer to the *Parallels Command Line Reference Guide*.

Listing Virtual Networks

Sometimes, you may wish to list the Virtual Networks existing on the Parallels server. To do this, you can use either the `vznetcfg` or `prlsrvctl` utility.

Listing Virtual Networks With `vznetcfg`

To list the Virtual Networks on your server using the `vznetcfg` utility, execute the following command:

```
# vznetcfg net list
Network ID      Status      Master Interface  Slave Interfaces
Shared         active     vnic0
Host-Only      active     vnic1
Bridged        active     eth0
vznetwork1     active     vnic2
```

In the example above, 4 Virtual Networks - `vznetwork1` and 3 default Virtual Networks - exist on the Parallels server. The information on these Virtual Networks is presented in the table having the following columns:

Column Name	Description
Network ID	The ID assigned to the Virtual Network.
Status	Indicates the status of the Virtual Network. It can be one of the following: <ul style="list-style-type: none"> active: the Virtual Network is up and running. configured: the information on the Virtual Network is present in the <code>/etc/vz/vznet.conf</code> file on the server, but the bridge to which the Virtual Network is bound is down or absent from the server. <p>Note: Detailed information on the <code>vznet.conf</code> file is given in the <i>Parallels Command Line Reference Guide</i>.</p>
Master Interface	Displays the adapter on the server connected to the Virtual Network, if any.
Slave Interfaces	Lists the adapters in virtual machines joined to the Virtual Network, if any.

Listing Virtual Networks With `prlsrvctl`

You can also use the `prlsrvctl` utility to list the Virtual Networks existing on your server. To do this, run the following command:

```
# prlsrvctl net list
Network ID      Type      Bound To
Shared         shared    vnic0
Host-Only      host-only vnic1
Bridged        bridged   eth0
vznetwork1     host-only vnic2
```

This utility displays the following information on Virtual Networks:

Column Name	Description
-------------	-------------

Network ID	The name assigned to the Virtual Network.
Type	The networking mode set for the Virtual Network.
Bound To	The adapter on the Parallels server connected to the Virtual Networks, if any.

Deleting a Virtual Network

At any time, you can remove a Virtual Network that you do not need any more from the physical server. To do this, you can use both the `vznetcfg` and `prlsrvctl` utilities. For example, you can delete the `vznetwork1` Virtual Network by running one of the following commands:

```
# vznetcfg net del vznetwork1
```

or

```
# prlsrvctl net del vznetwork1
```

To check that `vznetwork1` has been successfully removed, execute one of these commands:

```
# vznetcfg net list
```

Network ID	Status	Master Interface	Slave Interfaces
Shared	active	vnic0	
Host-Only	active	vnic1	
Bridged	active	eth0	

or

```
# prlsrvctl net list
```

Network ID	Type	Bound To
Shared	shared	vnic0
Host-Only	host-only	vnic1
Bridged	bridged	eth0

Note: Detailed information on the `vznetcfg` and `prlsrvctl` utilities is provided in the *Parallels Command Line Reference Guide* and their manual pages.

Managing Adapters in Virtual Machines

This section provides information on how you can manage virtual network adapters in your virtual machine. You will learn to do the following:

- Create new virtual network adapters and delete existing ones.
- Configure the parameters of an existing virtual network adapter (e.g. assign an IP address to it).
- Join virtual network adapters to Virtual Networks.

All these operations are described in the following subsections in detail.

Creating and Deleting Virtual Adapters

A virtual machine can have up to 16 virtual network adapters. Each adapter can be connected to a different network. Let us assume that you wish to create a new virtual adapter for the MyVM virtual machine. To do this, you can execute the following command :

```
# pct1 set MyVM --device-add net
Creating net1 (+) type=shared iface='default' mac=XXXXXXXXXXXX
The VM has been successfully configured.
```

To check that the network adapter (net1) has been successfully added to the virtual machine, run this command:

```
# pct1 list --info MyVM
ID: {f3b3d134-f512-324b-b0b1-dbd642f5220b}
Name: Windows XP
...
net0 (+) type=shared iface='default' mac=001C42566BCF
net1 (+) type=shared iface='default' mac=001C42AF3D69
```

At any time, you can remove the newly created network adapter (net1) by executing the following command:

```
# pct1 set MyVM --device-del net1
Remove the net1 device.
The VM has been successfully configured.
```

For the full of options that can be used when creating a new virtual network adapter, refer to the *Parallels Command Line Reference Guide*.

Configuring Virtual Adapter Parameters

Parallels Server for Mac Bare Metal Edition allows you to configure the following parameters of virtual machine adapters:

Configuring the MAC Address

If you need for some reason to regenerate the current MAC address of a network adapter, you can use the following command:

```
# pct1 set MyVM --device-set net1 --mac 00:1C:42:2D:74:00
Creating net1 (+) network=Bridged mac=001C422D7400
The VM has been successfully configured.
```

This command sets the MAC address of 00:1C:42:2D:74:00 MAC address for the net1 adapter in the MyVM virtual machine. If do not know what MAC address to assign to your virtual adapter, you can make `pctl set` automatically generate a new MAC address. To do this, run the following command:

```
# pct1 set MyVM --device-set net1 --mac auto
Creating net1 (+) network=Bridged mac=001C42C84F3E
The VM has been successfully configured.
```

Configuring the IP Parameters

As any other standalone server, each virtual machine must have a number of TCP/IP settings configured in the proper way to successfully operate on the network. These settings include:

- an IP address for each virtual network adapter inside the virtual machine
- the default gateways to be used by the virtual machine
- the default DNS servers to be used by the virtual machine

Usually, you define all these settings during the virtual machine creation. However, if you have not yet set any of the settings or want to modify any of them, you can use the `pctl set` command. For example, you can execute the following command to assign the IP address of 192.129.129.20 to the net1 adapter in the MyVM virtual machine and set for it the default gateway with the IP address 192.129.129.1 and the DNS server with the IP address of 192.192.192.10:

```
# pct1 set MyVM --device-set net1 --ipadd 192.129.129.20 --gw 192.129.129.1 --
nameserver 192.192.192.10
```

Along with a static assignment of network parameters to a virtual adapter, you can make the adapter receive its TCP/IP settings automatically using the Dynamic Host Configuration Protocol (DHCP). For example, you can run this command to make the net1 adapter in the MyVM virtual machine get its IP settings through DHCP:

```
# pct1 set MyVM --device-set net1 --dhcp yes
Creating net1 (+) network=Bridged mac=001C42C84F3E
Enable automatic reconfiguration for this network adapter.
The VM has been successfully configured.
```

Detailed information on all options which can be used with the `pctl set` command to manage virtual machine adapter parameters is given in the *Parallels Command Line Reference Guide* and the `pctl` manual pages.

Connecting Virtual Machines to Virtual Networks

In Parallels Server for Mac Bare Metal Edition, you can connect your virtual machines to Virtual Networks of the following types:

- *Bridged networks.* This type of Virtual Networks allows the virtual machine to use one of the physical server's network adapters, which makes it appear as a separate computer on the network the corresponding adapter belongs to.
- *Shared networks.* This type of Virtual Networks allows the virtual machine to use the current network connections of your Parallels server.
- *Host-only networks.* This type of Virtual Networks allows the virtual machine to access only the Parallels server and the virtual machines joined to this network.

To connect your virtual machines to any of these networks, use the `pctl set` command. For example, the following session shows you how to connect the `net0` adapter of the MyVM virtual machine to the Bridged Virtual Network (this is one of the default Virtual Networks created on the Parallels server during the Parallels Server for Mac Bare Metal Edition installation).

Before connecting the MyVM virtual machine to the Bridged Virtual Network, you may wish to check the network adapter associated with this Virtual Network. You can do it, for example, using the following command:

```
# prlsrvctl net list
Network ID      Type      Bound To
Shared          shared    vnic0
Host-Only       host-only vnic1
Bridged         bridged   eth0
vznetwork1     host-only vnic2
```

From the command output, you can see that the Bridged Virtual Network is attached to the `eth0` physical adapter on the Parallels server. It means that, after connecting the MyVM virtual machine to the Bridged Virtual Network, the virtual machine will be able to access all the computers on the network where the `eth0` adapter is connected.

Now you can run the following command to join the `net0` adapter of the MyVM virtual machine to the Bridged Virtual Network:

```
# pctl set MyVM --device-set net0 --network Bridged
Creating net0 (+) network=Bridged mac=001C422D7493
The VM has been successfully configured.
```

CHAPTER 5

Managing Licenses

The given chapter provides information on managing Parallels Server for Mac Bare Metal Edition licenses. In particular, you will know how to view the current license status, to install a new license on your server or to update an existing one, to transfer the license from one server to another, etc.

In This Chapter

Installing the License	73
Updating the Current License	74
Transferring the License to Another Server	75
Viewing the Current License	76

Installing the License

Depending on the way you have obtained your Parallels Server for Mac Bare Metal Edition license, it can be installed on the Parallels server as follows:

- If you have obtained the license in the form of a product key, you can install it on the server using the `-p` option of the `vzlicload` command. For example, you can execute the following command to install the `XXXXXX-XXXXXX-XXXXXX-XXXXXX-XXXXXX` product key:

```
# vzlicload -p 5BVMF2-560MM0-D28DQA-B59NTE-10H4HG
Processing product key "XXXXXX-XXXXXX-XXXXXX-XXXXXX-XXXXXX"...
License VZSRV was loaded successfully
---
1 of 1 licenses was loaded
```

Note: You can also use the `vzlicload` utility to upgrade the license. For example, this may be necessary if your current license does not support using Parallels Virtual Automation for managing Parallels servers and their virtual machines.

- If you have obtained the license in the form of an activation code, you can install it on the server using the `-a` option of the `vzlicupdate` command. For example:

```
# vzlicupdate -a XXXXXX-XXXXXX-XXXXXX-XXXXXX-XXXXXX
```

where `XXXXXX-XXXXXX-XXXXXX-XXXXXX-XXXXXX` is your activation code. When executed, `vzlicupdate` connects to the Parallels Key Authentication (KA) licensing server and transmits the specified activation code there. In its turn, the licensing server generates a license file, sends it back to the server from where the activation code has been dispatched, and automatically installs it on this server. So, before executing the aforementioned command, make sure that your Parallels server is connected to the Internet.

If you are activating your installation by means of an activation key, you must have an active Internet connection to successfully complete the license installation. Otherwise, you will be presented with the corresponding warning message informing you of the steps you have to take to activate your license. As a rule, these steps include the following:

- 1 Visiting the <http://www.parallels.com/en/support/virtuozzo/activate> web page and activating the license manually.
- 2 Providing the following information on this web page:
 - In the **Product Code** field, specify your license activation code.
 - In the **HWID** field, provide the ID of your server.
 - In the **Enter following digits** field, type the digits displayed next to this field.
- 3 Clicking the **ACTIVATE LICENSE** button.

If you have entered the correct information on the **Virtuozzo License Activation** page, you will be provided with a link to a license file that you should download to and install on the server. For example, you can run this command to install the obtained license file

```
# vzlicload -f /etc/vzlicense
```

This command will install the license file with the name of `vzlicense` on your server.

Updating the Current License

In Parallels Server for Mac Bare Metal Edition, you can use the `vzlicupdate` utility to update the license currently installed on the Parallels server. When executed, the utility tries to connect to the Parallels Key Authentication (KA) server and to retrieve a new license and install it on the server. To update your license, do the following:

- 1 Make sure that the Parallels server where you wish to update the license is connected to the Internet.
- 2 Execute the following command on the server:

```
# vzlicupdate
Start updating license [6E62.3D01.6BEC.E8D7.CE42.4517.68CB.E102]
...
```

By default, `vzlicupdate` tries to access the KA server having the hostname of `ka.parallels.com`. However, you can explicitly specify what KA server to use using the `--server` option:

```
# vzlicupdate --server ka.server.com
```

In this case, the `vzlicupdate` utility will try to connect to the KA server with the hostname of `ka.server.com`, to get a new license from this server, and to install it on the server where `vzlicupdate` has been executed.

Transferring the License to Another Server

Sometimes, you may wish to transfer licenses from one Parallels server (*source server*) to another (*destination server*). For example, this may be the case if the server where the license is installed starts experiencing problems or requires the hardware upgrade.

The procedure of transferring a license from one Parallels server to another depends on the license type and can be one of the following:

- If you have activated your Parallels Server for Mac Bare Metal Edition installation by means of a product key, you can transfer the installed license from the source to the destination server as follows:
 - Remove the installed license from the source server (e.g. using the `vzlicload -r product_key` command).
 - Log in to the destination server.
 - Install the product key on the destination server. Detailed information on how to install Parallels Server for Mac Bare Metal Edition licenses is provided in *Installing a License* (p. 73).
- If you have activated your Parallels Server for Mac Bare Metal Edition installation by means of an activation code, you can use the `vzlicupdate` utility to move licenses between Parallels servers. For example, to transfer a license that has been installed using the `XXXXXX-XXXXXX-XXXXXX-XXXXXX-XXXXXX` activation code, do the following:
 1. Ascertain that the source server is shut down, or the license is removed from this server.
 2. Make sure that the destination server is up and connected to the Internet.
 3. Log in to the destination server (e.g. via `ssh`).
 4. Execute the following command on the destination server:

```
# vzlicupdate -t -a XXXXXX-XXXXXX-XXXXXX-XXXXXX-XXXXXX
```

When executed, `vzlicupdate` sends the activation code to the Parallels KA server, thus informing the server of its intention to transfer the license to a new Parallels server. The KA server verifies the received code, generates a new license file, sends it back to the destination server, and installs it there.

You can check that the license transferal has completed successfully using the `vzlicview` utility. For example:

```
# vzlicview
Show installed licenses...
VZSRV
    status="ACTIVE"
    version=4.0
    serial="XXXXXX-XXXXXX-XXXXXX-XXXXXX-XXXXXX"
    expiration="05/01/2009 23:59:59"
    ...
```

Detailed information on the `vzlicview` utility and its output is provided in *Viewing Current License* (p. 76).

Viewing the Current License

The given subsection familiarizes you with the way to view the information on the license installed on your Parallels server.

Viewing the License

In Parallels Server for Mac Bare Metal Edition, you can use the `vzlicview` utility to view the information on the installed license and find out its current status. When executed, this utility processes the license currently installed on the Parallels server and prints the license contents along with its status. A sample output of `vzlicview` is given below:

```
# vzlicview
Show installed licenses
VZSRV
    status="ACTIVE"
    version=4.0
    serial="XXXXXX-XXXXXX-XXXXXX-XXXXXX-XXXXXX"
    expiration="12/01/2006 23:59:59"
    graceperiod=86400 (86400)
    key_number="PSBM.00000001.0000"
    cpu_total=64 (1)
    ct_total=8200 (1)
    max_vzmcPMC_users=128
    max_vzcc_users=260
    platform="Any"
    product="PSBM"
    vzpp_allowed=1
    backup_mgmt_allowed=1
    workflow_mgmt_allowed=1
    vzagent_allowed=1
    nr_vms=10
    architecture="Any"
```

The command output shows the full information about the license. The main license parameters are listed in the following table:

Column Name	Description
status	The license status. The information on all possible license statuses is provided in <i>License Statuses</i> (p. 78).
version	The version of Parallels Server for Mac Bare Metal Edition with which the license is compatible.
serial	The license serial number.
expiration	The license expiration date, if it is time-limited.
graceperiod	The period during which Parallels Server for Mac Bare Metal Edition continues functioning after your license has expired, in seconds.
key_number	The number under which the license is registered on the Parallels Key Authentication server.
cpu_total	The total number of central processor units (CPUs) which can be installed on the Parallels server.
ct_total	The total number of virtual machines which can simultaneously run on the Parallels server.
max_vzmc_users	The number of users able to simultaneously connect to the server using Parallels Management Console.
max_vzcc_users	The number of users able to simultaneously connect to the server using Parallels Virtual Automation (formerly, Parallels Infrastructure Manager).
platform	The operating system with which the license is compatible.

<code>product</code>	The product name for which the license has been issued.
<code>vzpp_allowed</code>	Indicates whether you can manage virtual machines using Parallels Power Panel: <ul style="list-style-type: none"> ▪ 1: the 'Parallels Power Panel' functionality is enabled ▪ 0: the 'Parallels Power Panel' functionality is disabled
<code>backup_mgmt_allowed</code>	Indicates whether the 'backup' functionality is enabled for the given server: <ul style="list-style-type: none"> ▪ 1: the 'backup' functionality is enabled ▪ 0: the 'backup' functionality is disabled
<code>workflow_mgmt_allowed</code>	Indicates whether the 'virtual machine requesting' functionality is enabled for the given server: <ul style="list-style-type: none"> ▪ 1: the 'virtual machine requesting' functionality is enabled ▪ 0: the 'virtual machine requesting' functionality is disabled
<code>vzagent_allowed</code>	Indicates whether you are allowed to use the Parallels Agent functionality on the given server: <ul style="list-style-type: none"> ▪ 1: the Parallels Agent functionality is enabled ▪ 0: the Parallels Agent functionality is disabled
<code>nr_vms</code>	The number of virtual machines which can simultaneously run on the Parallels server.
<code>architecture</code>	The system architecture with which the license is compatible.

License Statuses

When viewing information on your license, pay special attention to the license status that can be one of the following:

ACTIVE	The license installed on the server is valid and active.
VALID	The license the utility parses is valid and can be installed on the server.
EXPIRED	The license has expired and, therefore, could not be installed on the server.
GRACED	The license has been successfully installed on the server; however, it has expired and is currently on the grace period (i.e. it is active till the end of the grace period).
INVALID	The license is invalid (for example, because of the server architecture mismatch) or corrupted.

CHAPTER 6

Keeping Your System Up To Date

This chapter explains the ways to keep your Parallels server up to date. The components you need to take care of are the following:

- Parallels Server for Mac Bare Metal Edition software
- virtual machines created on the Parallels server

In This Chapter

Updating Parallels Server for Mac Bare Metal Edition Software	80
Updating Parallels Server for Mac Bare Metal Edition Software	81
Updating Software In Virtual Machines	84

Updating Parallels Server for Mac Bare Metal Edition Software

Parallels Server for Mac Bare Metal Edition is constantly developing: there appear new versions of the Parallels Server for Mac Bare Metal Edition core and of existing utilities, OS and application templates are perfected, new templates and utilities are also added from time to time. Thus, Parallels Server for Mac Bare Metal Edition may sometimes be repackaged to include the latest changes in any of its parts. As these changes grow in number, new product versions are released.

Parallels Server for Mac Bare Metal Edition provides a special utility - `vzup2date` - allowing you to easily and quickly update your Parallels server. The main components that need to be updated are the following:

- Parallels Server for Mac Bare Metal Edition system software (packages built by Parallels)
- Parallels Server for Mac Bare Metal Edition templates

`vzup2date` is intended to relieve Parallels Server for Mac Bare Metal Edition administrators of the necessity to manually update existing Parallels Server for Mac Bare Metal Edition installations. It provides a single information channel for learning if updated product versions are available. In other words, a regular launching of this utility helps ensure that you always have the latest version of Parallels Server for Mac Bare Metal Edition installed.

The `vzup2date` utility can be launched in two modes:

- Graphical mode. In this mode, you use a special wizard to update either the Parallels Server for Mac Bare Metal Edition system files or templates depending on the options passed to `vzup2date`.
- Command line mode containing two submodes:
 - the *batch* submode
 - the *messages* submode

In comparison to the graphical mode, the command line mode provides more possibilities for the Parallels Server for Mac Bare Metal Edition updates management (e.g. the ability to use special filters while selecting updates for your system).

Both modes are described in the following subsections in detail.

Updating Parallels Server for Mac Bare Metal Edition Software

Parallels Server for Mac Bare Metal Edition provides a special utility, `vzup2date`, allowing you to easily and quickly update the Parallels Server for Mac Bare Metal Edition system software on your Parallels server. The `vzup2date` utility can be launched in two modes:

- Graphical mode. In this mode, you use a special wizard to update the Parallels Server for Mac Bare Metal Edition system files.
- Command line mode containing two submodes:
 - the *batch* submode
 - the *messages* submode

In comparison to the graphical mode, the command line mode provides more possibilities for the Parallels Server for Mac Bare Metal Edition updates management (e.g. the ability to use special filters while selecting updates for your system).

Both modes are described in the following subsections in detail.

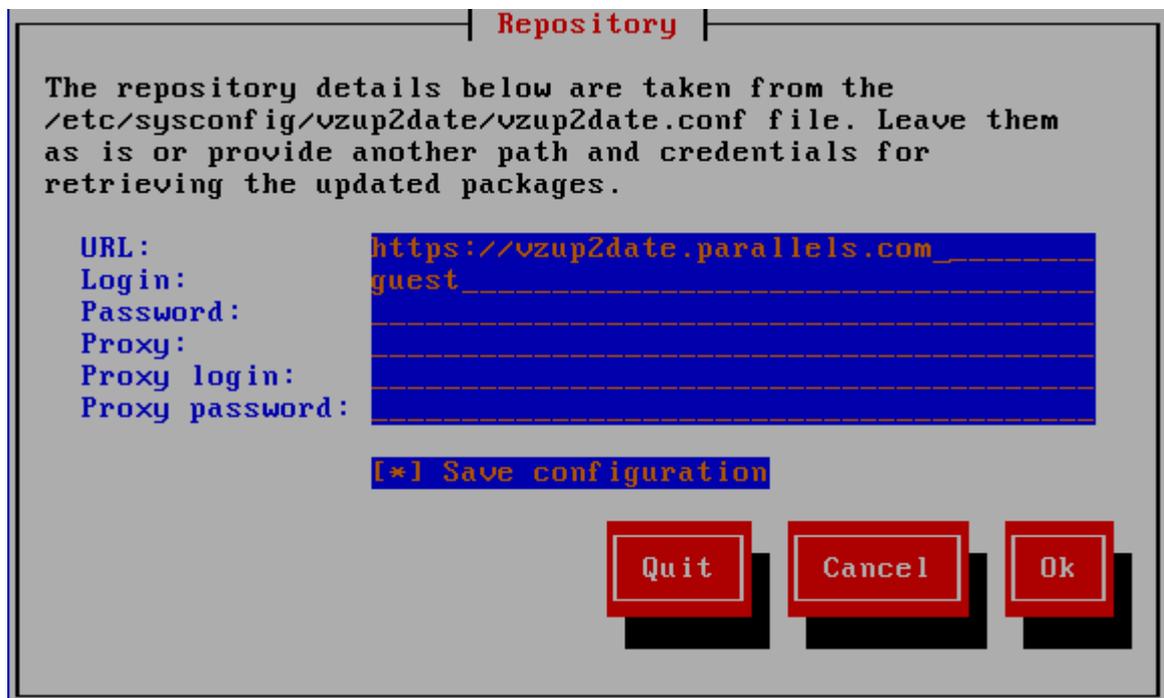
Updating in Graphical Mode

In the graphical mode, the `vzup2date` utility can be launched in two submodes. If invoked without any parameters or with the `-s` switch, it is supposed to check and, if necessary, download and install Parallels Server for Mac Bare Metal Edition system files. On the other hand, specifying the `-z` option when invoking the utility tells it to perform the same operations for OS and application EZ templates. There is no single interface for checking system files and templates at once, as these operations are different in nature. Therefore, you should consecutively call the `vzup2date` utility with and without the `-z` option, if you wish to check for all available system and template updates.

Note: You can explicitly specify that the `vzup2date` utility is to be run in the graphical mode by passing the `-m interactive` switch to it.

To run the `vzup2date` utility in the graphical mode, you should invoke it without any parameters or with the `-s` switch. In this mode, the `vzup2date` utility is implemented as a wizard the first few steps of which are common for both modes. After you launch the utility from the command line, you will be presented with the **Welcome** screen. In this window, you can do one of the following:

- Click the **Next** button to connect to the Parallels default repository.
- Click the **Configure** button to display the current settings used to connect to the repository housing Parallels Server for Mac Bare Metal Edition updated packages and templates and to configure it, if necessary.



The information on this screen is taken from the `/etc/sysconfig/vzup2date/vzup2date.conf` file on the Parallels server. If you want to change this information and save the changes to the configuration file, enter the correct settings into the fields provided, and press **OK**.

Once you press **Next** in the **Welcome** window, the utility will try to connect to the specified repository (either the Parallels default repository or your own one) and, if the connection is successful, display the **Updates** window. This window displays the list of updates that can be installed on your Parallels server. If you want to update to the latest Parallels Server for Mac Bare Metal Edition core and utilities versions, just press **Next** on this screen, and the `vzup2date` utility will download and install them asking your confirmation before each action.

Note: The `vzup2date` utility might see that the selected update includes an updated version of the `vzup2date` utility itself. In this case you will first have to perform an update of this utility and then to re-launch it and select the desired Parallels Server for Mac Bare Metal Edition system update once again.

If you have a reason not to install the latest updates for both the Parallels Server for Mac Bare Metal Edition core and utilities, press **Customize**. In this case, you can choose whether to perform customization on the Parallels Server for Mac Bare Metal Edition core or utilities. This step is skipped if updates are currently available either only for the core or only for utilities. In the next step, you will be asked to choose the desired core or utilities updates, in case there are many.

Notice that the bottommost update includes the functionality of all the other updates. You can select any of the intermediary updates and press **Select** to go back to the **List of Selected Updates** screen and read the information on this update. You will be able to perform customization more than once until you finally decide on the set of updates to be applied and press **Next**.

Downloading and installing the selected updates is straightforward.

Updating in Command Line Mode

Another way of updating your Parallels Server for Mac Bare Metal Edition system files and templates is to run the `vzup2date` utility in the command line mode and to pass the corresponding commands, switches, and options to it. While executing `vzup2date` in the command line mode, you can choose between the batch and messages submodes. Both submodes can be used to update either the Parallels Server for Mac Bare Metal Edition system files or EZ templates and have the identical syntax. However, the output produced by these commands is different. The messages submode output is less user friendly than that of the batch submode and is mostly suitable for machine processing.

To run the `vzup2date` utility in the command line mode, you can use either the `-m batch` switch or the `-m messages` switch intended for executing `vzup2date` in the batch and messages submodes, respectively.

Let us assume that you wish to update Parallels Server for Mac Bare Metal Edition system files by installing the latest core in the batch submode. To do this, you can issue the following command on the Parallels server:

```
# vzup2date -m batch install --core
```

This will check the Parallels Server for Mac Bare Metal Edition repository for the latest core updates and, in the case of finding any, download and install them on your server.

To update your Parallels Server for Mac Bare Metal Edition installation, you may need to edit the `/etc/sysconfig/vzup2date/vzup2date.conf` file to specify the repository from where the updates are to be downloaded or configure a number of other parameters. Detailed information on the `vzup2date.conf` file is provided in the *Parallels Command Line Reference Guide*.

Note: To perform the aforementioned operations in the messages submode, you should pass the `-m messages` option to the `vzup2date` utility instead of `-m batch`.

Updating Software In Virtual Machines

To keep software in your virtual machines up to date, you can use the same means you would use on standalone computers running the corresponding operating systems:

- In Linux-based virtual machines, you can use the native Linux updaters (`up2date`, `yum`, or `yast`).
- In Windows-based virtual machines, you can use the native Windows updaters (e.g. the Windows Update tool).

You should regularly run these updaters to ensure that your system has the latest updates and fixes (including security patches) installed. For more information on native updaters, refer to the documentation shipped with your operating system.

Glossary

Guest operating system (Guest OS). An operating system installed inside a virtual machine. It can be any of the supported Windows, Linux, or Mac operating systems.

Hardware virtualization. A virtualization technology allowing you to virtualize physical servers at the hardware level. Hardware virtualization provides the necessary environment for creating and managing Parallels virtual machines.

Parallels Management Console. A Parallels Server for Mac Bare Metal Edition management and monitoring tool with graphical user interface. Parallels Management Console is cross-platform and can run on Microsoft Windows, Linux, and Mac computers.

Parallels Server. A hardware virtualization solution that enables you to efficiently use your physical server's hardware resources by sharing them between multiple virtual machines created on this server.

Parallels server (or physical server or server). A server where the Parallels Server for Mac Bare Metal Edition software is installed for hosting Parallels virtual machines.

Parallels Server for Mac Bare Metal Edition license. A special license that you should install on the physical server to be able to start using Parallels Server for Mac Bare Metal Edition. Every physical server must have its own license installed.

Virtual machine (VM). A computer emulated by Parallels Server for Mac Bare Metal Edition. A virtual machine is functionally identical to an isolated standalone computer, with its operating system, IP addresses, processes, files, its own users database, its own configuration files, its own applications, system libraries, and so on.

Index

A

- About This Guide - 5
- Adding a New Device - 47

B

- Backups Overview - 28
- Basics of Hardware Virtualization - 9

C

- Changing the Disk Type - 43
- Compacting the Virtual Disk - 46
- Configuring Network Settings - 20
- Configuring Passwordless Access to the Source Node - 35
- Configuring Virtual Adapter Parameters - 70
- Configuring Virtual Devices - 51
- Configuring Virtual Network Parameters - 66
- Connecting an Adapter to a Virtual Network - 63
- Connecting Virtual Machines to Virtual Networks - 71
- Copying a Virtual Machine Within the Server - 25
- Create a Template - 41
- Creating a Snapshot - 54
- Creating a Virtual Machine - 18
- Creating a Virtual Network - 65
- Creating and Deleting Virtual Adapters - 69
- Creating VLAN Adapter - 62

D

- Deleting a Device - 52
- Deleting a Snapshot - 56
- Deleting a Virtual Machine - 27
- Deleting a Virtual Network - 68
- Deploying a Template - 42
- Documentation Conventions - 6

F

- Feedback - 8

G

- General Migration Requirements - 37
- Getting Help - 8
- Glossary - 85

I

- Increasing the Virtual Disk Capacity - 44
- Initializing the Newly Added Disk - 49
- Installing Parallels Tools - 21
- Installing the License - 73
- Introduction - 5

K

- Keeping Your System Up To Date - 79

L

- License Statuses - 78
- Listing Adapters - 61
- Listing Snapshots - 55
- Listing Templates - 42
- Listing Virtual Machines - 23
- Listing Virtual Networks - 67

M

- Making Screenshots - 57
- Managing Adapters in Virtual Machines - 68
- Managing Licenses - 72
- Managing Network Adapters on the Parallels Server - 60
- Managing Parallels Server for Mac Bare Metal Edition Network - 60
- Managing Snapshots - 53
- Managing Templates - 41
- Managing Virtual Machine Backups - 28
- Managing Virtual Machine Devices - 46
- Managing Virtual Machine Disks - 43
- Managing Virtual Machine Resources - 58
- Managing Virtual Networks - 64
- Migrating a Physical Computer to a Virtual Machine - 39
- Migrating Virtual Machines - 36
- Migrating Virtual Machines Between Parallels Servers - 38

O

- Operations on Virtual Machines - 17
- Organization of This Guide - 6

P

- Parallels Management Console - 14

Parallels Server 4.0 for Mac Bare Metal
Edition Basics - 9
Parallels Virtual Machines - 10
Pausing a Virtual Machine - 26
Performing Initial Configuration - 19
Physical Server Availability Considerations -
16

R

Reducing the Virtual Disk Capacity - 45
Requirements for Migrating to Virtual
Machines - 40
Resource Management - 15
Reverting to a Snapshot - 56
Running Commands in a Virtual Machine - 27

S

Setting Startup Parameters - 20
Setting the Password for a Virtual Machine -
20
Starting, Stopping, and Querying Status of a
Virtual Machine - 22
Storing Extended Information on a Virtual
Machine - 24
Support of Virtual and Real Media - 13
Supported Guest Operating Systems - 19
Suspending a Virtual Machine - 26

T

Transferring the License to Another Server -
75

U

Understanding Licensing - 15
Updating in Command Line Mode - 84
Updating in Graphical Mode - 82
Updating Parallels Server for Mac Bare Metal
Edition Software - 80, 81
Updating Software In Virtual Machines - 84
Updating the Current License - 74
Using pbackup and prestore - 33
Using pctl backup and pctl restore - 30

V

Viewing the Current License - 76
Viewing the License - 77
Virtual Machine Files - 12
Virtual Machine Hardware - 11