Parallels Virtuozzo Containers 4.7 for Linux

User's Guide

Parallels IP Holdings GmbH.
c/o Parallels International GmbH.
Parallels International GmbH
Vordergasse 49
CH8200 Schaffhausen
Switzerland
Tel: + 41 526320 411
Fax: + 41 52672 2010
www.parallels.com

Contents

CHAPTER 1

Introduction

This chapter provides basic information about Parallels Virtuozzo Containers 4.7 and this guide.

In This Chapter

About This Guide

This guide is meant to provide comprehensive information on Parallels Virtuozzo Containers 4.7—high-end server virtualization software for Linux-based servers. The issues discussed in this guide cover the necessary theoretical conceptions as well as practical aspects of working with Parallels Virtuozzo Containers. The guide will teach you to create and administer Containers (sometimes also called *Virtual Environments*, or *VEs*) on servers running the Parallels Virtuozzo Containers software and to employ both graphical and command line interfaces for performing various tasks.

> **Note:** The guide does not familiarize you with the process of installing, configuring, and deploying Parallels Virtuozzo Containers systems. Detailed information on these operations is given in the *Parallels Virtuozzo Containers Installation Guide*.

According to the task-oriented approach, most topics of this guide are devoted to a particular task and the ways to perform it. However, Parallels Virtuozzo Containers is equipped with as many as three different tools to perform administrative tasks:

- the command-line interface
- Parallels Management Console with the graphical user interface
- Parallels Virtual Automation with the web interface

The given guide describes the ways to administer Parallels Virtuozzo Containers and perform main tasks on Hardware Nodes (servers running the Parallels Virtuozzo Containers software) and in the Container context using Parallels Management Console and the command-line interface. As to Parallels Virtual Automation, it is provided with a comprehensive online help system.

Besides, there is another tool for managing Containers—Parallels Power Panel. This web-based tool is mainly regarded as a means for individual Container users to manage their personal Containers and also has its own online help system.

Organization of This Guide

Chapter 2, Parallels Virtuozzo Containers Philosophy, is a must-read chapter that helps you grasp the general principles of Parallels Virtuozzo Containers operation. It provides an outline of Parallels Virtuozzo Containers architecture and main features, of the way Parallels Virtuozzo Containers stores and uses configuration information, and of the Parallels Virtuozzo Containers licensing policy.

Chapter 3, Operations on Containers, describes operations you can perform on Containers: creating and deleting Containers, starting and stopping them, backing up and restoring Containers, and so on. You will also learn how to migrate Containers from one Hardware Node to another.

Chapter 4, Managing Resources, focuses on configuring and monitoring the resource control parameters for Containers. These parameters comprise disk quotas, network accounting and shaping, CPU and system resources.

Chapter 5, Real-Time Monitoring in Parallels Virtuozzo Containers, explains the way to keep track of the resources consumption by running Containers and the Hardware Node itself in real time.

Chapter 6, Managing Services and Processes, describes the operations you can perform on processes and services in Parallels Virtuozzo Containers by using both the command-line utilities and Parallels Management Console graphical interface.

Chapter 7, Managing Parallels Virtuozzo Containers Network, familiarizes you with the Parallels Virtuozzo Containers network structure, enumerates Parallels Virtuozzo Containers networking components, and explains how to manage these components in Parallels Virtuozzo Containers-based systems.

Chapter 8, Managing Hardware Nodes, centers on all those operations you can perform on Hardware Nodes.

Chapter 9, Advanced Tasks, enumerates those tasks that are intended for advanced system administrators who would like to obtain deeper knowledge about Parallels Virtuozzo Containers capabilities.

Chapter 10, Mastering Parallels Management Console, focuses on those tasks that are most comfortably accomplished using not the command-line utilities, but Parallels Management Console graphical interface.

Chapter 11, Troubleshooting, suggests ways to resolve common inconveniences should they occur during your work with the Parallels Virtuozzo Containers software.

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.

Formatting convention	Type of Information	Example
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.
Italics	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 *OS templates*. To remove a Container, type `vzctl delete` *`ctid`*.
`Monospace`	The names of commands, files, and directories.	Use `vzctl start` to start a Container.
`Preformatted`	On-screen computer output in your command-line sessions; source code in XML, C++, or other programming languages.	`Saved parameters for Container 101`
`Monospace Bold`	What you type, as contrasted with on-screen computer output.	**`C:\vzlist -a`**
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 shipped with Parallels Virtuozzo Containers 4.7 that can help you use the product more effectively. These resources include:

- *Getting Started With Parallels Virtuozzo Containers 4.7 for Linux*. This guide provides basic information on installing Parallels Virtuozzo Containers 4.7 on your server, creating new Containers, and performing the main operations on them.
- *Parallels Virtuozzo Containers 4.7 for Linux Installation Guide*. This guide provides exhaustive information on the process of installing, configuring, and deploying your Parallels Virtuozzo Containers system. Unlike the *Getting Started With Parallels Virtuozzo Containers 4.7 for Linux* guide, it contains a more detailed description of the operations needed to install and set Parallels Virtuozzo Containers to work (e.g., planning the structure of your network and performing the Parallels Virtuozzo Containers unattended installation). Besides, it does not include the description of any Container-related operations.
- *Parallels Virtuozzo Containers 4.7 for Linux Templates Management Guide*. This guide is meant to provide complete information on Parallels Virtuozzo Containers templates—an exclusive Parallels technology allowing you to efficiently deploy standard Linux applications inside Containers and to greatly save the server resources (physical memory, disk space, and so on).
- *Parallels Virtuozzo Containers 4.7 for Linux Reference Guide*. This guide is a complete reference on all Parallels Virtuozzo Containers configuration files and command-line utilities.
- *Parallels Management Console Help*. This help system provides detailed information on Parallels Management Console—a graphical user interface tool for managing Hardware Nodes and Containers.
- *Parallels Virtual Automation Online Help*. This help system shows you how to work with Parallels Virtual Automation—a tool providing you with the ability to manage Hardware Nodes and Containers with the help of a standard Web browser on any platform.
- *Parallels Power Panel Online Help*. This help system deals with Parallels Power Panel—a means for administering individual Containers through a common Web browser on any platform.

Feedback

If you spot a typo in this guide, or if you have an opinion about how to make this guide more helpful, you can share your comments and suggestions with us by completing the Documentation Feedback form on our website (http://www.parallels.com/en/support/usersdoc/).

CHAPTER 2

Parallels Virtuozzo Containers Philosophy

This chapter describes the general principles of Parallels Virtuozzo Containers operation. It provides an outline of the Parallels Virtuozzo Containers architecture and lets you understand the Parallels Virtuozzo Containers licensing policy.

In This Chapter

About Parallels Virtuozzo Containers Software

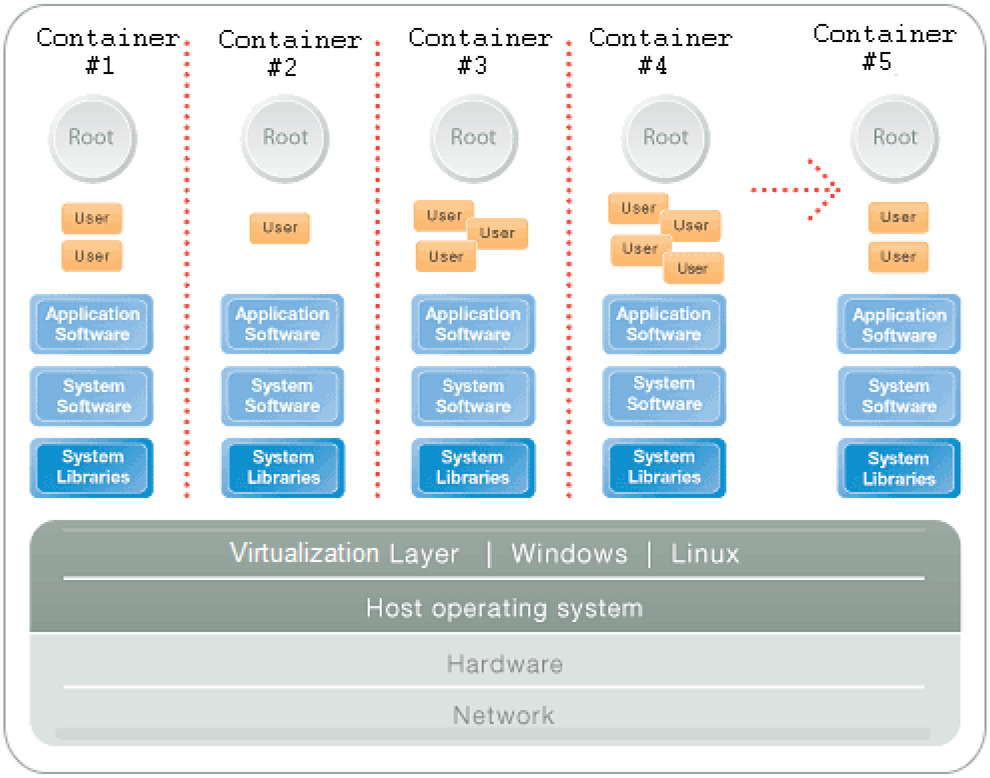
This section provides general information about the Parallels Virtuozzo Containers software and its applications.

What is Parallels Virtuozzo Containers

Parallels Virtuozzo Containers is a patented OS virtualization solution. It creates isolated partitions or Containers on a single physical server and OS instance to utilize hardware, software, data center and management effort with maximum efficiency. The basic Parallels Virtuozzo Containers capabilities are:

- **Intelligent Partitioning**. Divide a server into as many as hundreds of Containers with full server functionality.
- **Complete Isolation**. Containers are secure and have full functional, fault and performance isolation.
- **Dynamic Resource Allocation**. CPU, memory, network, disk and I/O can be changed without rebooting.
- **Mass Management**. Suite of tools and templates for automated, multi-Container and multi-server administration.

The diagram below represents a typical model of the Parallels Virtuozzo Containers-based system structure:



The Parallels Virtuozzo Containers OS virtualization model is streamlined for the best performance, management, and efficiency. At the base resides a standard Host operating system which can be either Windows or Linux. Next is the virtualization layer with a proprietary file system and a kernel service abstraction layer that ensure the isolation and security of resources between different Containers. The virtualization layer makes each Container appear as a standalone server. Finally, the Container itself houses the application or workload.

The Parallels Virtuozzo Containers OS virtualization solution has the highest efficiency and manageability making it the best solution for organizations concerned with containing the IT infrastructure and maximizing the resource utilization. The Parallels Virtuozzo Containers complete set of management tools and unique architecture makes it the perfect solution for easily maintaining, monitoring, and managing virtualized server resources for consolidation and business continuity configurations.

What is Container

A Container is a virtual private server that is functionally identical to an isolated standalone server:

- Each Container has its own processes, users, files and provides full administrative access.
- Each Container has its own IP addresses, port numbers, filtering and routing rules.
- Each Container can have its own configuration for the system and application software, as well as its own versions of system libraries. It is possible to install or customize software packages inside a Container independently from other Containers or the host system. Multiple distributions of a package can be run on one and the same server.
- Each Container has its own unique `root` user with full control over the given Container and full access to other user accounts inside this Container.

Parallels Virtuozzo Containers Applications

Parallels Virtuozzo Containers 4.7 can be efficiently applied in a wide range of areas: enterprise server consolidation, web and applications hosting, software development and testing, user training, and so on.

If you administer a number of Linux dedicated servers within an enterprise, you can benefit from the Parallels Virtuozzo Containers solution in the following ways:

- Reduce the number of required physical servers and corresponding support by grouping a multitude of your enterprise servers onto a single server without losing a bit of valuable information and without compromising performance.
- Increase server utilization and maximize server potential.
- Provision servers in minutes by using the technology of Parallels Virtuozzo Containers templates.
- Migrate Containers in the time of network data transfer, nearly eliminating the planned downtime and enabling fast reaction to unplanned downtime situations.
- Monitor OS and application versions and update/upgrade the current software easily across all of your physical servers running the Parallels Virtuozzo Containers software and their Containers.
- Guarantee Quality-of-Service in accordance with a corporate service level agreement (SLA).
- Automate routine tasks such as upgrades and updates.
- Minimize software license and support requirements.

Due to its unique efficiency and completeness, Parallels Virtuozzo Containers has also a wide variety of profitable uses for hosting service providers allowing them to:

- Provide complete self-administration panels (Parallels Power Panel) including system backup/restore and monitoring tools.
- Have a multitude of customers with their individual full-featured Containers sharing a single physical server.
- Transparently move customers and their environments between servers, without any manual reconfiguration.
- Increase profitability through the better management and leverage of hardware and software investments.
- Automate service provisioning by using the technology of Parallels Virtuozzo Containers templates.

Besides, Parallels Virtuozzo Containers proves invaluable for IT educational institutions that can now provide every student with a personal Linux server, which can be monitored and managed remotely. Software development companies may use Containers for testing purposes and the like.

Distinctive Features of Parallels Virtuozzo Containers

The concept of Parallels Virtuozzo Containers is distinct from the concept of traditional virtual machines in the respect that Containers always run the same OS kernel as the host system (that is, Linux on Linux or Windows on Windows). This single-kernel implementation technology allows you to run Containers with a near-zero overhead. Thus, Parallels Virtuozzo Containers offer an order of magnitude higher efficiency and manageability than traditional virtualization technologies.

OS Virtualization

From the point of view of applications and Container users, each Container is an independent system. This independence is provided by a virtualization layer in the kernel of the host OS. Note that only a negligible part of the CPU resources is spent on virtualization (around 1-2%). The main features of the virtualization layer implemented in Parallels Virtuozzo Containers are the following:

- Container looks like a normal Linux system. It has standard startup scripts, software from vendors can run inside Container without Parallels Virtuozzo Containers-specific modifications or adjustment.
- A user can change any configuration file and install additional software.
- Containers are fully isolated from each other (file system, processes, Inter Process Communication (IPC), `sysctl` variables).
- Containers share dynamic libraries, which greatly saves memory.
- Processes belonging to a Container can be scheduled for execution on all available CPUs. Consequently, Containers are not bound to only one CPU and can use all available CPU power.

Using Virtuozzo File System

Virtuozzo File System (VZFS) is a legacy file system that allows sharing common files among multiple Containers without sacrificing flexibility. Container users can modify, update, replace, and delete shared files. When a user modifies a shared file, VZFS creates a private copy of that file transparently for the user. Thus, modifications do not affect other users of the same file.

Although VZFS can help you save disk space and memory, it also has a number of limitations:

- You cannot store Containers using VZFS in Parallels Cloud Storage clusters.
- To migrate or restore a Container, you always need to have a corresponding OS template installed on the destination server.
- VZFS-based Containers lack some functionality provided by Parallels Virtuozzo Containers 4.7 (like creating and managing snapshots).

Note: For more information on VZFS, see the documentation for Parallels Server Bare Metal 5.0.

Templates

A template (or a package set) in Parallels Virtuozzo Containers is a set of original application files repackaged for mounting over Virtuozzo File System. Usually, it is just a set of RPM packages for Red Hat like systems. Parallels Virtuozzo Containers provides tools for creating templates, installing, upgrading, adding them to and removing them from a Container. Using templates lets you:

- Share the RAM among similar applications running in different Containers to save hundreds of megabytes of memory.
- Share the files comprising a template among different Containers to save gigabytes of disk space.
- Deploy applications simultaneously in many Containers.
- Use different versions of an application in different Containers (for example, perform an upgrade only in certain Containers).

There are two types of templates in Parallels Virtuozzo Containers 4.7. These are OS templates and application templates. An OS template is an operating system and the standard set of applications to be found right after the installation. Parallels Virtuozzo Containers uses OS templates to create new Containers with a preinstalled operating system. An application template is a set of repackaged software packages optionally accompanied with configuration scripts. Parallels Virtuozzo Containers uses application templates to add extra software to an existing Container. For example, you can create a Container on the basis of the CentOS 5 OS template and add the MySQL application to it using the MySQL application template.

For detailed information on Parallels Virtuozzo Containers templates, see the *Parallels Virtuozzo Containers 4.7 Templates Management Guide*.

Resource Management

Parallels Virtuozzo Containers resource management controls the amount of resources available to Containers. The controlled resources include such parameters as CPU power, disk space, a set of memory-related parameters. Resource management allows Parallels Virtuozzo Containers to:

- effectively share available Hardware Node resources among Containers
- 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 Containers
- manage a multitude of Hardware Nodes in a unified way by means of Parallels Management Console and Parallels Virtual Automation
- collect usage information for system health monitoring

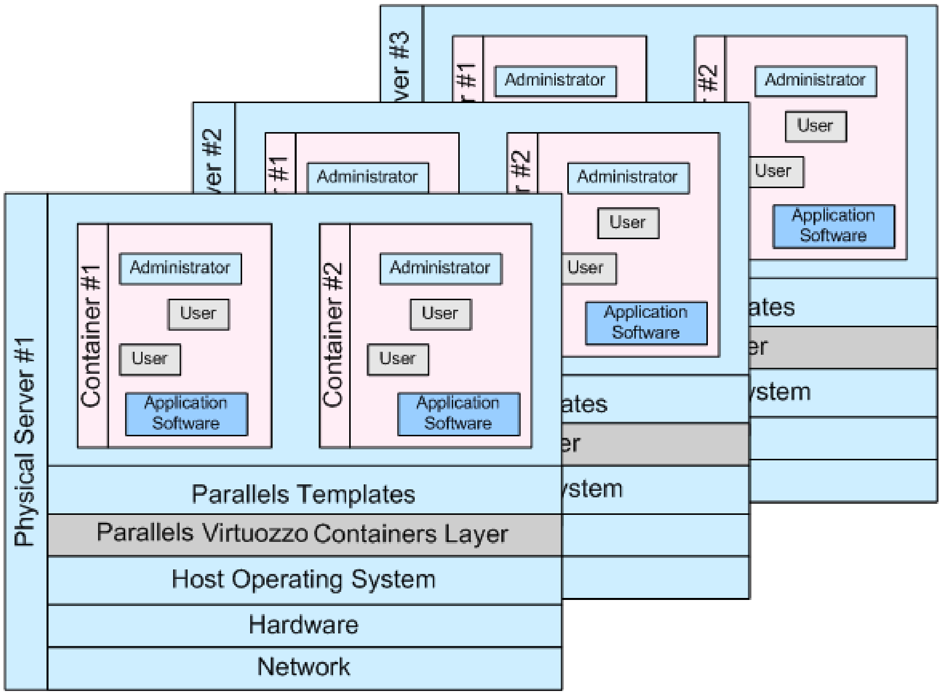
Resource management is much more important for Parallels Virtuozzo Containers than for a standalone server since server resource utilization in a Parallels Virtuozzo Containers-based system is considerably higher than that in a typical system.

Main Principles of Parallels Virtuozzo Containers Operation

This section describes the basics of Parallels Virtuozzo Containers technology and discusses the main tools for managing Parallels Virtuozzo Containers-based systems.

Basics of Parallels Virtuozzo Containers Technology

In this section, we will try to let you form a more or less precise idea of the way the Parallels Virtuozzo Containers software operates on your computer. Please see the figure below:



This figure presumes that you have a number of physical servers united into a network. In fact, you may have only one dedicated server to effectively use the Parallels Virtuozzo Containers software for the needs of your network. If you have more than one Parallels Virtuozzo Containers-based physical server, each one of the servers will have a similar architecture. In Parallels Virtuozzo Containers terminology, such servers are called *Hardware Nodes* (or just *Nodes*), because they represent hardware units within a network.

Parallels Virtuozzo Containers 4.7 is installed on a Linux operating system configured in a certain way. For example, such customized configuration should include the creation of a `/vz` partition, which is the basic partition for hosting Containers and which must be way larger than the root partition.

Note: For the full list of supported operating systems and detailed instructions on installing Linux (called *Host Operating System* in the picture above) on physical servers, see the *Parallels Virtuozzo Containers 4.7 Installation Guide*.

Once Parallels Virtuozzo Containers is installed, you can run Parallels Virtuozzo Containers services supporting virtualization on your server. This support is presented above as *Parallels Virtuozzo Containers Layer*. The Parallels Virtuozzo Containers layer ensures that Containers, sharing the same *Hardware Node* and the same OS kernel, are isolated from each other. A Container is a kind of 'sandbox' for processes and users.

Before you are able to create a Container, you need to install the corresponding OS template in Parallels Virtuozzo Containers 4.7. This is displayed as *Parallels Templates* in the scheme above. Different Containers can be based on different OS templates and thus run different version of Linux (for example, Ubuntu 10.4 or Fedora 13). Once you install at least one OS template, you can create any number of Containers with the help of various Parallels management tools (the Parallels Virtuozzo Containers command-line tools, Parallels Virtual Automation, or Parallels Management Console), configure their network and/or other settings, and work with these Containers as with fully functional LInux servers.

Parallels Virtuozzo Containers Configuration

Parallels Virtuozzo Containers 4.7 allows you to flexibly configure various settings for your Parallels Virtuozzo Containers system in general as well as for each and every Container. Among these settings are disk and user quota, network parameters, default file locations and configuration sample files, and others.

Parallels Virtuozzo Containers stores the configuration information in two types of files: the global configuration file `/etc/vz/vz.conf` and Container configuration files `/etc/vz/conf/<CT_ID>.conf`. The global configuration file defines global and default parameters for Container operation, for example, logging settings, enabling and disabling disk quota for Containers, the default configuration file and OS template on the basis of which a new Container is created, and so on. On the other hand, a Container configuration file (`/etc/vz/conf/CT_ID`) defines the parameters for a particular Container, such as disk quota and allocated resources limits, IP address and host name, and so on. If a parameter is configured in both the global configuration file and the Container configuration file, the Container configuration file takes precedence. For the list of parameters that can be configured in the global and Container configuration files, see the *Parallels Virtuozzo Containers 4.7 Reference Guide*.

The configuration files are read when the Parallels Virtuozzo Containers software and/or Containers are started. However, Parallels Virtuozzo Containers standard utilities (for example, `vzctl`) allow you to change many configuration settings “on-the-fly”, either without modifying the corresponding configuration files or with their modification (if you want the changes to apply the next time the Parallels Virtuozzo Containers software and/or Containers are started).

Some Parallels Virtuozzo Containers utilities have their own configuration files. For example, `vzbackup`, which is responsible for backing up Container private areas and configuration files, has its own global configuration file `/etc/vzbackup.conf` and can have a number of per-Node configuration files located in the backup directory. This directory is defined in the backup global configuration file. Both the global backup configuration file and per-Node ones are located on a central *Backup Node*. There are a number of other specific configuration files. All of them are described in detail in the *Parallels Virtuozzo Containers 4.7 Reference Guide*.

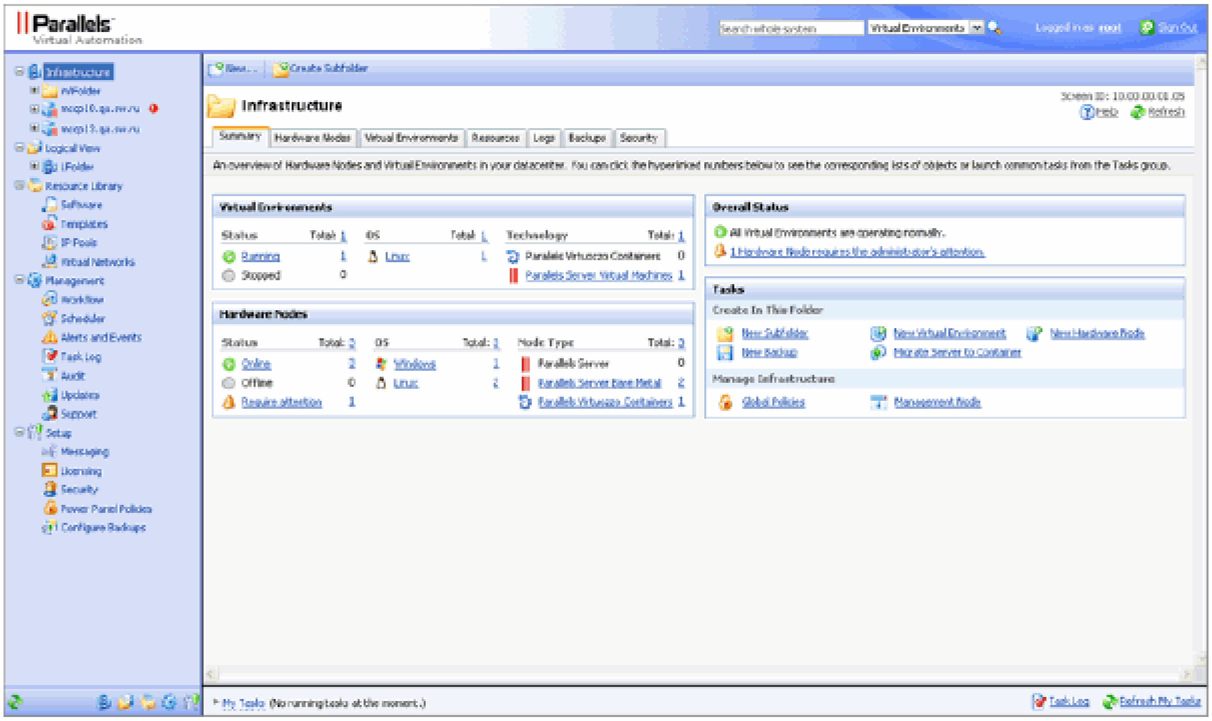
Parallels Virtual Automation Overview

Parallels Virtual Automation is designed for Hardware Node administrators and provides them with the ability to manage multiple Hardware Nodes and all Containers residing on them with the help of a standard web browser on any platform. The list of supported browsers is given below:

- Internet Explorer 6 and above
- Firefox 2.0 and above
- Safari 3.0 and above

Chances are that you will also be able to use other browsers, but Parallels Virtuozzo Containers has not been extensively tested with them.

The Parallels Virtual Automation interface has been designed to let the Parallels Virtuozzo Containers server administrator quickly perform all possible tasks through an intuitive navigation system:



An overview of Hardware Nodes and Virtual Environments in your datacenter. You can click the hyperlinked numbers below to see the corresponding lists of objects or launch common tasks from the Tasks group.

Virtual Environments

Status	Total: 1	OS	Total: 1	Technology	Total: 1
Running	1	Linux	1	Parallels Virtuozzo Containers	0
Stopped	0			Parallels Server Virtual Machines	1

Overall Status

All Virtual Environments are operating normally.

1 Hardware Node requires the administrator's attention.

Hardware Nodes

Status	Total: 3	OS	Total: 3	Node Type	Total: 3
Online	3	Windows	1	Parallels Server	0
Offline	0	Linux	2	Parallels Server Bare Metal	2
Require attention	1			Parallels Virtuozzo Containers	1

(No running tasks at the moment.)

The main components the Parallels Virtual Automation interface include:

- The left menu frame listing and allowing to access all your Hardware Nodes and Containers and the main types of operations to be performed on them with the help of Parallels Virtual Automation.
- The toolbar on top of the right frame allowing to perform on your Hardware Nodes and Containers the actions most frequently called for in your routine management work and, when necessary, a few more buttons allowing to perform additional actions on the objects listed in the content part of the right frame (Container backups, packages updates, etc.).
- The content part on the right frame displaying the currently accessed Hardware Nodes or Containers, the key information (their statuses, configuration, etc.) and links to advanced actions.

Note: Detailed information on Parallels Virtual Automation is given in its comprehensive online help system and the *Parallels Virtual Automation Administrator*'s *Guide*.

Parallels Power Panel Overview

Wherever Parallels Virtuozzo Containers is applied, there are people who are supposed to be administrators of particular Containers only, with no access rights to Hardware Nodes. Such people can be subscribers to a hosting provider, university students, administrators of a particular server within an enterprise, etc. Personal Containers can be managed with the help of Parallels Power Panel. Power Panel is a means for administering personal Containers through a common browser: Internet Explorer, Mozilla, and others. It allows Container administrators to do the following:

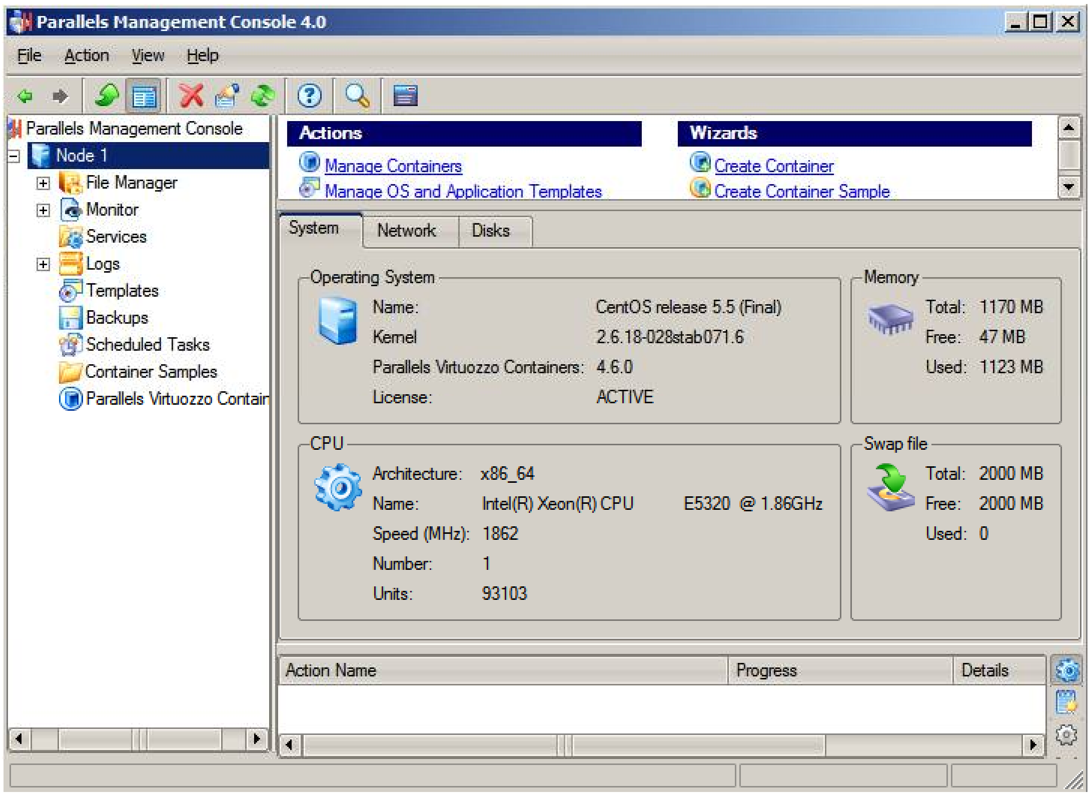
- Start, stop, or restart the Container.
- Back up and restore the Container.
- Change the Administrator password of the Container.
- Start, stop, or restart certain services inside the Container.
- View the processes currently running in the Container and send signals to them.
- View the current resources consumption and resources overusage alerts.
- Connect to the Container by means of RDP.
- View the system logs.

For more information on Parallels Power Panel, see its online help system.

Note: Apart from Parallels Power Panel, Container administrators are able to use the standard Windows Remote Desktop Connection (RDP) or MS Terminal Service Client (MS TSC) to connect to their Containers and work inside them.

Parallels Management Console Overview

Parallels Management Console is a remote management tool for Parallels Virtuozzo Containers with a graphical user interface. You can use to control Hardware Nodes, to manage Containers, and to monitor the system. The main window of Management Console consists of two parts: the tree pane on the left, and view pane on the right. There is a list of Hardware Nodes in the tree pane. The Hardware Node subtree represents various aspects of its management, for example, **Logs**, **Services**, and **Templates**. The content of the view pane depends on the selected item in the tree pane.



Operating System

Name: CentOS release 5.5 (Final)

Kernel 2.6.18-028stab071.6

Parallels Virtuozzo Containers: 4.6.0

License: ACTIVE

Memory

Total: 1170 MB

Free: 47 MB

Used: 1123 MB

CPU

Architecture: x86_64

Name: Intel(R) Xeon(R) CPU E5320 @ 1.86GHz

Speed (MHz): 1862

Number: 1

Units: 93103

Swap file

Total: 2000 MB

Free: 2000 MB

Used: 0

Below the view pane on the right, there is also a small Actions/Messages/Operations pane. You can switch between the modes by clicking the corresponding buttons to the right of this pane. The Actions pane displays the progress of Parallels Management Console actions. The Messages pane displays the detailed diagnostics of various Management Console errors. The Operations pane shows the result of various asynchronous tasks performed with Hardware Nodes and their Containers.

Parallels Management Console uses a typical client/server architecture. The client Management Console program runs on Microsoft Windows XP/2003/2008/2008 R2. The client application with the graphical user interface connects to the Parallels Agent software, which is running on the Hardware Node. Parallels Agent communicates with the client via the well-documented open Parallels Agent XML API and controls the Hardware Node itself and its Containers.

The client can control multiple Hardware Nodes simultaneously by connecting to multiple Parallels Agents. As the communications between the client and Parallels Agents are secure, the Management Console workstation may be located virtually anywhere on the network.

More detailed information on installing Parallels Management Console is given in the *Parallels Virtuozzo Containers 4.7 Installation Guide*.

Hardware Node Availability Considerations

Hardware Node availability is more critical than the availability of a typical server. Since it runs multiple Containers providing a number of critical services, Hardware Node outage may be very costly. Hardware Node outage can be as disastrous as the simultaneous outage of a number of servers running critical services.

To increase the availability of your Hardware Node, we suggest you follow the recommendations below:

- Use RAID storage for critical Container private areas. Do prefer hardware RAID, but software mirroring RAID might suit too as a last resort.
- Do not run software on the Hardware Node itself. Create special Containers where you can host necessary services such as BIND, FTPD, HTTPD, and so on. On the Hardware Node itself, you need only the SSH daemon. Preferably, the Node should accept connections from a pre-defined set of IP addresses only.
- Do not create users on the Hardware Node itself. You can create as many users as you need in Containers. Remember: compromising the Hardware Node means compromising all Containers as well.

CHAPTER 3

Operations on Containers

This chapter describes how to perform day-to-day operations on Containers.

Note: We assume that you have successfully installed, configured, and deployed your Parallels Virtuozzo Containers system. If you have not, refer to the *Parallels Virtuozzo Containers 4.7 Installation Guide*.

In This Chapter

Creating Containers

This section guides you through the process of creating a Container. We assume that you have successfully installed Parallels Virtuozzo Containers and prepared at least one OS EZ template. If you do not have any OS EZ templates prepared for creating Containers, see the *Parallels Virtuozzo Containers 4.7 Templates Management Guide* first.

Before You Begin

Before you start creating a Container, do the following:

- Check that the Hardware Node is visible on your network. You should be able to connect to/from other hosts. Otherwise, Containers will not be accessible from other servers.
- Check that you have at least one IP address per Container and the addresses belong to the same network as the Hardware Node or routing to the Containers has been set up via the Hardware Node.

To create a new Container, you need to complete the following tasks:

1. Choose an ID for the Container.
2. Choose an OS template for the Container.
3. Create the Container.

Choosing a Container ID

Every Container has a numeric ID, also known as Container ID, associated with it. The ID is a 32-bit integer number beginning with zero and unique for a given Hardware Node. When choosing an ID for a Container, follow the simple guidelines below:

- ID 0 is used for the Hardware Node itself. You cannot and should not try to create a Container with ID 0.
- The Parallels Virtuozzo Containers software reserves the IDs ranging from 0 to 100. *Do not create Containers with IDs below 101.*

The only strict requirement for a Container ID is to be unique for a particular Hardware Node. However, if you are going to have several computers running Parallels Virtuozzo Containers 4.7, we recommend assigning different Container ID ranges to them. For example, on Hardware Node 1 you create Containers within the range of IDs from 101 to 1000; on Hardware Node 2 you use the range from 1001 to 2000, and so on. This approach makes it easier to remember on which Hardware Node a Container has been created, and eliminates the possibility of Container ID conflicts when a Container is migrated from one Hardware Node to another.

Another approach to assigning Container IDs is to follow some pattern of Container IP addresses. Thus, for example, if you have a subnet with the 10.0.x.x address range, you may want to assign ID 17015 to the Container with IP address 10.0.17.15, ID 39108 to the Container with IP address 10.0.39.108, and so on. You can also think of your own patterns for assigning Container IDs depending on the configuration of your network and your specific needs.

Before you decide on a new Container ID, you may want to make sure that no Container with this ID has yet been created on the Node. The easiest way to check this is to run the following command:

```
# vzlist -a 101
Container not found
```

This output shows that Container 101 does not exist on the Node; otherwise it would be present in the list.

If you use Parallels Management Console, click on the name of your Hardware Node in the left pane and then on the **Parallels Virtuozzo Containers** item. The Management Console right pane will display the list of Containers existing on the Node.

WARNING! When deciding on a Container ID, do not use IDs that were once assigned to Containers unless you are sure that no data belonging to the old Containers remains on the Node. Otherwise, the administrator of the newly-created Container may get access to this data—that is, to the backups of the old Container, its logs, statistics, and so on.

Choosing an OS EZ Template

Before starting to create a Container, you need to choose the OS EZ template to base the Container on. You can have several OS EZ templates installed on the Node and prepared for the Container creation. Use the `vzpkg list` command to see all available OS templates:

```
# vzpkg list -O
centos-5-x86_64                      2011-04-21 23:59:44
fedora-core-13-x86_64          2011-04-11 12:45:52
```

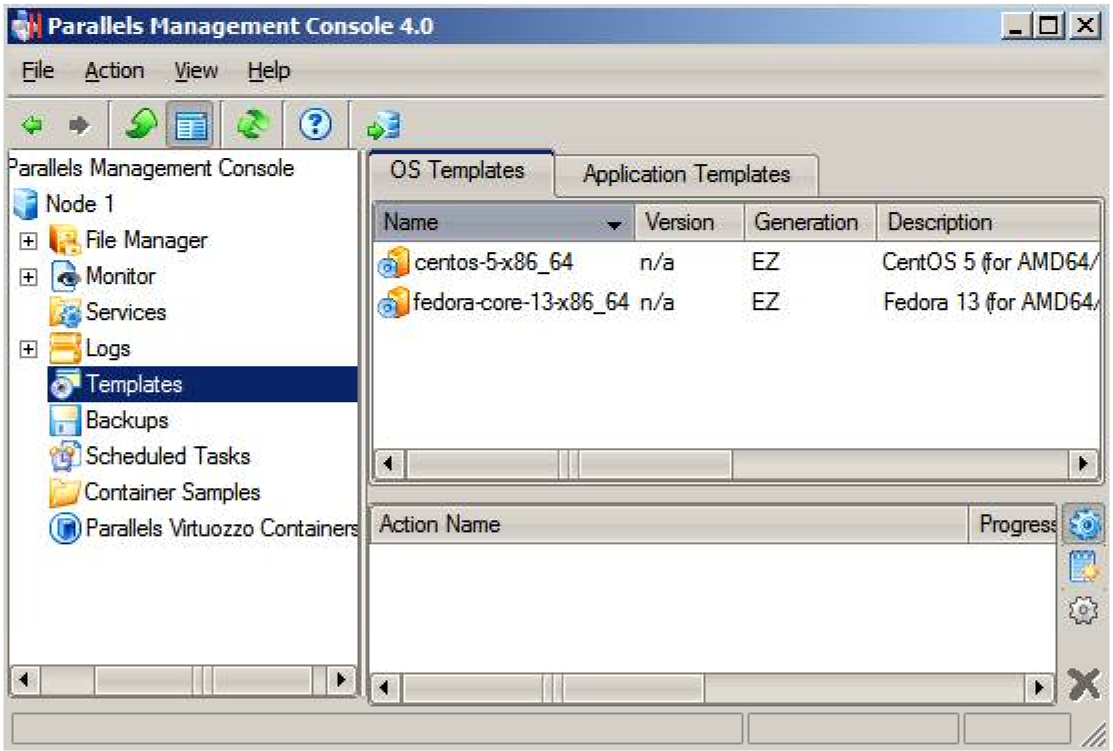
The `-O` option passed to the `vzpkg list` command allows you to list only OS EZ templates installed on the Node. As you can see, two OS templates—`centos-5-x86_64` and `fedora-core-13-x86_64`—are currently available on the server. The time displayed beyond OS EZ templates indicates when these templates were cached.

You can also use the `--with-summary` option to display brief information on the installed OS EZ templates:

```
# vzpkg list -O --with-summary
centos-5-x86_64      :CentOS 5 (for AMD64/Intel EM64T) EZ OS template
fedora-core-13-x86 :Fedora 13 (for AMD64/Intel EM64T) EZ OS template
```

For more information on the `vzpkg list` command, see the *Parallels Virtuozzo Containers 4.7 Reference Guide*.

In Parallels Management Console, you can click the **Templates** item under the corresponding Hardware Node name and then the **OS Templates** tab to see the list of the installed OS templates.



Name	Version	Generation	Description
centos-5-x86_64	n/a	EZ	CentOS 5 (for AMD64/
fedora-core-13-x86_64	n/a	EZ	Fedora 13 (for AMD64/

List of Supported Linux Distributions for Containers

The current version of Parallels Virtuozzo Containers allows you to create Containers running the following Linux distributions:

- Red Hat Enterprise Linux 5.6 and 6.x
- CentOS 5.6 and 6.x
- Debian 5.0 and 6.0
- Fedora 14
- OpenSUSE 11 with Service Packs 4
- SUSE Linux Enterprise Server 10 and 11
- Ubuntu 10.04, 10.10, and 11.04

Creating a Container

Once you choose the Container ID and OS EZ template, you can create the Container private area using the `vzctl create` command. A private area is the directory containing the VZFS symlinks, copy-on-write area, and private files of the given Container. The private area is mounted to the `/vz/root/CT_ID` directory on the Hardware Node and provides Container users with a complete Linux file system tree.

The `vzctl create` command requires only the Container ID and the name of the OS template as arguments. However, to avoid setting all the Container resource control parameters after creating the private area, you can specify a sample configuration to be used for the new Container. All sample configuration files are stored in the `/etc/vz/conf` directory and have names with the following mask: `ve-<configname>.conf-sample`. The most commonly used sample is the `ve-basic.conf-sample` file. this sample file has resource control parameters suitable for most Containers.

Thus, for example, you can create a new Container by executing the following command:

```
# vzctl create 101 --ostemplate redhat-el5-x86 --config basic
Creating Container private area (redhat-el5-x86)
Container is mounted
Postcreate action done
Container is unmounted
Container private area was created
Delete port redirection
Adding port redirection to Container(1): 4643 8443
```

This command creates a Container with ID 101, bases it on the `redhat-el5-x86` OS EZ template, and takes all necessary configuration parameters from the `ve-basic.conf-sample` sample configuration file.

If you specify neither an OS template nor a sample configuration, `vzctl` will try to take the corresponding values from the global configuration file (`/etc/vz/vz.conf`). So you can set the default values in this file using your favorite text file editor and do without specifying these parameters each time you create a new Container, for example:

```
DEF_OSTEMPLATE=".redhat-el5-x86"
CONFIGFILE="basic"
```

Now you can create a Container with ID 101 with the following command:

```
# vzctl create 101
Creating Container private area (redhat-el5-x86)
Container is mounted
Postcreate action done
Container is unmounted
Container private area was created
Delete port redirection
Adding port redirection to Container(1): 4643 8443
```

In principle, now you are ready to start your newly created Container. However, typically you need to set its network IP address, hostname, DNS server address and `root` password before starting the Container for the first time.

Configuring Containers

As a rule, the process of configuring a Container includes the following tasks:

- setting Container startup parameters
- setting Container network parameters
- setting Container user passwords
- configuring Quality of Service (Service Level) parameters

For all these tasks, the `vzctl set` command is used. Using this command for setting Container startup parameters, network parameters, and user passwords is explained later in this subsection. Service Level Management configuration topics are discussed in the **Managing Resources** chapter (p. 120).

Setting Startup Parameters

You can use the `vzctl set` command to set the `onboot` startup parameter for a Container. Setting this parameter to `yes` makes the Container automatically boot at the Hardware Node startup. For example, to enable Container 101 to automatically start when you boot the Hardware Node, you can execute the following command:

```
# vzctl set 101 --onboot yes --save
Saved parameters for Container 101
```

The `onboot` parameter will have effect on the next Hardware Node startup.

Setting Network Parameters

To make a Container accessible from the network, you need assign a valid IP address and hostname to it and configure a DNS server the Container will use. You may also wish to start the SSH daemon inside the Container. The session below illustrates setting all these network parameters for Container 101:

```
# vzctl set 101 --hostname server101.parallels.com --save
Hostname for Container set: server101.parallels.com
Saved parameters for Container 101
# vzctl set 101 --ipadd 10.0.186.1/24 --save
Adding IP addresses to the pool: 10.0.186.1
Saved parameters for Container 101
# vzctl set 101 --ipadd fe80::20c:29ff:fe01:fb08 --save
Adding IP addresses to the pool: fe80::20c:29ff:fe01:fb08
Saved parameters for Container 101
# vzctl set 101 --nameserver 192.168.1.165 --save
File resolv.conf was modified
Saved parameters for Container 101
```

These commands will configure Container 101 as follows:

- Set IPv4 address `10.0.186.1` with subnet mask `255.255.255.0` and IPv6 address `fe80::20c:29ff:fe01:fb08`.

  > **Note:** You can assign network masks to Containers operating in the `venet0` networking mode only if the `USE_VENET_MASK` parameter in the Parallels Virtuozzo Containers configuration file is set to `yes`.

- Set the hostname `server101.parallels.com`.
- Set the DNS server address to `192.168.1.165`.

The `--save` flag saves all the parameters to the Container configuration file. You can execute the above commands when the Container is running. In this case, if you do not want the applied values to persist, you can omit the `--save` option and the applied values will be valid only until the Container shutdown.

To check whether SSH is running inside the Container, you can use the `vzctl exec` command. This command allows you to execute any commands in the Container context. In Red Hat-based distributions, `sshd` is dependent on `xinetd`, so your session may look like the following:

```
# vzctl start 101
[This command starts Container 101, if it is not started yet]
# vzctl exec 101 service xinetd status
xinetd is stopped
# vzctl exec 101 service xinetd start
Starting xinetd: [  OK  ]
# vzctl exec 101 service xinetd status
xinetd is started
```

The above example assumes that Container 101 is created on the CentOS OS template. For other OS templates, consult their documentation.

For more information on running commands inside a Container from the Hardware Node, see the **Running Commands in Containers** subsection (p. 114).

Setting the root Password for Containers

Setting the `root` user password is necessary for connecting to a Container via SSH or Parallels Power Panel. By default, the `root` account is locked in a newly created Container, and you cannot log in to it. To unlock the root account, you can run the following commands on the Hardware Node:

```
# vzctl start 101
[This command starts Container 101, if it is not started yet]
# vzctl set 101 --userpasswd root:test
```

In this example, we set the root password for Container 101 to `test`. Now you can log in to the Container via SSH as root and administer it in the same way you would administer a standalone Linux server: install additional software, add users, set up services, and so on. The password will be set inside the Container in the `/etc/shadow` file in an encrypted form and will not be stored in the Container configuration file. Therefore, if you forget the password, you have to reset it. Note that `--userpasswd` does not requires the `--save` switch; the password is anyway persistently set for the Container.

While you can create users and set passwords for them using the `vzctl exec` or `vzctl set` commands, it is suggested that you delegate user management to Container administrators advising them of the Container `root` account password.

Starting, Stopping, Restarting, and Querying the Status of Containers

A Container can be started up and shut down like an ordinary server. For example, to start Container 101, you can use the following command:

```
# vzctl start 101
Starting Container ...
Container is mounted
Adding port redirection to Container(1): 4643 8443
Adding IP address(es): 10.0.186.101
Hostname for Container 101 set: test.parallels.com
Container start in progress...
```

To check the status of a Container, use the `vzctl status` command:

```
# vzctl status 101
VEID 101 exist mounted running
```

The output displays the following information:

- Whether the Container private area exists.
- Whether this private area is mounted.
- Whether the Container is running.

In our case, `vzctl` reports that Container 101 exists, its private area is mounted, and the Container is running. Alternatively, you can make use of the `vzlist` utility:

```
# vzlist 101
CTID       NPROC STATUS  IP_ADDR         HOSTNAME
 101         20 running 10.0.186.101     test.parallels.com
```

The following command is used to stop a Container:

```
# vzctl stop 101
Stopping Container ...
Container was stopped
Container is unmounted
```

`vzctl` has a two-minute timeout for the Container shutdown scripts to be executed. If the Container is not stopped in two minutes, the system forcibly kills all the processes in the Container. The Container will be stopped in any case, even if it is seriously damaged. To avoid waiting for two minutes if you know that the Container is corrupt, you can use the `--fast` switch:

```
# vzctl stop 101 --fast
Stopping Container ...
Container was stopped
Container is unmounted
```

Make sure that you do not use the `--fast` switch with healthy Containers, as the forcible killing of Container processes may be potentially dangerous.

To restart a Container, you can use the `vzctl restart` command:

```
# vzctl restart 101
Stopping Container ...
Container was stopped
Container is unmounted
Starting Container ...
Container is mounted
Adding IP address(es): 10.0.186.101
Container start in progress...
```

Note: You can also use Container names to start, stop, and restart Containers. For detailed information on Container names, see the **Setting Names for Containers** section (p. 43).

Listing Containers

Sometimes, you may want to get an overview of the Containers existing on the Hardware Node and to get additional information about them: their IP addresses, hostnames, current resource consumption, and so on. In the most general case, you can get the list of all Containers by running the following command:

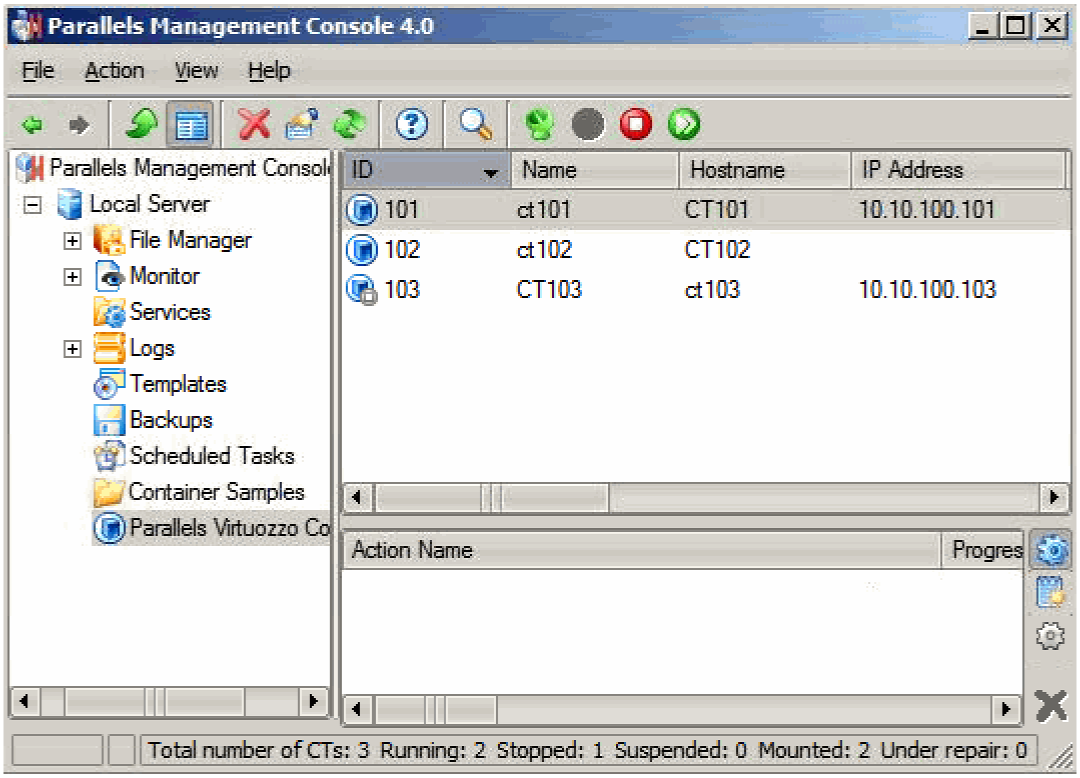
```
# vzlist -a
      CTID      NPROC STATUS  IP_ADDR         HOSTNAME
         1        135 running 10.101.60.79    localhost
       101          8 running 10.101.66.1     ct101.parallels.com
       102          7 running 10.101.66.159   ct102.parallels.com
       103          - stopped 10.101.66.103   ct103.parallels.com
```

The `-a` switch tells the `vzlist` utility to output both running and stopped Containers. By default, only running Containers are shown. The default columns inform you of the Container IDs, the number of running processes inside Containers, their status, IP addresses, and hostnames. You can customize this output as desired by using `vzlist` command line switches. For example:

```
# vzlist -o veid,diskinodes.s -s diskinodes.s
     CTID  DQINODES.S
         1     400000
       101     200000
       102     200000
```

This shows only running Containers with the information about their IDs and soft limit on disk inodes, with the list sorted by this soft limit. The full list of the `vzlist` command-line switches and output and sorting options is given in the *Parallels Virtuozzo Containers 4.7 Reference Guide*.

In Parallels Management Console, you can display the list of all Containers by clicking the **Parallels Virtuozzo Containers** item.



ID	Name	Hostname	IP Address
101	ct101	CT101	10.10.100.101
102	ct102	CT102	
103	CT103	ct103	10.10.100.103

Total number of CTs: 3 Running: 2 Stopped: 1 Suspended: 0 Mounted: 2 Under repair: 0

You can see that currently Containers 101, 102, and 103 exist on the Hardware Node. All the Container vital information (its IP address(es), hostname, statuses, etc.) is presented in the table having the following columns:

Column Name	Description
ID	The ID assigned to the Container.
Name	The name assigned to the Container. This name can be used, along with the Container ID, to perform Container-related operations on the Hardware Node.
Hostname	The hostname of the Container.
IP Address	The IP address assigned to the Container.
Status	The current status of the Container.
Resources	The circle opposite the corresponding Container reflects the current state of the resource parameters consumed by the Container: • If the resource consumption lies within 90% of the limits defined for the Container, the green circle with a white tick is displayed. It means that the Container experiences no shortage in resources required for the normal course of work. • If the Container consumes between 90% and 100% of the limits defined for it, the orange circle with a white exclamation mark is displayed. • If the Container is currently consuming 100% or more of the limits defined for it, the red circle with a white exclamation mark is displayed. A Container is allowed to consume more than 100% of its quota only in extreme situations. If you do not solve the problem in a reasonable time, applications running inside the Container may be denied some of the resources, so application crashes and other problems are most probable.
OS	The OS template the Container is based on.
Architecture	The system architecture of the Container.
Original Sample	The name of the configuration sample the Container is based on.
Description	The Container description.

To facilitate working with Containers, you can sort them by different parameters listed in the table above: their ID, type, hostname, status, IP address, and so on. Just click the column with the appropriate name to put Containers in the desired order.

Setting Names for Containers

You can assign an arbitrary name to a Container and use it, along with the Container ID, to perform Container-related operations. For example, you can start or stop a Container by specifying the Container name instead of its ID.

You can assign names to Containers using the `--name` option of the `vzctl set` command. For example, to set the `computer1` name for Container 101, run this command:

```
# vzctl set 101 --name computer1 --save
Name computer1 assigned
Saved parameters for Container 101
```

You can also set a name for Container 101 by editing its configuration file:

1 Open the configuration file of Container 101 (`/etc/vz/conf/101.conf`) for editing, and add the following string to the file:

```
NAME="computer1"
```

2 In the `/etc/vz/names` directory on the Node, create a symbolic link with the name of `computer1` pointing to the Container configuration file. For example:

```
# ln --symbolic /etc/vz/conf/101.conf /etc/vz/names/computer1
```

When specifying names for Containers, keep in mind the following:

- Names may contain the following symbols: `a-z`, `A-Z`, `0-9`, underscores (_), dashes (-), spaces, the symbols from the ASCII character table with their code in the 128-255 range, and all the national alphabets included in the Unicode code space.
- Container names cannot consist of digits only. Otherwise, there would be no way to distinguish them from Container IDs.
- If it contains one or more spaces, the Container name must be put in single or double quotes.

Once you assign the `computer1` name to Container 101, you can start using it instead of ID 101 to perform Container-related operations. For example:

- You can stop Container 101 with the following command:

```
# vzctl stop computer1
Stopping Container ...
Container was stopped
Container is unmounted
```

- You can start Container 101 anew by running the following command:

```
# vzctl start computer1
Starting Container ...
...
```

You can find out what name is assigned to Container 101 in one of the following ways:

- Using the `vzlist` utility:

```
# vzlist -o name 101
NAME
computer1
```

- Checking the `NAME` parameter in the Container configuration file (`/etc/vz/conf/101.conf`):

```
# grep NAME /etc/vz/conf/101.conf
NAME="computer1"
```

- Checking the `NAME` parameter in the `/etc/vz/names/computer1` file which is a symlink to the Container configuration file:

```
# grep NAME /etc/vz/names/computer1
NAME="computer1"
```

You can also use Parallels Management Console to set names for Containers:

1. Select the **Parallels Virtuozzo Containers** item under the corresponding Hardware Node, right-click the Container, and choose **Properties**.
2. On the **General** tab of the displayed window, enter a name in the **Name** field.
3. Click **OK**.

Storing Extended Information on Containers

Sometimes, it may be difficult to remember the information on certain Containers. The probability of this increases together with the number of Containers and with the time elapsed since the Container creation. Parallels Virtuozzo Containers allows you to set the description for Containers and view it later on, if required. The description can be any text containing any Container-related information; for example, you can include the following in the Container description:

- the owner of the Container
- the purpose of the Container
- the summary description of the Container

Let us assume that you are asked to create a Container for a Mr. Johnson who is going to use it for hosting the MySQL server. So, you create Container 101 and then execute the following command on the Hardware Node:

```
# vzctl set 101 --description "Container 101
> owner—Mr. Johnson
> purpose—hosting the MySQL server" --save
Saved parameters for Container 101
```

This command saves the following information related to the Container: its ID, owner, and the purpose of its creation. At any time, you can display this information by running the following command:

```
# vzlist -o description 101
DESCRIPTION
Container 101
owner—Mr. Johnson
purpose—hosting the MySQL server
```

You can also view the Container description by checking the `DESCRIPTION` parameter of the Container configuration file (`/etc/vz/conf/101.conf`). However, the data stored in this file are more suitable for parsing by the `vzlist` command rather than for viewing by a human because all symbols in the `DESCRIPTION` field except the alphanumerical ones ('a-z', 'A-Z', and '0-9'), underscores ('_'), and dots ('.') are transformed to the corresponding hex character code.

When working with Container descriptions, keep in mind the following:

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

To provide a description for a Container in Parallels Management Console, do the following:

1 Select the **Parallels Virtuozzo Containers** item under the corresponding Hardware Node, right-click the Container, and choose **Properties**.

2 On the **General** tab of the displayed window, type the necessary information in the **Description** field.

3 Click **OK**.

Migrating Containers

A Hardware Node is the system with higher availability requirements in comparison with a typical Linux system. If you are running your company mail server, file server, and web server in different Containers on one and the same Hardware Node, then shutting it down for hardware upgrade will make all these services unavailable at once. To facilitate hardware upgrades and load balancing between Hardware Nodes, Parallels Virtuozzo Containers provides you with the ability to migrate Containers from one physical box to another.

Migrating Containers is possible if Parallels Virtuozzo Containers 4.7 is installed on two or more Hardware Nodes. You can choose one of the two ways to migrate a Container:

- Migrating a Container using the standard migration technology. In this case, there is a short downtime needed to stop and start the Container during its migration from the Source Node to the Destination Node.
- Migrating a Container using the zero downtime migration technology. In this case, the 'stop' and 'start' operations are not performed and the migrated Container is restored on the Destination Node in the same state as it was at the beginning of the migration. This greatly reduces the migration time and puts it on the same footing as the delay caused by a short interruption in the network connectivity.

Both ways are described in the following subsections in detail.

Note: Containers created under the Parallels Virtuozzo Containers x86 version can be migrated to Hardware Nodes running the x86-64 version of Parallels Virtuozzo Containers, but not vice versa.

Standard Migration

The standard migration procedure allows you to move both stopped and running Containers. Migrating a stopped Container includes copying all Container private files from one Node to another and does not differ from copying a number of files from one server to another over the network. The migration procedure of a running Container, in turn, is a bit more complicated and can be described as follows:

1. After initiating the migration process, all Container private data is copied to the Destination Node. During this time, the Container on the Source Node continues running.
2. The Container on the Source Node is stopped.
3. The Container private data copied to the Destination Node is compared with that on the Source Node, and if any files were changed during the first migration step, they are copied to the Destination Node again and rewrite the outdated versions.
4. The Container on the Destination Node is started.

There is a short downtime needed to stop the Container on the Source Node, copy the Container private data changes to the Destination Node, and start the Container on the Destination Node. However, this time is very short and does not usually exceed one minute.

Note: Before the migration, it may be necessary to detach the Container from its caches. For more information on cached files, see the **Cleaning Up Containers** subsection (p. 160).

The following session moves Container 101 from the current Hardware Node to a new Node named `ts7.parallels.com`:

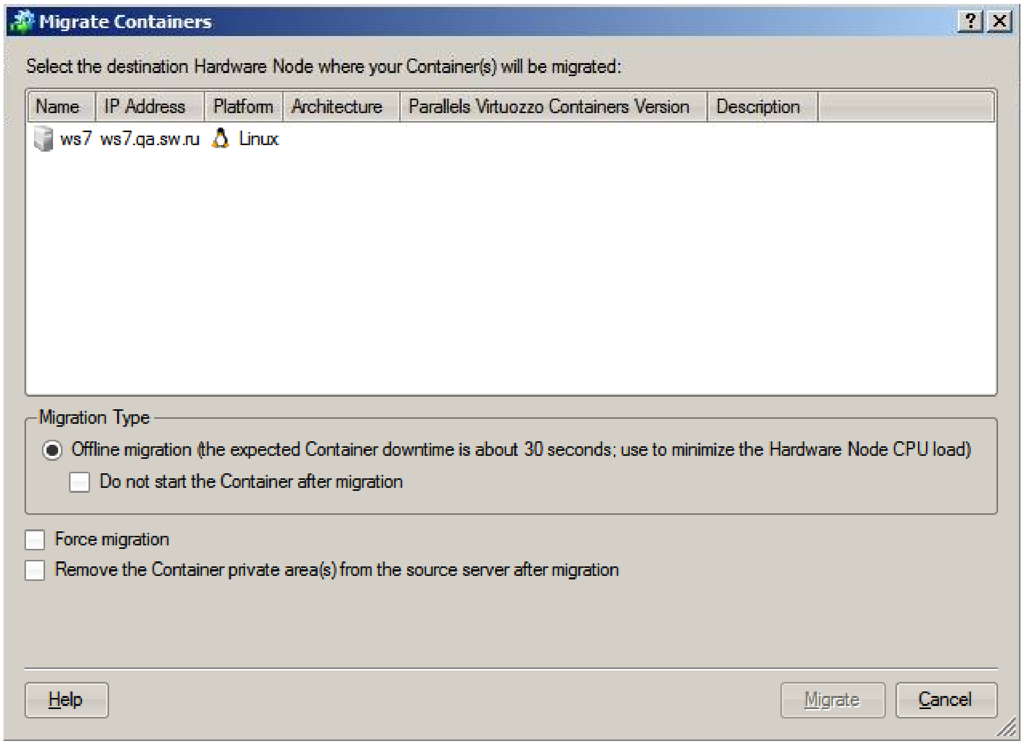
```
# vzmigrate ts7.parallels.com 101
root@ts7.parallels.com's password:
vzmsrc: Connection to destination Hardware Node (ts7.parallels.com) \
is successfully established
vzmsrc: Moving/copying Container#101 -> Container#101, [], [] ...
vzmsrc: Container migrating mode : first stage sync, with tracking, \
second stage sync, with Container stopping
vzmsrc: Syncing private area of Container#101 [/vz/private/101] ...
 / 100% |*****************************|
vzmsrc: done
vzmsrc: Stopping Container#101 ...
vzmsrc: done
vzmsrc: Fast syncing private area of Container#101 [/vz/private/101] ...
 / 100% |*****************************|
vzmsrc: done
vzmsrc: DST: Starting Container#101 ...
vzmsrc: DST: done
vzmsrc: Successfully completed
```

You can specify more than one Container ID simultaneously. In this case, all specified Containers will be moved to a new Hardware Node one by one.

Important! For the command to be successful, a direct SSH connection (on port 22) must be allowed between the Source and Destination Nodes.

By default, once the migration is complete, the Container private area and configuration file on the Source Node are renamed by receiving the `.migrated` suffix. However, if you want the Container private area on the Source Node to be removed after the successful Container migration, you can override the default `vzmigrate` behavior by changing the value of the `REMOVEMIGRATED` variable in the global configuration file (`/etc/vz/vz.conf`) to `yes` or by using the `-r yes` switch of the `vzmigrate` command.

To migrate one or more Containers using Parallels Management Console, select these Containers from the list in the right pane after clicking the **Parallels Virtuozzo Containers** item in the left pane. Then right-click the selection, and choose **Tasks** > **Migrate to Another Hardware Node**. Note that the Destination Node must be already registered in Management Console; otherwise, the migration option will not be available. The **Migrate Containers** window appears.



Select the destination Hardware Node where your Container(s) will be migrated:

Name	IP Address	Platform	Architecture	Parallels Virtuozzo Containers Version	Description
ws7	ws7.qa.sw.ru	Linux			

Migration Type

● Offline migration (the expected Container downtime is about 30 seconds; use to minimize the Hardware Node CPU load)

□ Do not start the Container after migration

□ Force migration

□ Remove the Container private area(s) from the source server after migration

In this window, do the following:

- Select the Destination Node where you want to move the Container.
- Make sure that the **Offline migration** radio button is selected. This option is used to migrate Containers by means of the standard migration technology.

You can also specify the following options for the Container:

- The **Do not start the Container after migration** check box, if selected, prevents the migrated Container from starting on the Destination Node after its successful migration. This option does not have any effect if the Container was not running on the Source Node.
- The **Force migration** check box, if selected, forces the Container migration even if the templates necessary for the Container correct operation are not installed on the Destination Node. However, it will be impossible to start such a Container after the migration in case of the absence of the needed templates.
- Select the **Remove the Container private area(s) from the source server after migration** check box to delete the Container private area from the Source Node after the Container successful migration.

When you are ready, click the **Migrate** button.

Zero-Downtime Migration

The `vzmigrate` utility allows you to migrate Containers from one Hardware Node to another with zero downtime. The zero-downtime migration technology has the following main advantages as compared to the standard one:

- The process of migrating a Container to another Node is transparent for you and the Container applications and network connections. That means that no modifications of system characteristics and operational procedures inside the Container are performed on the Source and Destination Nodes.
- The Container migration time is greatly reduced. In fact, the migration eliminates the service outage or interruption for Container end users.
- The Container is restored on the Destination Node in the same state as it was at the beginning of the migration.
- You can move the Containers running a number of applications that you do not want to be rebooted during the migration.

Note: Zero-downtime migration cannot be performed on Containers having one or several opened sessions established with the `vzctl enter` *`CT_ID`* command.

Before performing zero-downtime migration, it is recommended to synchronize the system time on the Source and Destination Nodes, for example, by means of NTP (http://www.ntp.org). The reason for this recommendation is that some processes running in the Container might rely on the system time being monotonic and thus might behave unpredictably if they see an abrupt step forward or backward in the time once they find themselves on the new Node with different system clock parameters.

In the current version of Parallels Virtuozzo Containers, you can use the following types of zero-downtime migration for migrating a Container:

- *Simple online migration*. In this case, the Container is dumped at the beginning of the migration—that is, all Container private data including the state of all running processes are saved to an image file. This image file is then transferred to the Destination Node where it is undumped.
- *Lazy online migration*. Using this type of online migration allows you to decrease the size of the dumped image file storing all Container private data and transferred to the Destination Node by leaving the main amount of memory in a locked state on the Source Node and swapping this memory from the Source Node on demand. Thus, the migrated Container can be started before the whole memory is transferred to the Destination Node, which drastically reduces the service delay of the corresponding Container. When a process tries to access a page of memory that has not yet been migrated, the request is intercepted and redirected to the Source Node where this page is stored.
- *Iterative online migration*. In this case, the main amount of Container memory is transferred to the Destination Node before the Container is dumped and saved to an image file. Using this type of online migration allows you to attain the smallest service delay.

- *Iterative + lazy online migration*. This type of online migration combines the techniques used in both the *lazy* and *iterative* migration types when some part of Container memory is transferred to the Destination Node before the Container is dumped and the rest is transported after the Container has been successfully undumped on the Node.

To migrate a Container by using the zero downtime migration technology, use the `--online` option of the `vzmigrate` utility. By default, the *iterative online migration* type is used to move a Container from one Hardware Node to another. For example, you can migrate Container 101 from the current Hardware Node to the Destination Node named `my_node.com` by executing the following command:

> **Note:** If the CPU capabilities of the Source Node exceed those of the Destination Node (for example, you migrate from a Source Node running the Core 2 Duo processor to a Destination Node running the Pentium 4 processor), the migration may fail and you will be presented with the corresponding warning message. However, if you are sure that the CPU power of the Destination Node is sufficient to start and run the Container being migrated, you can use the `-f` option to force the migration process.

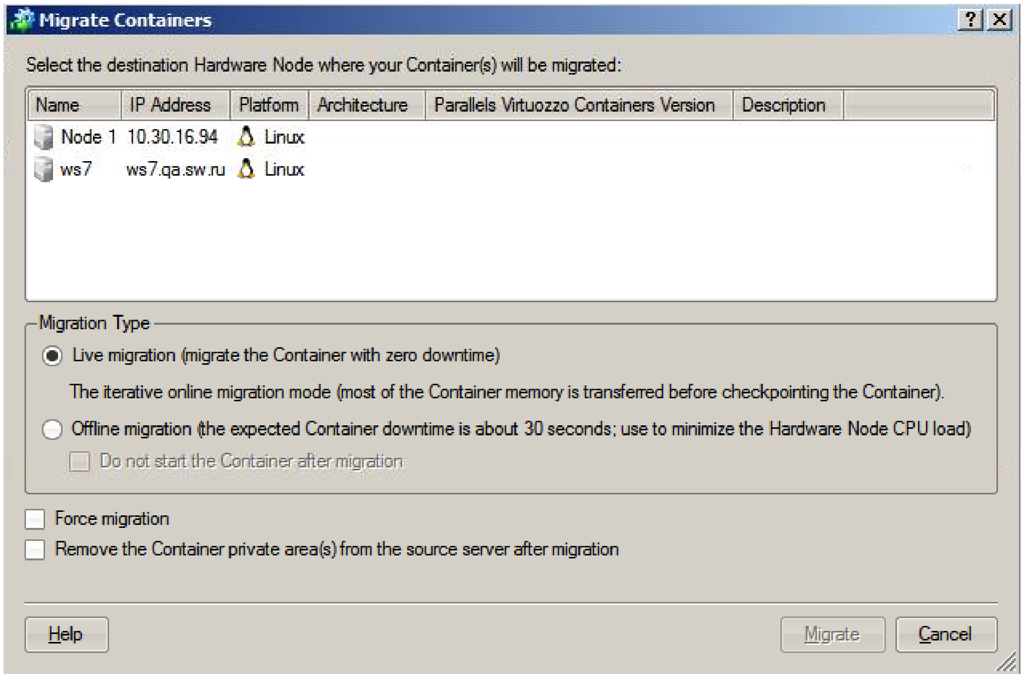
```
# vzmigrate --online --require-realtime my_node.com 101
Enter password:
Connection to destination Hardware Node (192.168.1.57) \
is successfully established
Moving/copying Container#101 -> Container#101, [], [] ...
Syncing private area '/vz/private/101'
 - 100% |****************************************
done
Suspending Container#101 ...
done
Dumping Container#101 ...
done
...
Migration completed
```

The `--require--realtime` option tells `vzmigrate` to move the Container by using the *iterative online migration* type only. So, if this migration type cannot be carried out, the command will fail and exit. If the `--require--realtime` option is omitted and the command fails, `vzmigrate` will try to move the Container by means of the *simple online migration*. You can specify more than one Container ID simultaneously. In this case, all specified Containers will be moved to a new Hardware Node one by one.

If you want to use another migration type for migrating Containers, you need to pass the following options to `vzmigrate`:

- Specify the `--noiter` option to migrate a Container using the *simple online migration* type.
- Specify the `--noiter` and `--lazy` options to migrate a Container using the *lazy online migration* type.
- Specify the `--lazy` option to migrate a Container using the *iterative + lazy online migrate* type.

To migrate one or more Containers in Parallels Management Console, select these Containers from the list in the right pane after selecting the **Parallels Virtuozzo Containers** item in the left pane. Then right-click the selection, and choose **Tasks** > **Migrate to Another Hardware Node**. Note that the Destination Node must be already registered in Parallels Management Console; otherwise, the migration option will not be available. The **Migrate Containers** window appears.



Select the destination Hardware Node where your Container(s) will be migrated:

Name	IP Address	Platform	Architecture	Parallels Virtuozzo Containers Version	Description
Node 1	10.30.16.94	Linux			
ws7	ws7.qa.sw.ru	Linux			

Migration Type

● Live migration (migrate the Container with zero downtime)

The iterative online migration mode (most of the Container memory is transferred before checkpointing the Container).

○ Offline migration (the expected Container downtime is about 30 seconds; use to minimize the Hardware Node CPU load)

□ Do not start the Container after migration

□ Force migration

□ Remove the Container private area(s) from the source server after migration

In this window, do the following:

- Select the Destination Node where you want to move the Container.
- Select the **Live migration** radio button used to migrate Containers by means of the zero downtime migration technology. The Container will be migrated using the iterative online migration type.

You can also specify the following options for the Container:

- The **Force migration** check box, if selected, forces the Container migration even if the templates necessary for the Container correct operation are not installed on the Destination Node. Keep in mind that you will not be able to start such a Container on the Destination Server.
- Select the **Remove the Container private area(s) from the source server after migration** check box to delete the Container private area from the Source Node after the Container successful migration.

When you are ready, click the **Migrate** button.

Migrating Containers Based on Standard Templates

If you have a Hardware Node running an old version of Parallels Virtuozzo Container (for example, Virtuozzo 3.0), you probably have Containers that are based on standard OS templates and use one or more standard application templates. To migrate such Containers to a Hardware Node running Parallels Virtuozzo Containers 4.7, you need to complete the following tasks:

1. Migrate the OS standard template the Container is based on and all standard applications templates the Container uses from the Source Node (that is, the Node where the template is installed) to the Destination Node (that is, the Node with Parallels Virtuozzo Containers 4.7 where you plan to migrate Container 101).
2. Migrate the Container from the Source Node to the Destination Node.

The following example shows you how to migrate Container 101. This example assumes the following:

- Container 101 is based on the `redhat-as4` OS standard template.
- The `postgresql-as4` standard application template is applied to Container 101.
- The Destination Node has the IP address of 192.168.0.197.

Migrating Standard Templates

In the first step, you need to migrate the OS and application standard templates used by Container 101 from the Source to the Destination Node. To do this, execute the following command on the Source Node:

```
# vzmtemplate 192.168.0.197 redhat-as4 postgresql-as4
root@192.168.0.197's password:
Connection to Destination Node (192.168.0.197) is successfully established
Copying template "redhat-as4"
...
```

When executed, the `vzmtemplate` utility tries to connect to the Destination Node with IP address 192.168.0.197 and copy the specified templates there. By default, `vzmtemplate` logs in to the Destination Node as `root` and asks you for the password of this user. However, you can make the utility use other credentials to log in to the Destination Node. To do this, indicate the desired user name with the `@` symbol before the IP address in the command above (for example, `user1@192.168.0.123`). Keep in mind that the specified user must have the `root` privileges; otherwise, the command will fail.

To check that the `redhat-as4` and `postgresql-as4` templates have been successfully copied to the Destination Node, run the following command on this Node:

```
# vzpkg list
redhat-as4           20100918
postgresql-as4        20100918
```

As you can see, both templates are now available on the Destination Node.

Migrating the Container

Now that you have copied the standard templates, you can migrate Container 101. To do this, run the following command on the Source Node:

```
# vzmigrate 192.168.0.197 101
root@192.168.0.197's password:
vzmsrc: Connection to destination Hardware Node (192.168.0.197) is successfully
established
vzmsrc: Moving/copying Container#101 -> Container#101, [], [] ...
...
```

Migrating the Container may take some time; please wait for the command to complete.

Configuring Non-Root Access for Migrating Containers

By default, you need to run the `pmigrate` utility as the `root` user to migrate Containers. You can, however, configure your system so that `pmigrate` can be executed under another account. This process involves editing the `/etc/sudoers` file on the Node.

Let us assume that you want to run the `pmigrate` utility under the `pmigr_user` account. To do this:

1. Open the `/etc/sudoers` file for editing.
2. Locate the `Defaults requiretty` string in the file, and comment it:

```
# Defaults requiretty
```

3. Locate the following string in the file

```
root    ALL=(ALL)      ALL
```

   and replace `root` with the `pmigr_user` name:

```
pmigr_user   ALL=(ALL)      ALL
```

4. Save the file.

Moving Containers Within the Hardware Node

The `vzmlocal` utility allows you to move Containers within your Node. Moving a Container within one and the same Node consists in changing the Container ID and its private area and root paths. You can use `vzmlocal` to change the ID of the corresponding Container only or to additionally modify its private area and root path.

Let us assume that you want to change the ID of your Container from 101 to 111 and modify its private area and root paths from `/vz/private/101` to `/vz/private/my_dir` and from `/vz/root/101` to `/vz/root/ct111`, respectively. To do this, execute the following command on the Node:

```
# vzmlocal 101:111:/vz/private/my_dir:/vz/root/ct111
Moving/copying Container#101 -> Container#111,
[/vz/private/my_dir], [/vz/root/ct111] ...
...
Successfully completed
```

To check if Container 101 has been successfully moved to Container 111, you can use the following commands:

```
# vzlist -a
CTID      NPROC STATUS  IP_ADDR         HOSTNAME
   1         43 running 10.0.10.1       localhost
 111         - stopped 10.0.10.101      myContainer
# ls /vz/private
1  my_dir
# ls /vz/root
1  ct111
```

The commands output shows that the ID of Container 101 has been changed to 111, its private area is now located in the `/vz/private/my_dir` directory on the Node, and the path to its root directory is `/vz/root/ct111`.

Notes:

1. You can use the `vzmlocal` utility to move several Containers simultaneously.

2. You can run the `vzmlocal` utility on both running and stopped Containers.

In Parallels Management Console, you can move Containers within a Hardware Node using the **Move Container** wizard. To invoke the wizard, select the **Parallels Virtuozzo Containers** item under the corresponding Hardware Node name, right-click the Container you want to change the ID of, and choose **Tasks** > **Move Container**. The wizard asks you to complete a number of tasks:

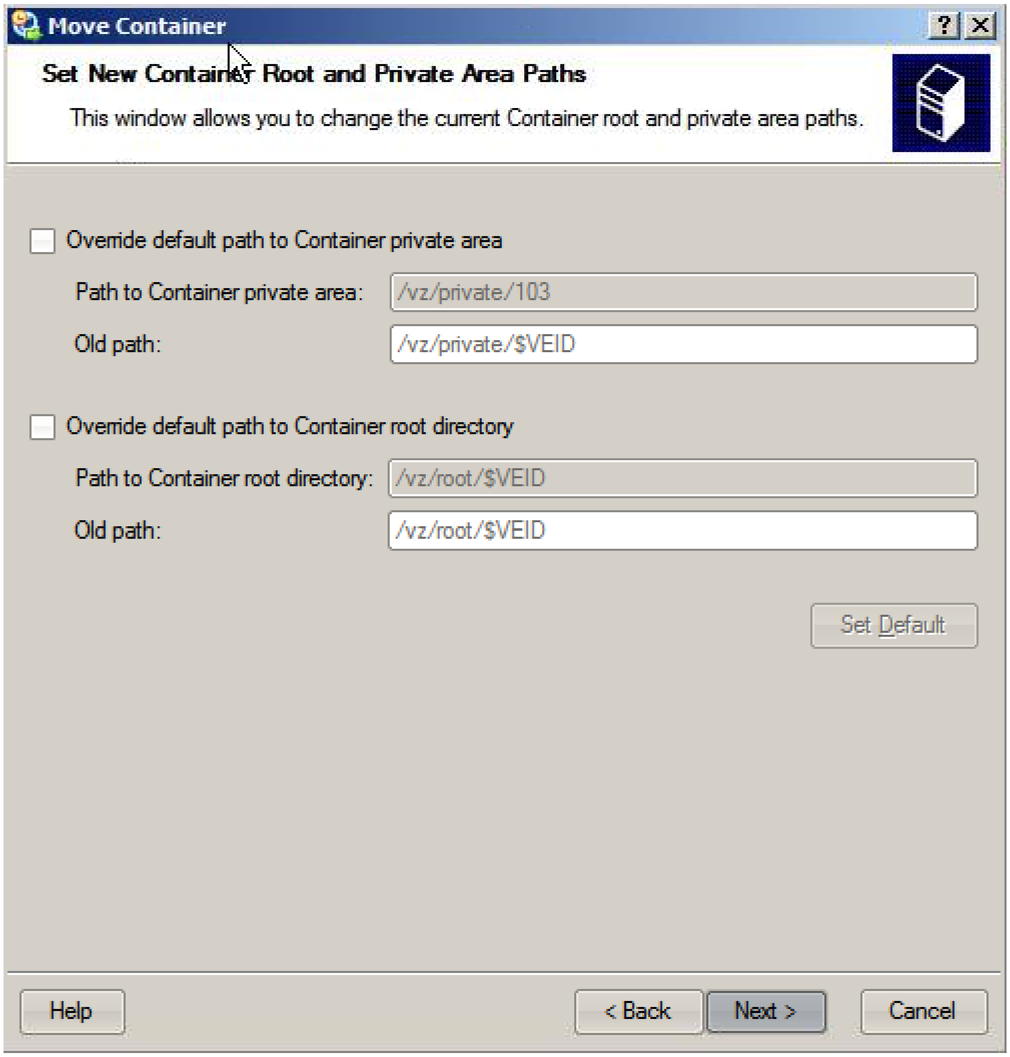
1 In the first step, you need to choose between two options:

- The first option (**Change Container ID**) lets you specify a new ID for the Container in addition to specifying its new root and private area paths. Note that if you choose this option, you will not be able to preserve the old ID for th Container.

- The second option (**Change Container location on Hardware Node**) allows you to specify the new root and private area paths without changing the Container ID.

2 If you choose the first option, you have to specify a new ID for the corresponding Container in the second step of the wizard. Note that the old ID for the Container will be lost and all Container private data will be transferred to the `/vz/private/<new_CT_ID>` directory where `<new_CT_ID>` denotes the new ID assigned to the Container (for example, `/vz/private/111` for Container 111).

3 Next, you are presented with the **Set New Container Root and Private Area Paths** window.



Set New Container Root and Private Area Paths

This window allows you to change the current Container root and private area paths.

☐ Override default path to Container private area

Path to Container private area: /vz/private/103

Old path: /vz/private/$VEID

☐ Override default path to Container root directory

Path to Container root directory: /vz/root/$VEID

Old path: /vz/root/$VEID

This window is displayed in one of the following cases:

- You selected the **Change Container ID** check box in the first step of the wizard, specified a new ID for the Container, and clicked **Next** in the **Specify New Container ID** window. In this case, the wizard will offer you to use the default paths, but will leave you the possibility to alter these paths. To configure a path, select the corresponding check box, and type the path in the field below the check box. If have made some changes to the default paths and want to revert to these paths, click the **Set Default** button.
- You selected the **Change Container location on Hardware Node** check box and clicked **Next** in the first step of the wizard. In this case, you can do the following:

  - Manually enter the new private and root paths for the Container.

  - Click the **Set Default** button to display and use the paths offered by the wizard.

4 In the last step of the **Move Container** wizard, you can review the settings made by you in the previous steps. Click the **Finish** button to start moving the Container. This may take some time.

Copying Containers Within the Hardware Node

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

In Parallels Virtuozzo Containers-based systems, you can use the `vzmlocal` utility to copy a Container within the given Hardware Node. For example, you can issue the following command to create Container 111 and make it be a complete copy of Container 101:

Note: You can clone both running and stopped Containers.

```
# vzmlocal -C 101:111
Moving/copying Container#101 -> Container#111, [], [] ...
Syncing private area '/vz/private/101'->'/vz/private/111'
...
Successfully completed
# vzlist -a
CTID      NPROC STATUS  IP_ADDR         HOSTNAME
  1         42 running 10.0.10.1       localhost
101         10 running 10.0.10.101         Container115
111          - stopped 10.0.10.115         Container115
```

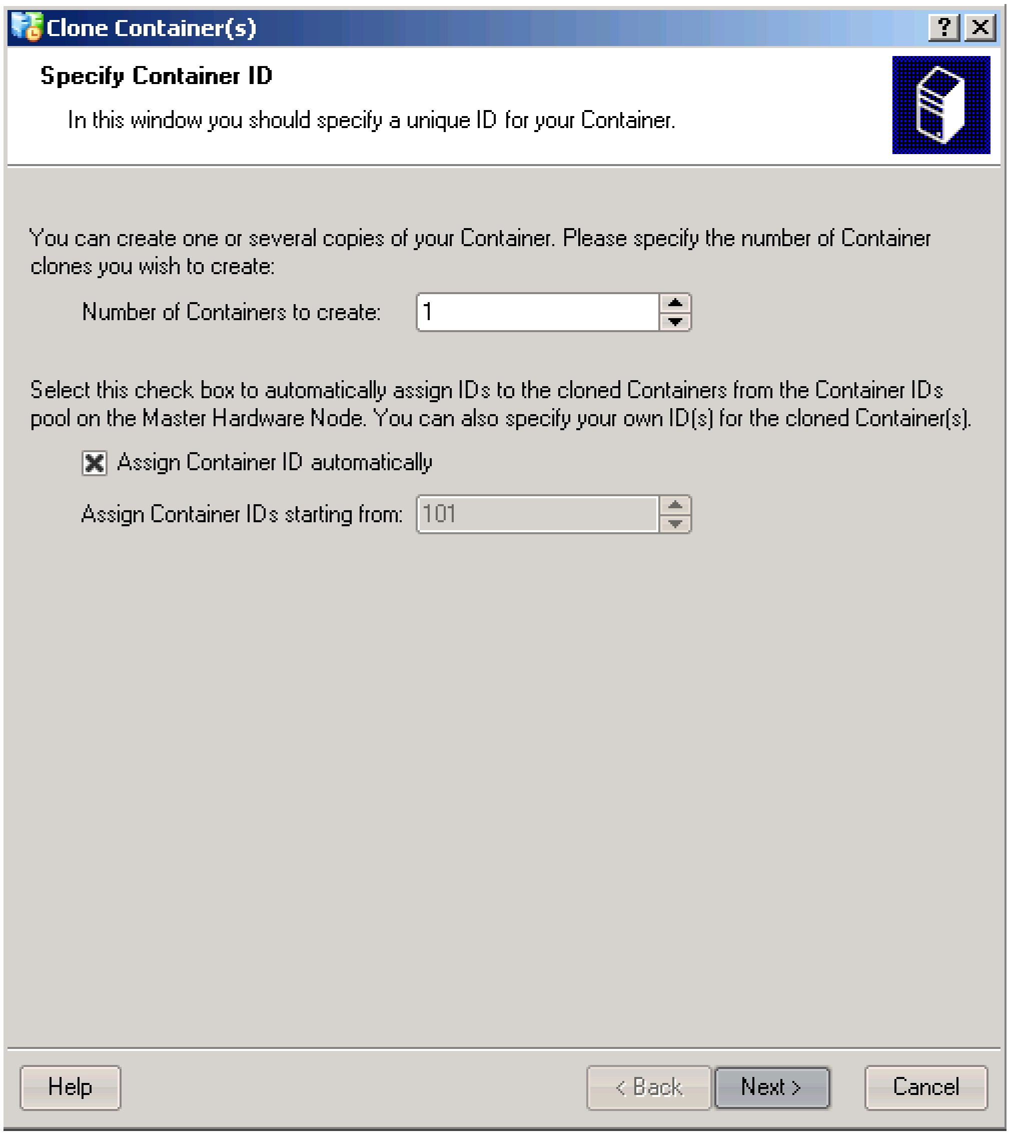
As you can see from the example above, a clone of Container 101 (i.e. Container 111) has been successfully created. However, before starting to use Container 111, you need to set another IP address and another hostname for this Container which are currently identical to those of Container 101. Please consult the **Configuring Containers** section (p. 35) to learn how you can do it.

The `vzmlocal` utility also enables you to override the default private area and root paths of the destination Container which, by default, are set to `/vz/private/<dest_CTID>` and `/vz/root/<dest_CTID>`, respectively (where `<dest_CTID>` denotes the ID of the resulting Container). In the case of Container 111, these paths are `/vz/private/111` and `/vz/root/111`. To define different private area and root paths for Container 111, you can execute the following command:

```
# vzmlocal -C 101:111:/vz/private/dir_111/:/vz/root/ct111
Moving/copying Container#101 -> Container#111, [], [] ...
Syncing private area '/vz/private/101'->'/vz/private/dir_111'
...
Successfully completed
# ls /vz/private
1  101 dir_111
# ls /vz/root
1  101 ct111
```

To clone a Container in Parallels Management Console, click **Parallels Virtuozzo Containers** under the name of the corresponding Hardware Node, right-click the Container you want to clone, and choose **Tasks** > **Clone Container(s)**. The **Clone Container** wizard will guide you through the process of cloning the Container:

1 First, you need to specify the number of Container clones to create and the starting Container ID.



Clone Container(s)

Specify Container ID

In this window you should specify a unique ID for your Container.

You can create one or several copies of your Container. Please specify the number of Container clones you wish to create:

Number of Containers to create: 1

Select this check box to automatically assign IDs to the cloned Containers from the Container IDs pool on the Master Hardware Node. You can also specify your own ID(s) for the cloned Container(s).

☒ Assign Container ID automatically

Assign Container IDs starting from: 101

Specify the number of clones to create in the **Number of Containers to create** field. By default, one Container clone is created.

Similarly to creating new Containers, the **Clone Container** wizard allows the simultaneous creation of several Container clones with IDs in a continuous series only. The default starting Container ID, which is automatically offered, is the first unoccupied ID starting from 101. For example, if you already have Containers with IDs from 101 through 105 and 107, the ID of 106 will be offered by default. And if you are creating only one Container clone, you can safely accept this number. Or you can specify any other number, and the system will check up if the ID is unoccupied. However, if you are going to create a number of Container clones, it is recommended to decide on an unoccupied ID series in advance.

2. In the second step, you are asked to specify a new name and a new hostname for the resulting Container. Type an arbitrary name you consider suitable for the Container in the **Name** field and indicate its hostname, if necessary, in the **Hostname** field.
3. In the **Assign Network Settings to Containers** window, you can view and configure the virtual network adapters that will be available inside the Container clone. Detailed information on all network parameters and on the way to manage them is provided in the **Configuring Virtual Adapter Parameters** subsection.
4. In the next step, you can change the path to the private area and root directory of the Container clone by selecting the corresponding **Override** check boxes and entering the desired paths in the fields provided.
5. The last window lets you review the parameters provided by you in the previous steps. You can also select the **Start the cloned Container after its creation** check box to immediately start the Container after its successful cloning. Click **Finish** to start the copying process.

Parallels Management Console also allows you to create several copies of a Container at once. To do this, right-click the Containers to clone in the right pane, choose **Tasks** > **Clone Container(s)**, and in the displayed window, provide the necessary information for the cloned Containers.

Backing Up and Restoring Containers

A regular backing up of the existing Containers is essential for any Hardware Node reliability. Any Container is defined by its private area, configuration files, action scripts, and quota information. Parallels Virtuozzo Containers allows to back up all these components. Each backup file can be of one of the following 3 types:

- A full backup containing all Container data. This kind of backup is the most time-consuming, space-intensive, and the least flexible one. However, full backups are the quickest to restore.
- An incremental backup containing only the files changed since the last full, differential, or incremental backup. Incremental backups record only the changes since the last Container backup (either full, differential, or incremental) and, therefore, are less in size and take less time to complete than the full and differential backups.
- A differential backup containing only the files changed since the last full backup. This kind of backup does not take into account available incremental and differential backup archives and always backs up all the files modified since the last full backup.

Using vzabackup/vzarestore Utilities

In Parallels Virtuozzo Containers 4.7, you can use the `vzabackup` and `vzarestore` utilities to back up and restore Containers. These utilities can be run on virtually every Node in your network, including

- the Source Node where the Container to be backed up is residing
- the Backup Node—a special Node with Parallels Virtuozzo Containers intended for storing Container backups
- any other physical server with Parallels Virtuozzo Containers

To successfully run the `vzabackup` and `vzarestore` utilities, make sure of the following:

1 A Node where you plan to run the utilities has the `vzabackup` package installed. You can find the `vzabackup` package in the `/virtuozzo/RPMS` directory of your Parallels Virtuozzo Containers distribution and install it using the `rpm -i` command.

2 A network connection can be established between the Source and Backup Nodes.

3 Forward and reverse DNS lookups are correctly configured for both the Source and Backup Nodes.

`vzabackup` is used to back up Containers. Let us assume the following:

- You want to create a full backup of Container 101 residing on the Source Node with the IP address of `192.168.0.15`.
- You can access the Source Node using the `source_root` user name and the `1234qwer` password.
- The backup will be created with the high level of compression.
- The backup will be stored on the Backup Node that you can access using the IP address of `192.168.200.200`, the `backup_root` user name, and the `1qaz2wsx` password.
- You want to exclude the `/tmp` directory in Container 101 from the backup.
- You want to set the following description for the resulting backup archive: `The MySQL database—latest changes.`

To create a backup with the aforementioned parameters, you can execute the following command on any Hardware Node with `vzabackup` installed and having the network connectivity to the Source and Backup Nodes:

```
# vzabackup -D "The MySQL database—latest changes." -C3 \
             --storage backup_root:1qaz2wsx@192.168.200.200 \
             source_root:1234qwer@192.168.0.15 -e 101 \
             --exclude-files /tmp
Starting backup operation for node '192.168.0.15'...
* Operation with the Container ct101 is started
* Backing up environment ct101 locally
* Checking parameters
* Dumping quota
* Creating backup 3dec6e78-2f3e-7c4a-a969-f5b27e188783/20101013154047
* Adjusting backup type (full)
* Backup storage: receiving backup file
* Preparing for backup operation
* Backing up private area
100% |**************************************************************|
* Sending private backup data
* Backup storage: storing private backup data
* Backup storage: filling resultant backup info
* Operation with the Container ct101 is finished successfully.
Backup operation for node '192.168.0.15' was finished successfully.
```

Once the command is complete, you can find the created backup archive in the backup directory on the Backup Node. By default, this directory is `/vz/backups`. Later on, you can restore the Container backup from this directory.

You can specify any number of Hardware Nodes IP addresses in the command line. You can also perform an incremental or a differential backup by additionally specifying the `-I` or `--Tdiff` option, respectively. If you indicate the `-I` or `--Tdiff` option, and the utility cannot find the corresponding full backup, a full backup is created. If you do not specify the Backup Node, the backup will be put to the backup directory on your local Node. Detailed information on all options that can be passed to the `vzabackup` utility is given in the *Parallels Virtuozzo Containers 4.7 Reference Guide*.

To restore Containers from their backups, you can use the `vzarestore` utility. However, before starting to restore Containers, you may want to view the information about your backups. For example, to examine the backups stored on the Backup Node with IP address `192.168.200.200`, you can run the following command on this Node (or on any other Node where the `vzabackup` package is installed):

```
# vzarestore --list --storage backup_root:1qaz2wsx@192.168.200.200
Show existing backups...
CTID Title       Creation date/time      Type  Size
101  ct101       2010-02-11T111507+0004  full  8.79 Mb
...
```

If you are running `vzarestore` on the Backup Node itself, you can omit the `--storage` option.

To restore Container 101, you can run this command on the Source Node:

```
# vzarestore 101 --storage backup_root:1qaz2wsx@192.168.200.200
Restore container: Container101 from 1361ac21-4cae-4981-...
Container ct101 was restored successfully.
```

This command restores the latest backup of Container 101 stored on the Backup Node with IP address `192.168.200.200` to the Node where you have run the command (in our case, on the Source Node). If you want to restore a specific (not the latest) Container backup, use the `-b` option and specify the ID of the backup instead of the Container ID. You can find out the backup ID assigned to a Container backup using the `-l` and `-f` options of the `vzarestore` command. You can also restore only certain files from the backup archive of Container 101 using the `--files` option. For detailed information on all options that can be used with the `vzarestore` utility, see the *Parallels Virtuozzo Containers 4.7 Reference Guide*.

Restoring Containers Based on Standard Templates

If you have one or more backups of Containers that are based on standard templates, you can use the following procedure to restore them on Hardware Nodes running Parallels Virtuozzo Containers 4.7 (called the Destination Node):

1. Make sure that the standard OS template and all standard application templates used by the Container you plan to restore are installed on the Destination Node.
2. Restore the Container.

Installing Standard Templates

To install a standard OS or application template, you can use the `rpm -i` command. For example, you can use the following command to install the Samba standard template:

```
# rpm -ivh samba-template-20060211-1.0-1.i386.rpm
Preparing...        ################################### [100%]
   1:samba-template ################################### [100%]
```

You can also use the `vzmtemplate` utility to copy standard templates from one Hardware Node to another. For example, you can copy the `redhat-as4` OS template installed on the Source Node to the Destination Node with the IP address of `192.168.0.9` by running this command:

```
# vzmtemplate root@192.168.0.9 redhat-as4
root@192.168.0.197's password:
Connection to Destination Node (192.168.0.9) is successfully established
Copying template "redhat-as4"
...
```

Restoring the Container

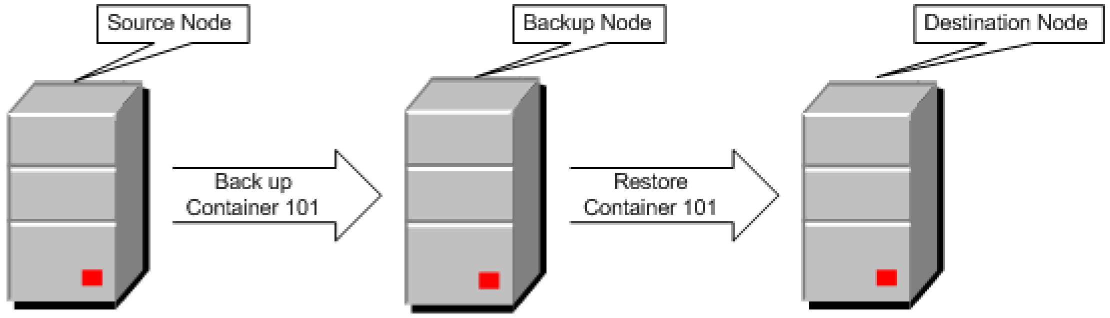
Now that all necessary standard templates are available on the Hardware Node with Parallels Virtuozzo Containers 4.7, you can restore the Container. For example, you can run the following command on the Destination Node:

```
# vzarestore 101 --storage backup_root:1qaz2wsx@192.168.200.200
Restore container: Container101 from 1361ac21-4cae-4981-...
Container ct101 was restored successfully.
```

This command restores the latest backup of Container 101 stored on the Backup Node with IP address `192.168.200.200` to the Destination Node. For more information on restoring Containers, see **Backing Up and Restoring Containers** (p. 61).

Managing Backups in Parallels Management Console

Parallels Management Console deals with three kinds of Nodes - the *Source Nodes* (the Nodes where Containers are hosted during their backing up); the *Backup Nodes* (the Nodes where Container backups are stored); and the *Destination Nodes* (the Nodes where Container backups are restored).



These Nodes are singled out by their functionality only. In reality, one and the same Hardware Node may perform two or even three functions. Usually, the Source and Destination Node are represented by one and the same Hardware Node, because you will likely want the Containers you back up to be restored to their original Node. However, setting up a dedicated Backup Node is recommended.

You should make sure that all the three Nodes are registered in Management Console before starting to work with them.

You can perform the following backup-related operations in Parallels Management Console:

- Assign the default Backup Node for the given Source Node.
- Set the default backup location on the Backup Node.
- Back up a single Container from the Source Node to the Backup Node.
- Back up a number of Containers or all Containers on the Node to the Backup Node.
- Restore a single Container from the Backup Node to the Destination Node.
- Restore a number of Containers or all Containers of a from the Backup Node to the Destination Node.
- Restore individual files from the Container backup on the Backup Node to the Destination Node.
- Manage the Backup Nodes.
- Search the backup of a given Container from the Source Node across all the Backup Nodes.
- Automate the task of backing up Containers by setting backups to be run on a schedule.

Setting Default Backup Parameters

Parallels Virtuozzo Containers allows you to specify a number of default backup parameters that can then be used when creating Container backup archives. These parameters include the following:

- default Backup Node
- default backup location on the Backup Node
- default backup compression level (p. 69)
- default backup type (p. 71)

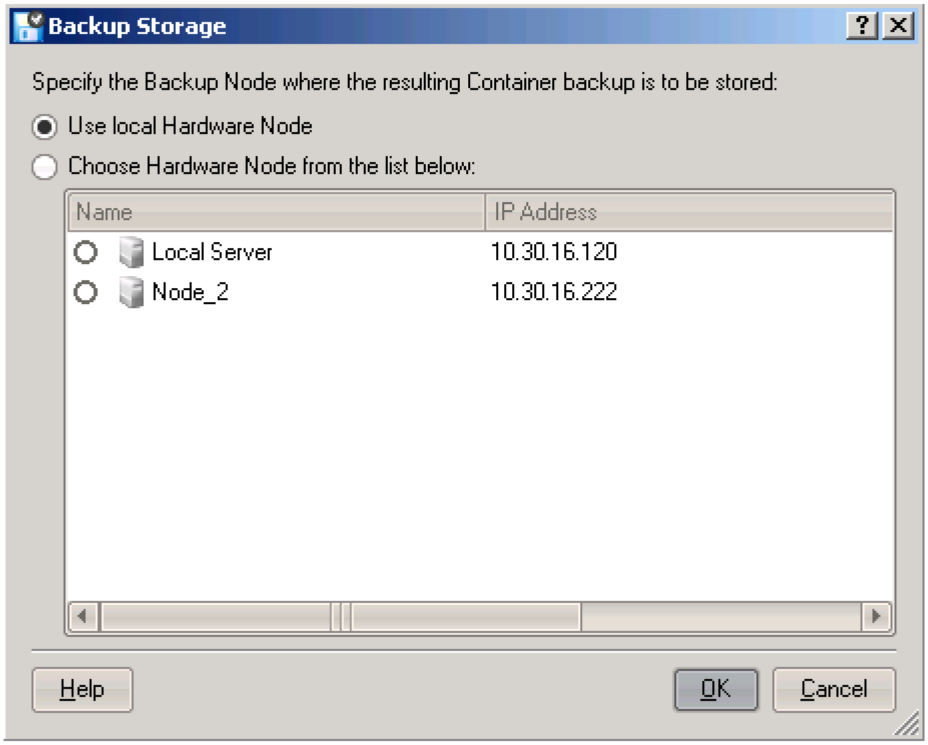
The following subsections describe in detail how to configure these parameters in Parallels Management Console.

Assigning the Default Backup Node

When backing up Containers from a Source Node, you need specify the Backup Node where the resulting backups will be stored. Parallels Management Console allows you to set the default Backup Node for a given Source Node by doing the following:

1 Right-click the respective Source Node, and choose **Backup** > **Set Default Backup Options**.

2 Click the **Change** button next to the **Server** field:



Specify the Backup Node where the resulting Container backup is to be stored:

● Use local Hardware Node

○ Choose Hardware Node from the list below:

Name	IP Address
○ Local Server	10.30.16.120
○ Node_2	10.30.16.222

3 In the **Backup Storage** window, do the following:

- If you do not want to use a dedicated Node for storing Container backups, select the **Use local Hardware Node** radio button, and click **OK** to set the Source Node as the default Backup Node.
- If you are going to use a dedicated Node for storing Container backups, select the **Choose Hardware Node from the list below** radio button. The table below this radio button lists all Nodes registered in Parallels Management Console together with their IP addresses. If the default Backup Node already exists for the given Source Node, it is selected in the table. Select the Node you want to be the default Backup Node for the Source Node, and click **OK.**

4 Click **OK**.

The assignment of the default Backup Node brings about the following effects:

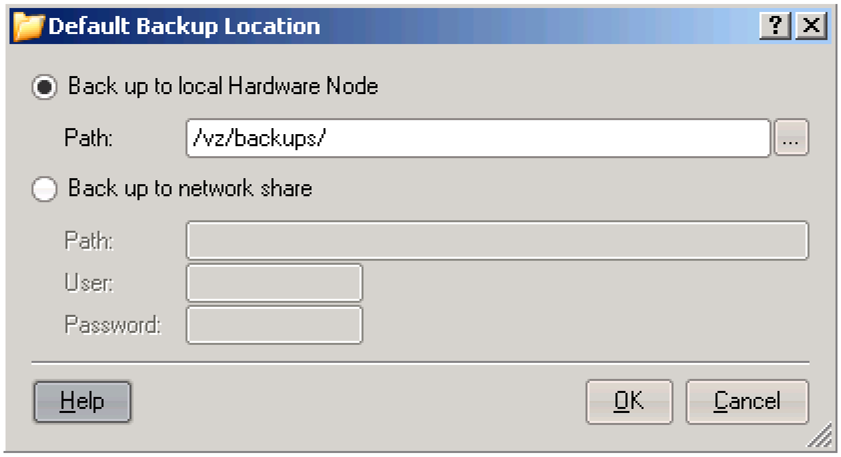
- When backing up Containers from the corresponding Source Node in Parallels Management Console and Parallels Virtual Automation using the 'default' backup mode, the backups are automatically placed onto the default Backup Node.
- When backing up Containers form the corresponding Source Node in Parallels Management Console and Parallels Virtual Automation using the 'custom' backup mode, you are automatically offered to place the backups onto the default Backup Node.
- When a Container administrator backs up his or her Container using Parallels Power Panel, the backup is automatically placed on the default Backup Node.

There are no restrictions as to which Hardware Node can be the default Backup Node. It just must be registered in Parallels Management Console (otherwise, it will not be displayed in the table on the **Backup Storage** screen) and have sufficient disk space for housing multiple backups.

Note: You can use any Hardware Node as a Backup Node irrespective of a Parallels Virtuozzo Containers version installed on this Node. So, you can back up a Container from the Node running the Parallels Virtuozzo Containers 32-bit version and store it on the Node running the Parallels Virtuozzo Containers 64-bit version, and vice versa.

Setting the Default Backup Location

Parallels Management Console allows you to change the location on the Backup Node where Container backups are stored. By default, the `/vz/backup` directory is used. To set another backup directory as the default one, right-click the Hardware Node in the left pane of the Parallels Management Console main window, and choose **Backup** > **Set Default Backup Location**. The **Default Backup Location** window appears.



● Back up to local Hardware Node

Path: /vz/backups/

○ Back up to network share

In this window, do the following:

- Select the **Back up to local Hardware Node** radio button, and specify a backup directory on one of the Backup Node local disk drives. To set a new backup directory, type its full path on the Node in the **Path** field or click the **...** button and select the desired directory in the displayed window.
- Select the **Back up to network share** radio button to specify a backup directory on a network share—that is, on a Backup Node network drive. To do this, enter the full path to the directory in the **Path** field (for example, `\\share\backup_directory`). If the network drive where the backup directory is located is password-protected, you need to additionally specify the user name and password to access this share in the **User** and **Password** fields, respectively.

Once you specify the path to a new directory for storing Container backups, click **OK** for the changes to take effect.

Note: While defining the default backup directory, make sure that the disk drive where this directory will be located has sufficient disk space for storing multiple Container backups.

Defining the Default Backup Compression Level

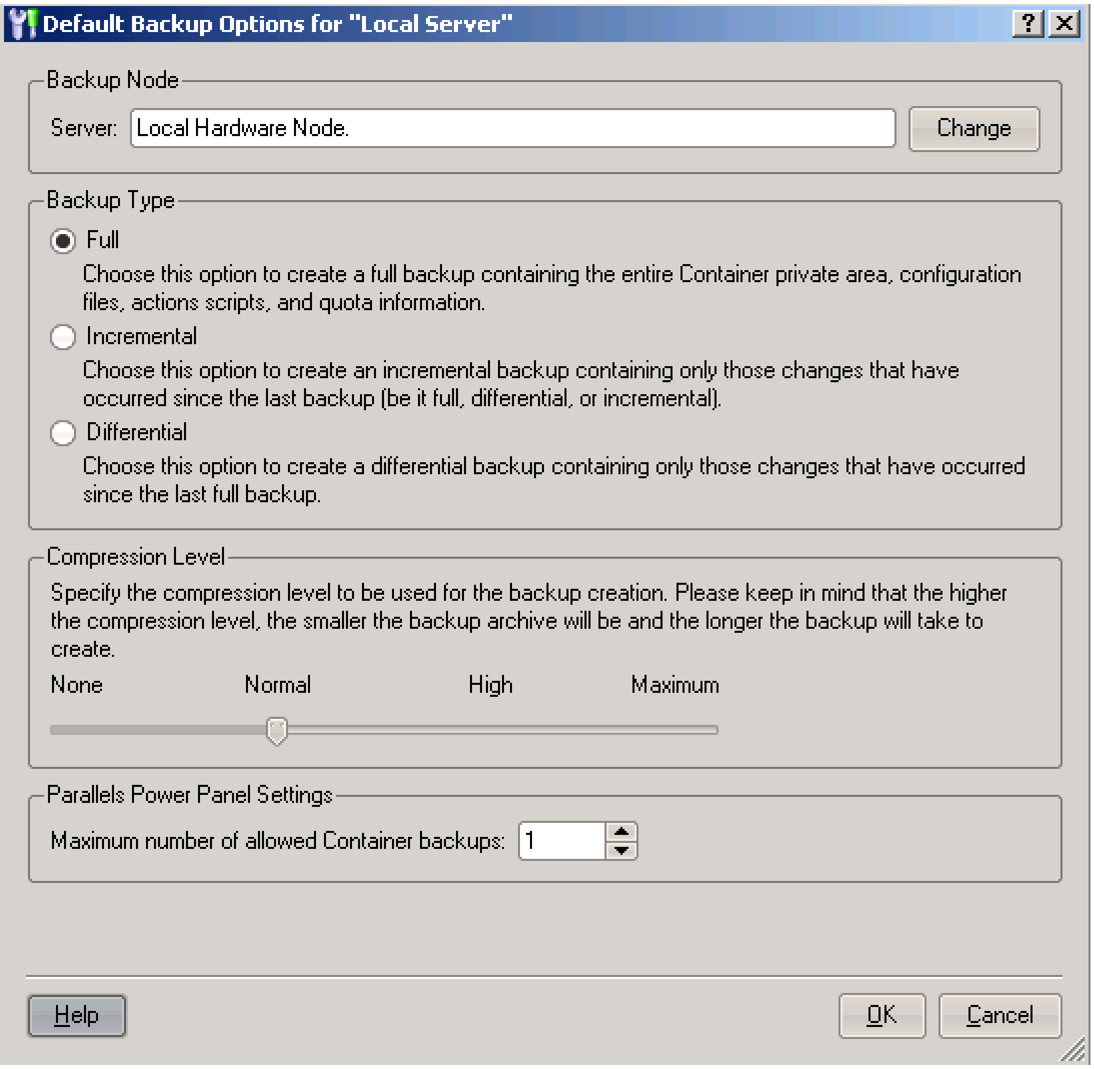
Parallels Virtuozzo Containers allows you to configure the default backup compression level by setting it to one of the following:

- **None**. In this case, the Container backup is created without any compression. Using this level of compression, you can greatly reduce the backup creation time. However, the size of the resulting backup file may significantly increase as compared to other compression levels.
- **Normal**. In this case, the Container backup is created with a normal level of compression. This compression level is set by default and is suitable for backing up most Container files and directories.
- **High**. In this case, the Container backup is created with the high level of compression. The size of the resulting backup file is smaller than that of the backup file compressed in the 'normal' and 'none' modes; however, it takes longer to create the backup file.
- **Maximum**. In this case, the Container backup is created with the maximum level of compression. The size of the resulting backup file is the smallest and the time of the backup creation is the longest.

In general, the optimal data compression level depends on the type of files to be stored in the backup archive. For example, it is advisable to use the 'normal' and 'none' compression types if most of the files to be backed up are already compressed (e.g., the files with the `.zip` and `.rar` extensions) or can be compressed with a low degree of efficiency (for example, all executable files with the `.exe` extension or image files with the `.jpg`, `.jpeg`., and `.gif` extensions).

To configure the default backup compression level, do the following:

1 Right-click the respective Source Node, and choose **Backup** > **Set Default Backup Options**.



Default Backup Options for "Local Server"

Backup Node

Server: Local Hardware Node.

Backup Type

● Full

Choose this option to create a full backup containing the entire Container private area, configuration files, actions scripts, and quota information.

○ Incremental

Choose this option to create an incremental backup containing only those changes that have occurred since the last backup (be it full, differential, or incremental).

○ Differential

Choose this option to create a differential backup containing only those changes that have occurred since the last full backup.

Compression Level

Specify the compression level to be used for the backup creation. Please keep in mind that the higher the compression level, the smaller the backup archive will be and the longer the backup will take to create.

Parallels Power Panel Settings

Maximum number of allowed Container backups: 1

2 Under the **Compression Level** group in the displayed window, move the slider to the left or to the right to specify the desired compression level.

3 Click **OK**.

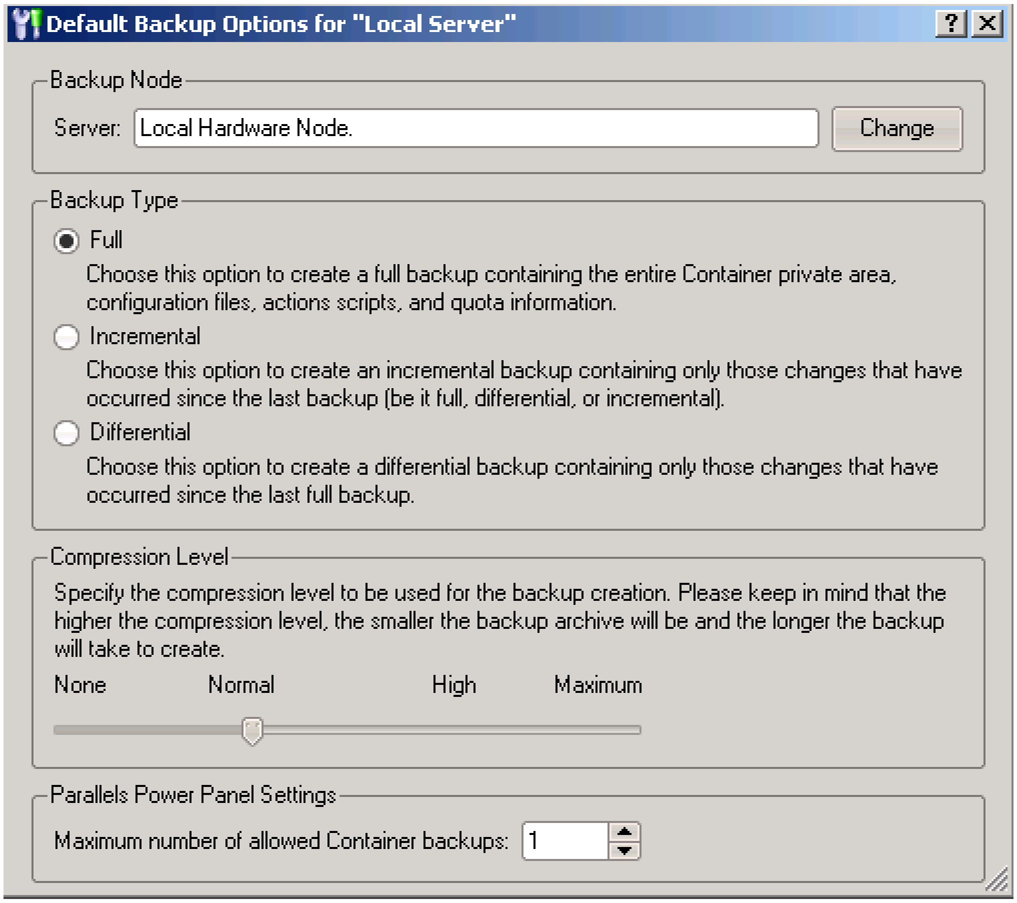
Specifying the Default Backup Type

Another parameter that you may wish to configure and that will be applied to all Container backups created using the default backup mode is the backup type. Each backup file can be of one of the following types:

- A full backup containing the whole Container private area and its configuration file.
- An incremental backup containing only the files changed since the full backup or the previous incremental backup. An incremental backup may prove very useful because it records only the changes since the last Container backup (either full or incremental) and therefore is much less in size and takes much less time than the full backup. However, after several consecutive incremental backups it is recommended to create a full backup anew and start the incremental backups chain from scratch.
- A differential backup containing only the files changed since the last full backup. As a rule, this kind of backup requires less space than a full backup, but more space than an incremental backup.

You can configure the default backup type by doing the following:

1 Right-click the respective Source Node, and choose **Backup** > **Default Backup Node Configuration**:



Default Backup Options for "Local Server"

Backup Node

Server: Local Hardware Node.

Backup Type

● Full

Choose this option to create a full backup containing the entire Container private area, configuration files, actions scripts, and quota information.

○ Incremental

Choose this option to create an incremental backup containing only those changes that have occurred since the last backup (be it full, differential, or incremental).

○ Differential

Choose this option to create a differential backup containing only those changes that have occurred since the last full backup.

Compression Level

Specify the compression level to be used for the backup creation. Please keep in mind that the higher the compression level, the smaller the backup archive will be and the longer the backup will take to create.

Compression Level: Normal

Parallels Power Panel Settings

Maximum number of allowed Container backups: 1

2 Under the **Backup Type** group in the displayed window, choose one of the following options:

- Select the **Full** radio button to always create full backup archives containing the whole Container private area, all Container-related configuration files, action scripts, etc.
- Select the **Incremental** or **Differential** radio button to always perform incremental or differential backups, respectively. If an incremental or differential backup is performed, and the corresponding full backup cannot be found, a full backup is automatically performed.

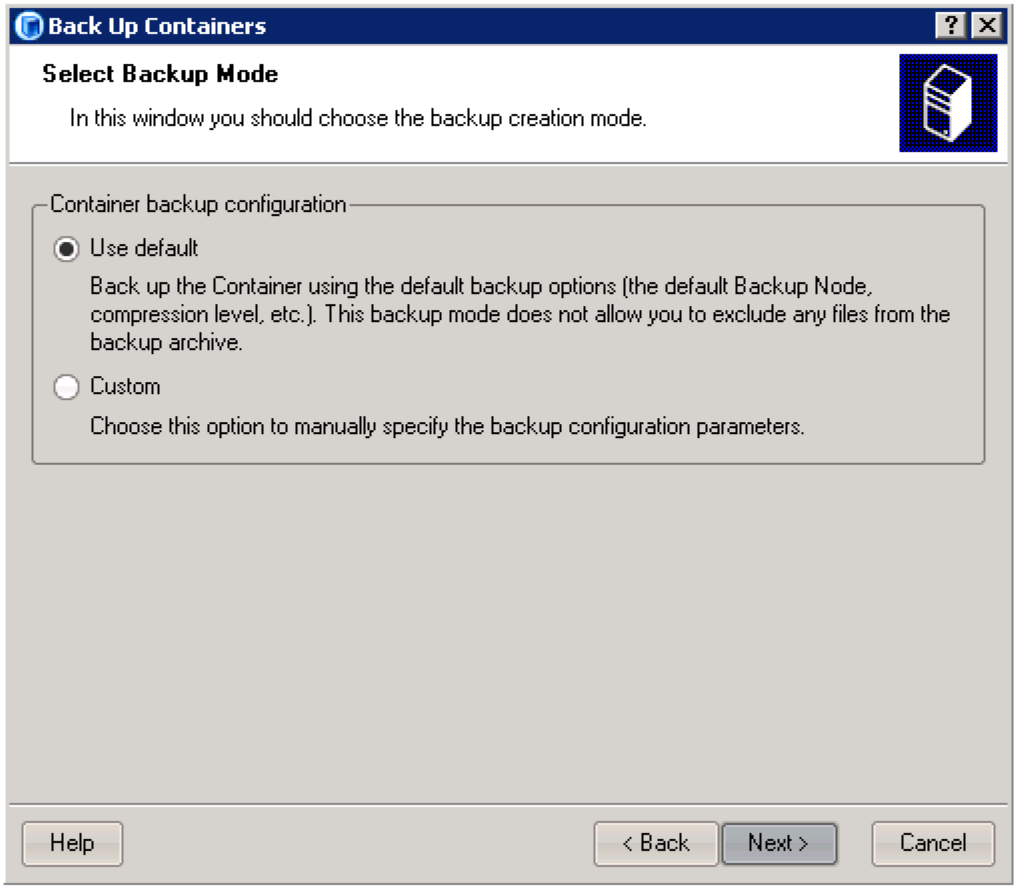
3 Click **OK**.

Backing Up a Single Container

To back up a single Container on the Source Node, do the following:

1 In Parallels Management Console, click the **Parallels Virtuozzo Containers** item under the corresponding Source Node to open the Container manager window.

2 Right-click the Container you want to back up, and choose **Backup** > **Back Up Container**. The **Back Up Containers** wizard opens.



Select Backup Mode

In this window you should choose the backup creation mode.

Container backup configuration

● Use default

Back up the Container using the default backup options (the default Backup Node, compression level, etc.). This backup mode does not allow you to exclude any files from the backup archive.

○ Custom

Choose this option to manually specify the backup configuration parameters.

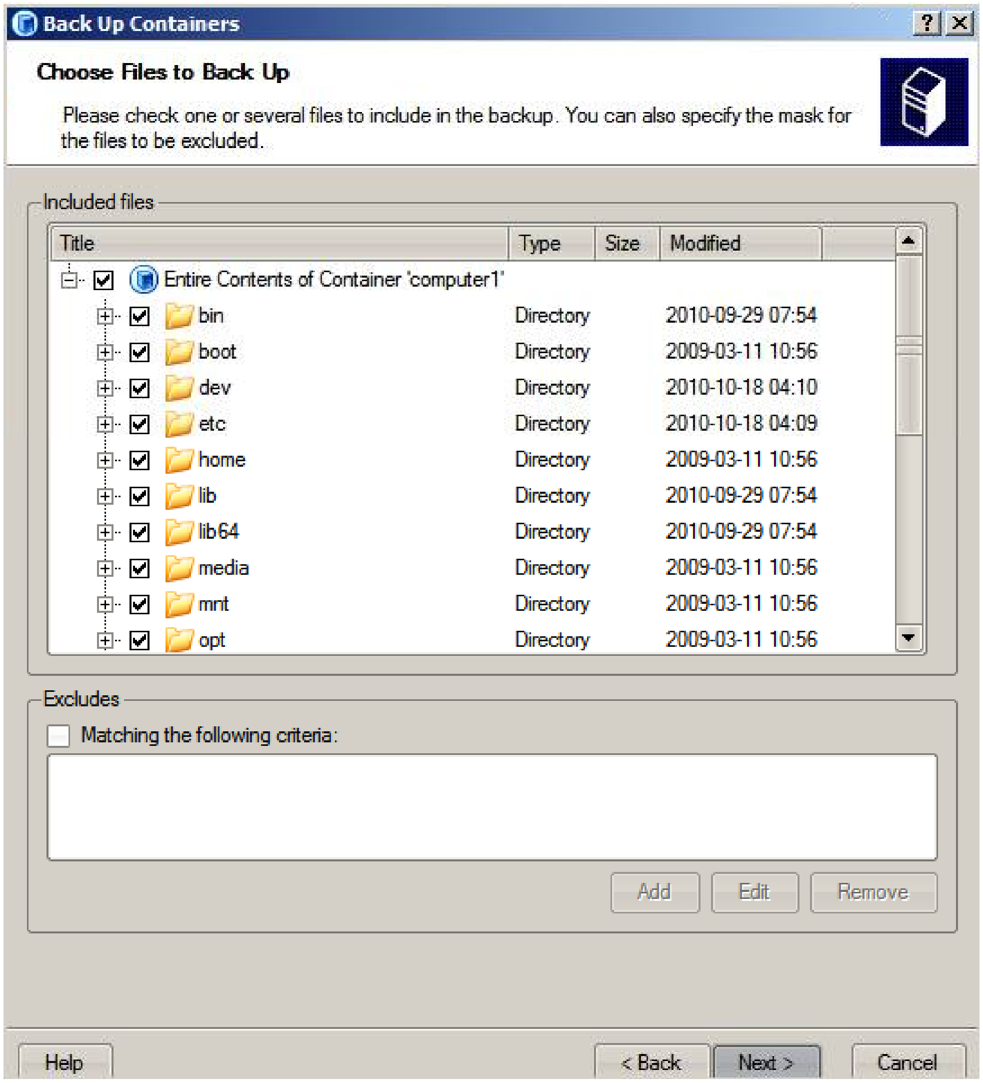
3 In the first step of the wizard, choose the Container backup mode:

- **Default**: select this radio button to back up the Container using the default backup mode. When run in this mode, the default backup parameters are used for creating the Container backup. You can only set the backup description and configure the default backup policy.

Note: Detailed information on what default backup parameters are and how to manage them is given in **Setting the Default Backup Parameters** (p. 66).

- **Custom**: select this radio button to manually set the parameters to be applied to the resulting backup archive. In this case you will have to go through a number of steps (**Steps 4** and **5**) of the **Back Up Containers** wizard and set all the parameters of the Container backup one by one.

4 In the second step of the wizard, specify the files and directories to be included in the backup:



Back Up Containers

Choose Files to Back Up

Please check one or several files to include in the backup. You can also specify the mask for the files to be excluded.

Included files

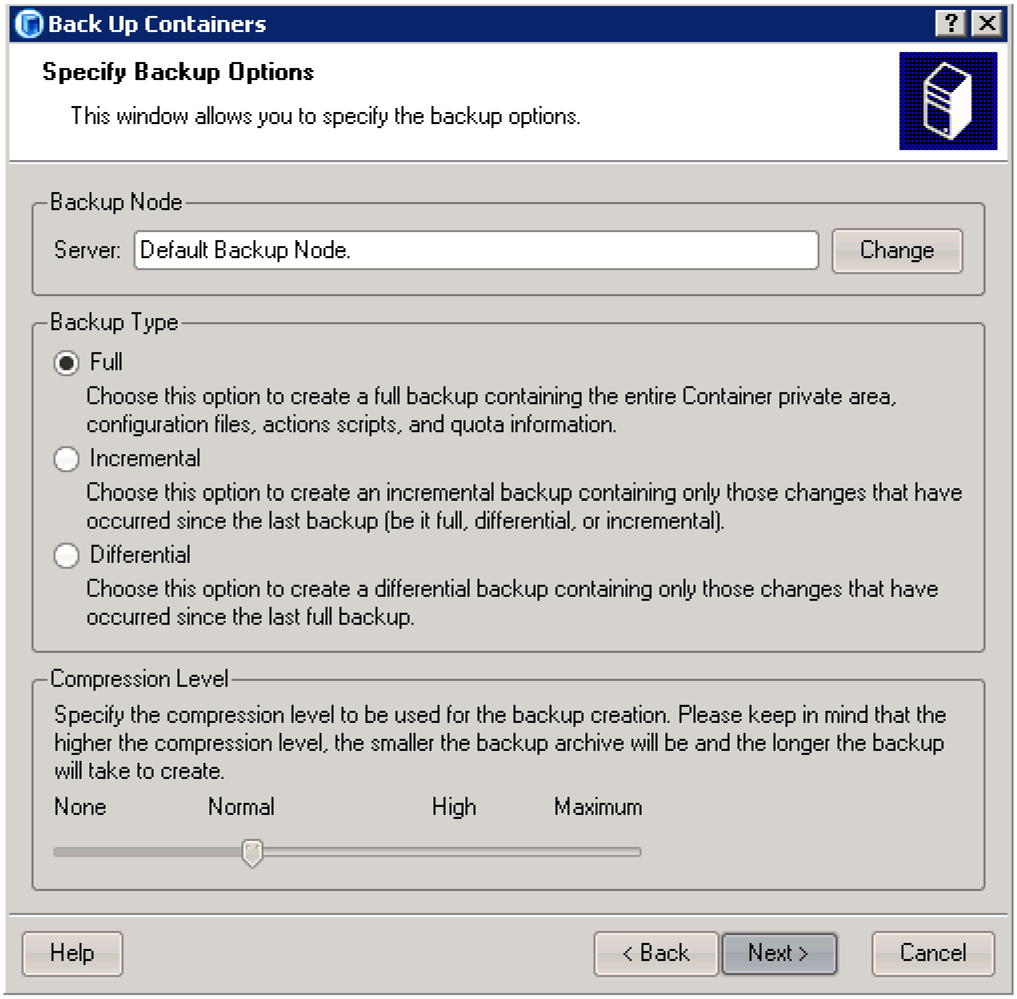
Title	Type	Size	Modified
☑ Entire Contents of Container 'computer1'			
☑ bin	Directory		2010-09-29 07:54
☑ boot	Directory		2009-03-11 10:56
☑ dev	Directory		2010-10-18 04:10
☑ etc	Directory		2010-10-18 04:09
☑ home	Directory		2009-03-11 10:56
☑ lib	Directory		2010-09-29 07:54
☑ lib64	Directory		2010-09-29 07:54
☑ media	Directory		2009-03-11 10:56
☑ mnt	Directory		2009-03-11 10:56
☑ opt	Directory		2009-03-11 10:56

Excludes

□ Matching the following criteria:

By default, all Container files and directories are included in the backup archive. To leave out a file or directory from the backup process, clear its check box in the **Included files** table. You can also select the **Matching the following criteria** check box and use the **Add**/**Edit**/**Remove** buttons to set the parameters to be met by the file/directory to exclude it from the backup process. You can specify the full path to the corresponding file/directory, enter its name, or define any filter compatible with standard Linux masking rules (i.e. with standard globs). For example, you can indicate `/usr/MyDirectory/MyFile.txt` to exclude the `MyFile.txt` file from the backup process or type `*.bmp` to leave out all files with the `bmp` extension.

5 Next, specify the main backup parameters.



Back Up Containers

Specify Backup Options

This window allows you to specify the backup options.

Backup Node

Server: Default Backup Node.

Backup Type

● Full
Choose this option to create a full backup containing the entire Container private area, configuration files, actions scripts, and quota information.

○ Incremental
Choose this option to create an incremental backup containing only those changes that have occurred since the last backup (be it full, differential, or incremental).

○ Differential
Choose this option to create a differential backup containing only those changes that have occurred since the last full backup.

Compression Level

Specify the compression level to be used for the backup creation. Please keep in mind that the higher the compression level, the smaller the backup archive will be and the longer the backup will take to create.

In this window, you can configure the following backup parameters:

- Choose the Backup Node where the backup is to be stored. You can leave the Backup Node offered by Parallels Management Console by default or use the **Change** button to specify another Backup Node. For detailed information on Backup Nodes, see **Assigning the Default Backup Node** (p. 66).
- Decide on the backup compression level: 'None', 'Normal', 'High', or 'Maximum'. Detailed information on compression levels is given in **Defining the Default Compression Level** (p. 69).
- Specify the backup type. It can be full, incremental, or differential. Detailed information on backup types is provided in **Specifying the Default Backup Type** (p. 71). If you are backing up a single Container, and no backup of this Container has been found on the Backup Node, the **Backup Type** group is not shown, and a full backup is automatically created.

6 In the next step of the wizard, you can set the following backup parameters:

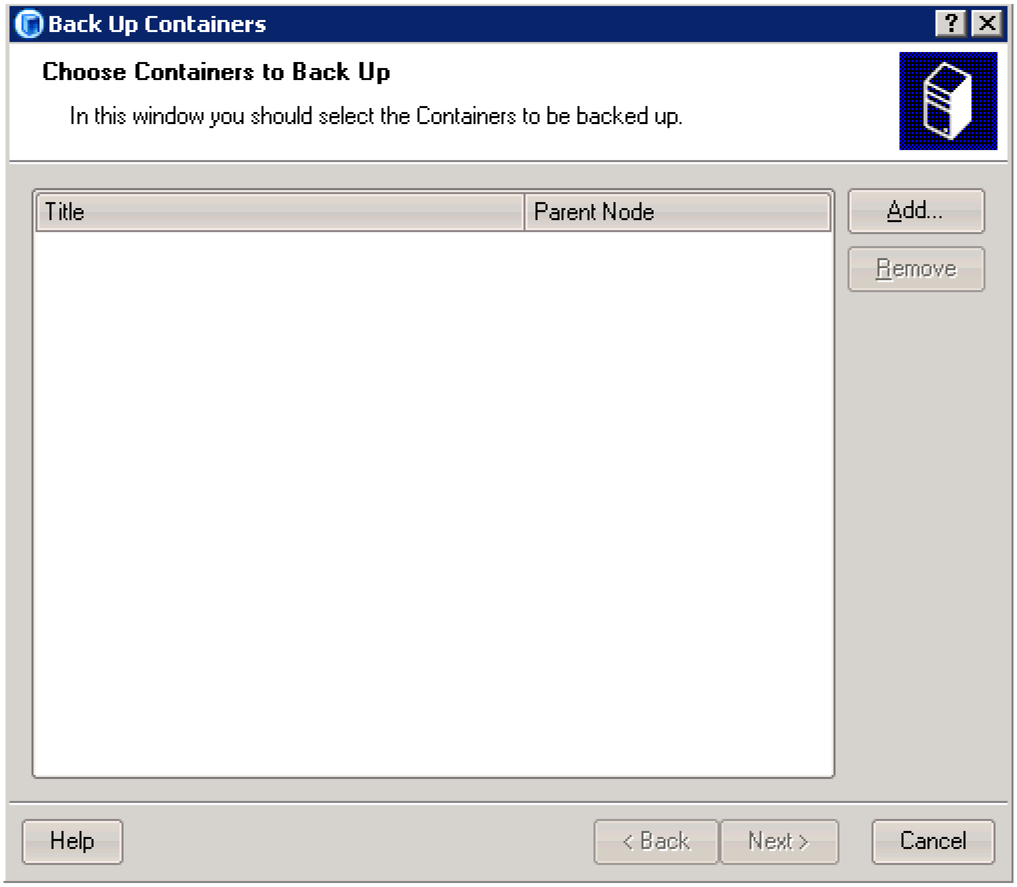
- Provide the backup description in the **Backup description** field, if necessary. The description can be any text containing any backup-related information (for example, the backup purpose).
- Do not stop the Container backup even if any errors appear (select the **Do not stop on errors** check box) or break the backup process if any malfunction occurs (clear the check box).
- Do not stop the backup process if one or more of the Containers to be backed up do not exist on the Source Node (select the **Ignore non-existent Containers** check box) or break the backup process in this case (clear the check box). You can use this option when backing up several Containers at once.

7 The last screen allows you to review the information provided by you in the previous steps of the wizard. Click **Finish** to start creating the Container backup; otherwise, click **Back** to return to any step and correct the corresponding parameter.

Backing Up Groups of Containers

To back up several or all Containers from a Source Node, right-click the **Parallels Virtuozzo Containers** item under the corresponding Source Node, and choose **Backup** > **Back up Containers**. The **Back Up Containers** wizard opens. In this wizard, do the following:

1 Choose the Containers from the Source Node to back up.



Back Up Containers

Choose Containers to Back Up

In this window you should select the Containers to be backed up.

Title	Parent Node

To schedule one or more Containers for backing up, click the **Add** button in the top left corner, and in the displayed window, select the names of the Containers to back up, and click **OK**. The selected Containers will be shown in the table in the center of the **Choose Containers to Back Up** screen. Click **Next** to proceed with the wizard.

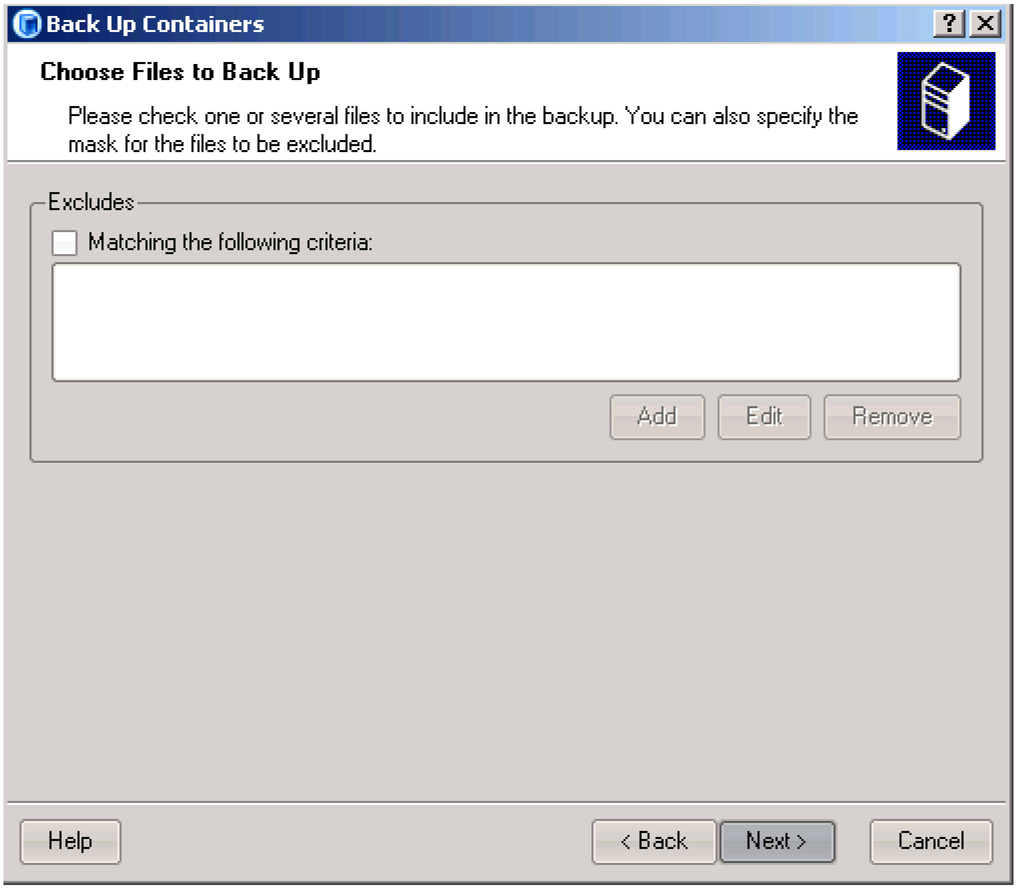
2 Choose the Container backup mode:

- **Default**: select this radio button to back up the Container using the default backup mode. When run in this mode, the default backup parameters are used for creating the Container backup. You can only set the backup description and configure the default backup policy.

Note: Detailed information on what default backup parameters are and how to manage them is given in **Setting Default Backup Parameters** (p. 66).

- **Custom**: select this radio button to manually set the parameters for the resulting backup archive. In this case, you will have to go through a number of steps (**Steps 3** and **4**) of the **Back Up Containers** wizard and set all the required backup parameters one by one.

3 Specify the files and directories to include in the backup.



Back Up Containers

Choose Files to Back Up

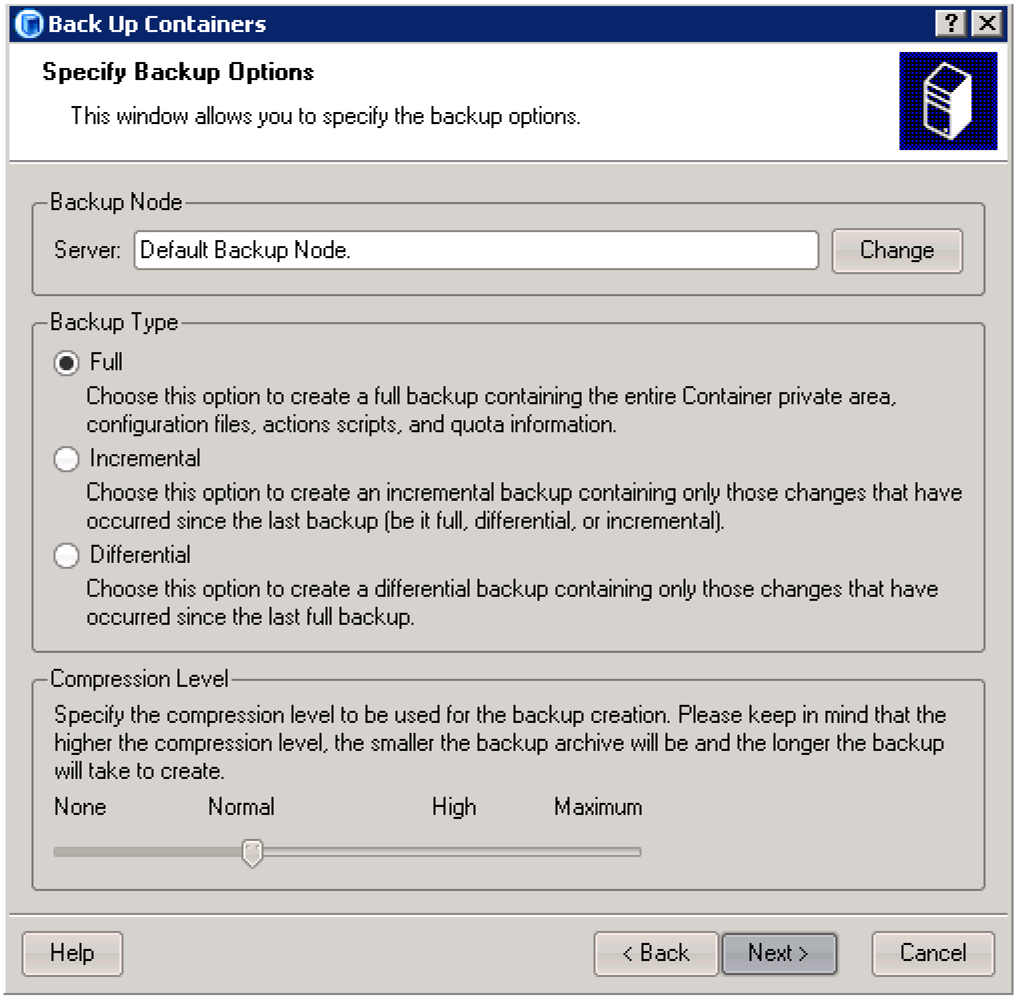
Please check one or several files to include in the backup. You can also specify the mask for the files to be excluded.

Excludes

□ Matching the following criteria:

By default, all Container files and directories are included in the backup archive. However, you can select the **Matching the following criteria** check box and use the **Add**/**Edit**/**Remove** buttons to set the parameters to be met by the file/directory to exclude it from the backup process. You can specify the full path to the corresponding file/directory, enter its name, or define any filter compatible with standard Linux masking rules (i.e. with standard globs). For example, you can indicate `/usr/MyDirectory/MyFile.txt` to exclude the `MyFile.txt` file from the backup process or type `*.bmp` to leave out all files with the `bmp` extension.

4 Next, specify the main backup parameters.



Specify Backup Options

This window allows you to specify the backup options.

Backup Node

Server: Default Backup Node.

Backup Type

● Full

Choose this option to create a full backup containing the entire Container private area, configuration files, actions scripts, and quota information.

○ Incremental

Choose this option to create an incremental backup containing only those changes that have occurred since the last backup (be it full, differential, or incremental).

○ Differential

Choose this option to create a differential backup containing only those changes that have occurred since the last full backup.

Compression Level

Specify the compression level to be used for the backup creation. Please keep in mind that the higher the compression level, the smaller the backup archive will be and the longer the backup will take to create.

In this window, you can configure the following backup parameters:

- *Backup Node*. This Node is the place where the Container backup will be stored. You can leave the Backup Node offered by Parallels Management Console by default or use the **Change** button to specify the desired Backup Node. For detailed information on Backup Nodes, see **Assigning the Default Backup Node** (p. 66).
- *Backup compression level*: 'None', 'Normal', 'High', or 'Maximum'. Detailed information on compression levels is provided in **Defining the Default Compression Level** (p. 69).
- *Backup type*. It can be full, incremental, or differential. Detailed information on backup types is provided in the **Specifying the Default Backup Type** subsection (p. 71).

5 In the next step of the wizard, you can set the following backup parameters:

- Provide the backup description in the **Backup description** field, if necessary. The description can be any text containing any backup-related information (for example, the backup purpose).
- Do not stop the Container backup even if any errors appear (select the **Do not stop on errors** check box) or break the backup process if any malfunction occurs (clear the check box).
- Do not stop the backup process if one or more of the Containers to be backed up do not exist on the Source Node (select the **Ignore non-existent Containers** check box) or break the backup process in this case (clear the check box).

6 Review the information provided by you in the previous steps of the wizard. Click **Finish** to start creating the Container backup or click **Back** to return to any step and correct the corresponding parameters.

Another way of backing up a number of Containers from the given Source Node is the following:

1 Expand the *Source Node* item in the left pane of the Parallels Management Console main window, and click the **Parallels Virtuozzo Containers** item to open the Containers manager window.

2 Select the Containers you want to back up. Use the CTRL and SHIFT keys for selecting a number of Containers.

3 Right-click the selection, choose **Back up Containers**.

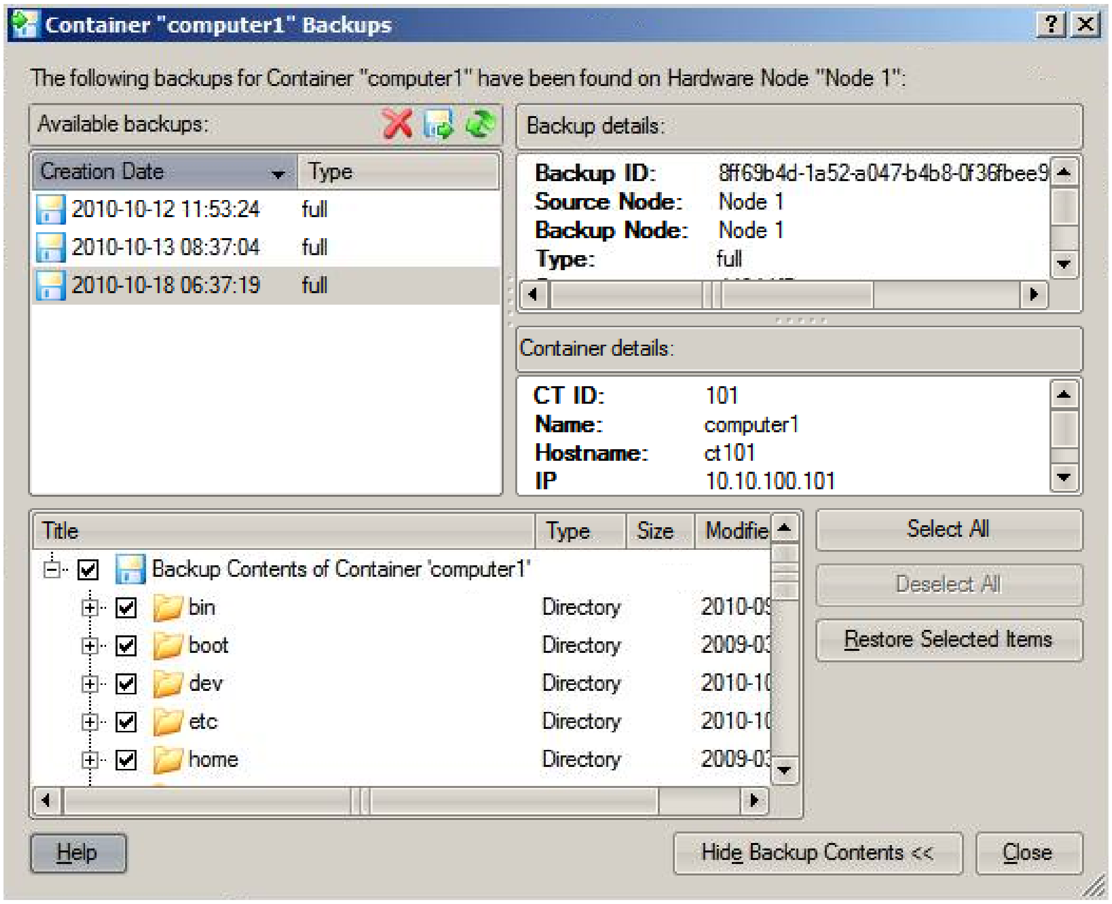
The aforementioned **Back Up Containers** wizard is opened directly at the second page because the first page (**Choose Containers to Back Up**) becomes unnecessary.

Browsing the Backup Contents

Parallels Management Console allows you to browse the directory structure of any Container backup as if this backup had already been restored and restore only the needed files and directories, if necessary. To view the backed up files and directories of a Container backup, do the following:

1 Choose the **Backups** item in the Parallels Management Console right pane, right-click the Container backup whose contents you want to browse, and choose **Properties**.

2 In the displayed window, select the corresponding backup in the **Available backups** table, and click the **Show Backup Contents** button at the bottom of the window.



Container "computer1" Backups

The following backups for Container "computer1" have been found on Hardware Node "Node 1":

Available backups:

Creation Date	Type
2010-10-12 11:53:24	full
2010-10-13 08:37:04	full
2010-10-18 06:37:19	full

Backup details:

Backup ID: 8ff69b4d-1a52-a047-b4b8-0f36fbee9
Source Node: Node 1
Backup Node: Node 1
Type: full

Container details:

CT ID: 101
Name: computer1
Hostname: ct101
IP 10.10.100.101

Title	Type	Size	Modifie
☑ Backup Contents of Container 'computer1'			
☑ bin	Directory		2010-0
☑ boot	Directory		2009-0
☑ dev	Directory		2010-1
☑ etc	Directory		2010-1
☑ home	Directory		2009-0

3 Double-click the directory to see its contents. The information on any file/directory inside the backup is shown in the table having the following columns:

Column Name	**Description**
Title	The name of the file/directory.
Type	Denotes whether the object is a file, directory, or Parallels Virtuozzo Containers file link (i.e. a link to the corresponding file on the Node).
Size	The size of the file.
Modified	The date and time when the file/directory was modified last time.

If you wish to restore any files and/or directories from the backup to the actual Container, select the check boxes near the corresponding files/directories and click the **Restore Selected Items** button. Detailed information on how to restore individual files/directories is provided in the **Restoring Container Files** subsection.

Restoring a Single Container

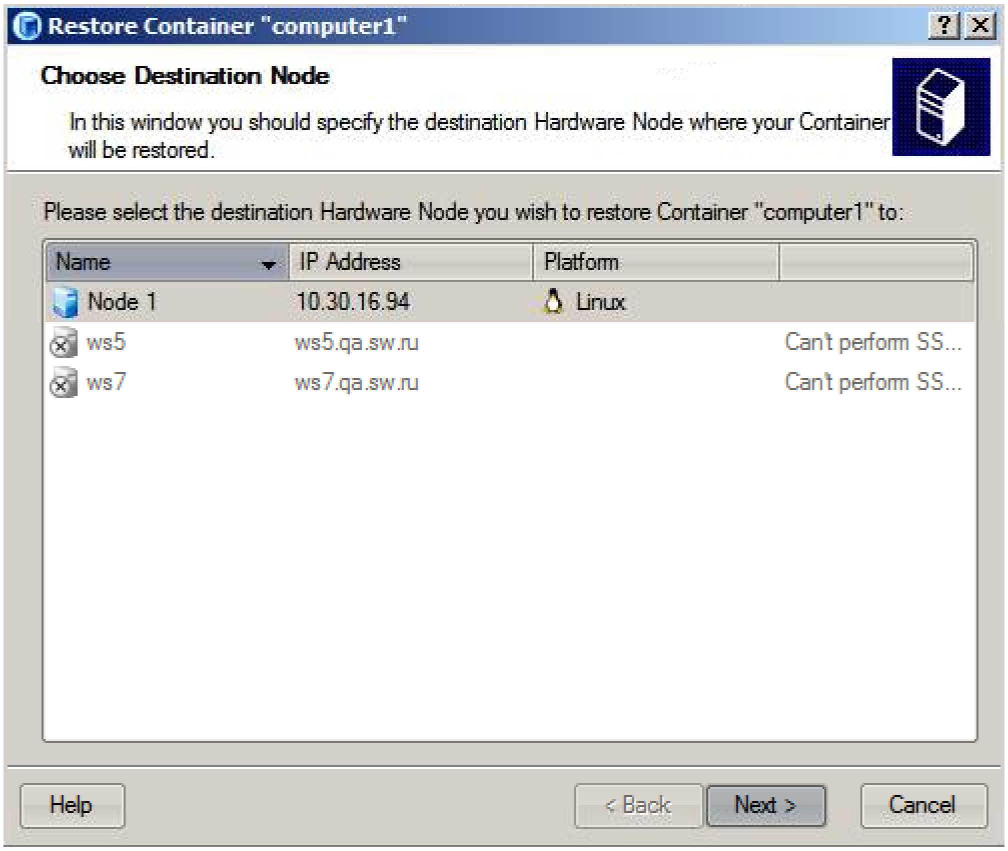
To restore a Container from its backup, do the following:

1 Expand the *Source Node* item in the left pane of the Parallels Management Console main window, and click the **Parallels Containers** item to open the Containers manager window.

2 Select the Container whose backup you want to restore from the Backup Node.

3 Click the right mouse button, and choose **Backup** > **Restore Container**.

The **Restore Container** wizard opens.



Restore Container "computer1"

Choose Destination Node

In this window you should specify the destination Hardware Node where your Container will be restored.

Please select the destination Hardware Node you wish to restore Container "computer1" to:

Name	IP Address	Platform	
Node 1	10.30.16.94	Linux	
ws5	ws5.qa.sw.ru		Can't perform SS...
ws7	ws7.qa.sw.ru		Can't perform SS...

In this wizard, do the following:

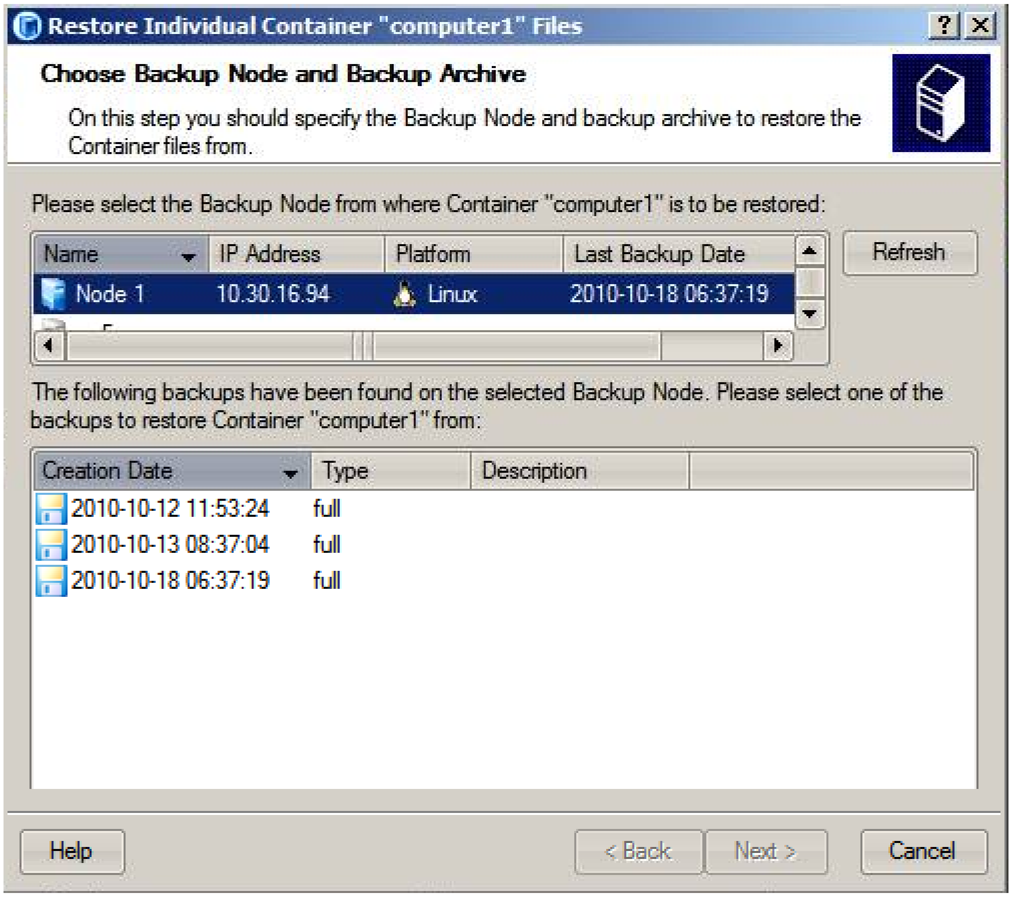
- In the **Choose Backup Node and Backup Archive** window:
  - Select the Backup Node. This Node is the place where the Container backup is stored. The **Last Backup Date** column in the list of Backup Nodes shows the date and time of the last backup (if any) of the selected Container on the corresponding Node.
  - Select the backup from which to restore the Container. A Container can have a number of backups made at different dates and of different types. As a rule, you choose the most recent backup, unless you have reasons to restore an intermediary one.
- In the **Review Container Restoration Settings** window:
  - Review the parameters provided by you in the previous step of the wizard.
  - Click the **Finish** button to start restoring the Container.

Note: During this operation, the Destination Node is supposed to be the same as the Source Node. For instructions on how to restore a Container to a Destination Node other than the Source Node, see **Managing the Backup Node**.

Restoring Container Files

Parallels Virtuozzo Containers allows you to browse the directory structure of any Container backup as if this backup had already been restored and restore only the needed files and folders/directories. To do this:

1. Expand the *Source Node* item in the left pane of the Parallels Management Console main window, and click the **Parallels Virtuozzo Containers** item.
2. Right-click the Container the files/folders of which you want to restore, and choose **Backup** > **Restore Individual Container Files**. The **Restore Individual Container Files** wizard opens.



Restore Individual Container "computer1" Files

Choose Backup Node and Backup Archive

On this step you should specify the Backup Node and backup archive to restore the Container files from.

Please select the Backup Node from where Container "computer1" is to be restored:

Name	IP Address	Platform	Last Backup Date
Node 1	10.30.16.94	Linux	2010-10-18 06:37:19

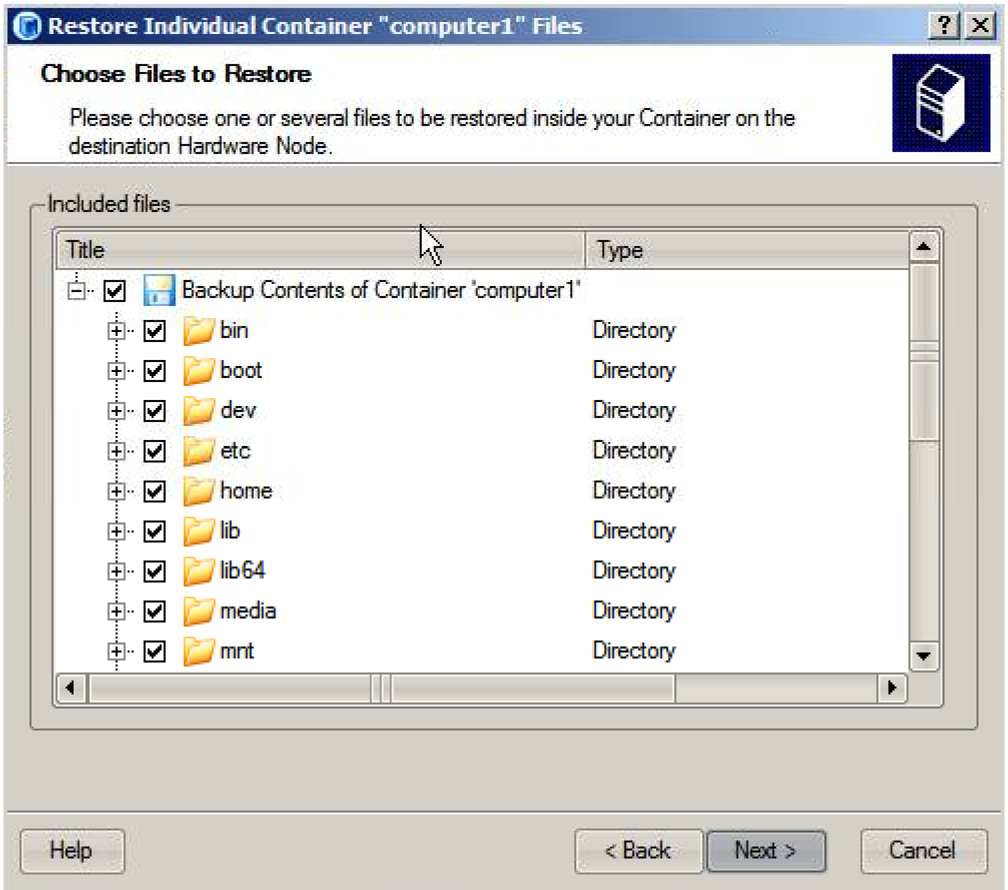
The following backups have been found on the selected Backup Node. Please select one of the backups to restore Container "computer1" from:

Creation Date	Type	Description
2010-10-12 11:53:24	full	
2010-10-13 08:37:04	full	
2010-10-18 06:37:19	full	

In the first step of the wizard, do the following:

- Choose the Backup Node. This Node is the place where the Container backup is stored. The **Last Backup Date** column in the list of Backup Nodes shows the date and time of the last backup (if any) of the selected Container on the corresponding Node.
- Choose the backup from which to restore the Container files/directories. A Container can have a number of backups made at different dates and of different types.

The second step of the wizard allows you to review and explore the contents of all the directories that were present inside the Container at the moment of the backup creation.



Restore Individual Container "computer1" Files

Choose Files to Restore

Please choose one or several files to be restored inside your Container on the destination Hardware Node.

Included files

Title	Type
☑ Backup Contents of Container 'computer1'	
☑ bin	Directory
☑ boot	Directory
☑ dev	Directory
☑ etc	Directory
☑ home	Directory
☑ lib	Directory
☑ lib64	Directory
☑ media	Directory
☑ mnt	Directory

Double-click the directory to see its contents. The information on any file/directory inside the backup archive is presented in the table having the following columns:

Column Name	Description
Title	The name of the file/directory.
Type	Denotes whether the object is a file, directory, or Parallels Virtuozzo Containers file link (i.e. a link to the corresponding file on the Node).
Size	The size of the file.
Modified	The date and time of the last modification of the file/directory.

To enqueue a file/directory for being restored, select its check box. You can restore all the files and subdirectories included in a given directory by selecting the check box next to this directory.

The last step of the wizard allows you to review the parameters provided by you in the previous steps of the wizard. If you are satisfied with the specified parameters, click **Finish** to start restoring the Container files/directories; otherwise, click **Back** and change the corresponding parameters.

Note: During this operation, the Destination Node is supposed to be the same as the Source Node. For instructions on how to restore Container files/folders/directories to a Destination Node other than the Source Node, see **Managing the Backup Node**.

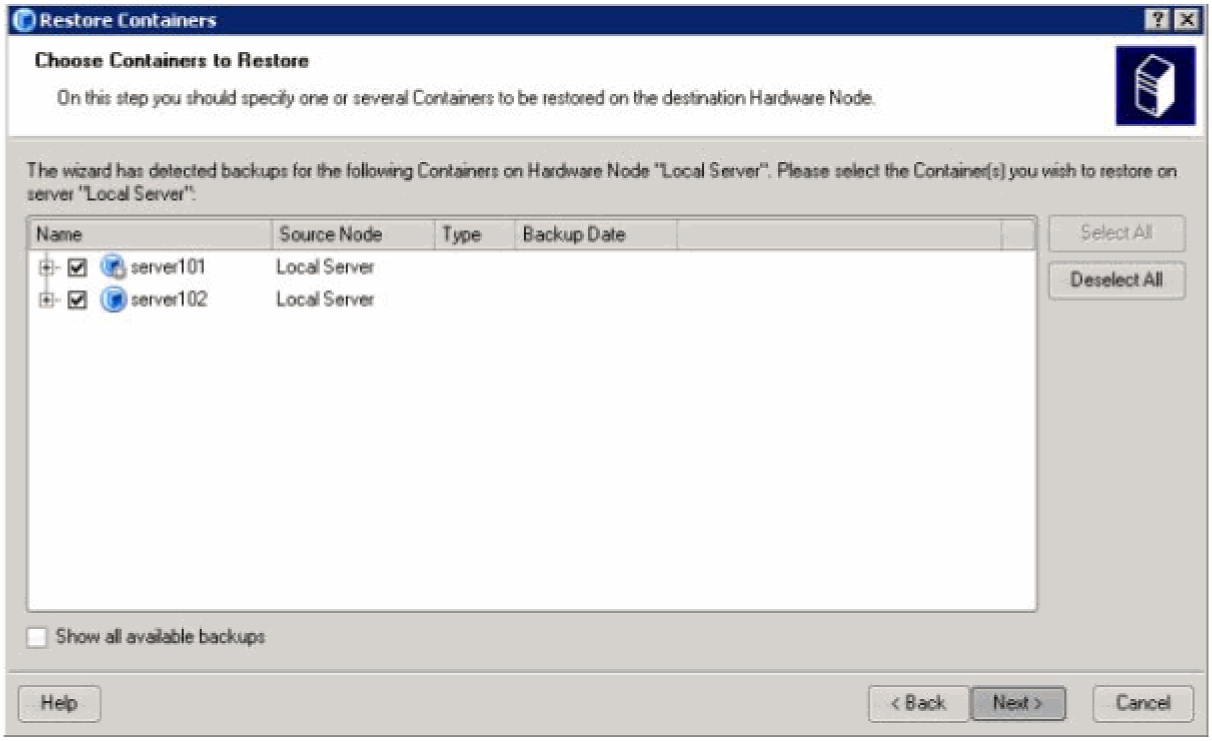
Restoring Groups of Containers

To restore several Containers of a single Source Node from their backups on the Backup Node, do the following:

1 Right-click the **Parallels Virtuozzo Containers** item under the corresponding *Source Node*, and choose **Backup** > **Restore Containers**. The **Restore Containers** wizard opens.

2 Choose the Backup Node on the **Choose Backup Node** screen. This Node is the place where the backups of the Source Node Containers are stored. The **Backup Availability** column in the list of Backup Nodes shows whether any backups have been found on the corresponding Node.

3 On the **Choose Containers to Restore** screen, select the Containers you want to restore from the Backup Node.



Choose Containers to Restore

On this step you should specify one or several Containers to be restored on the destination Hardware Node.

The wizard has detected backups for the following Containers on Hardware Node "Local Server". Please select the Container(s) you wish to restore on server "Local Server":

Name	Source Node	Type	Backup Date
☑ server101	Local Server		
☑ server102	Local Server		

□ Show all available backups

By default, all backups of Containers originally belonging to the Source Node are selected. You can exclude certain Containers from this list, as well as include in it any other backups found on this Backup Node—that is, the backups of those Containers not belonging to the Source Node. To include these other backups, you first need to make them visible by selecting the **Show all available backups** check box.

4. If the Containers to restore exist on the Destination Node, you are presented with the **Resolve Conflicts With Existing Containers** window listing these Containers. When deciding on whether to restore a Container, keep in mind that, during the restore process, all Container current data will be overwritten with data from the corresponding backup.
5. On the **Review Containers Restoration Settings** screen, click the **Finish** button to start restoring the Containers.

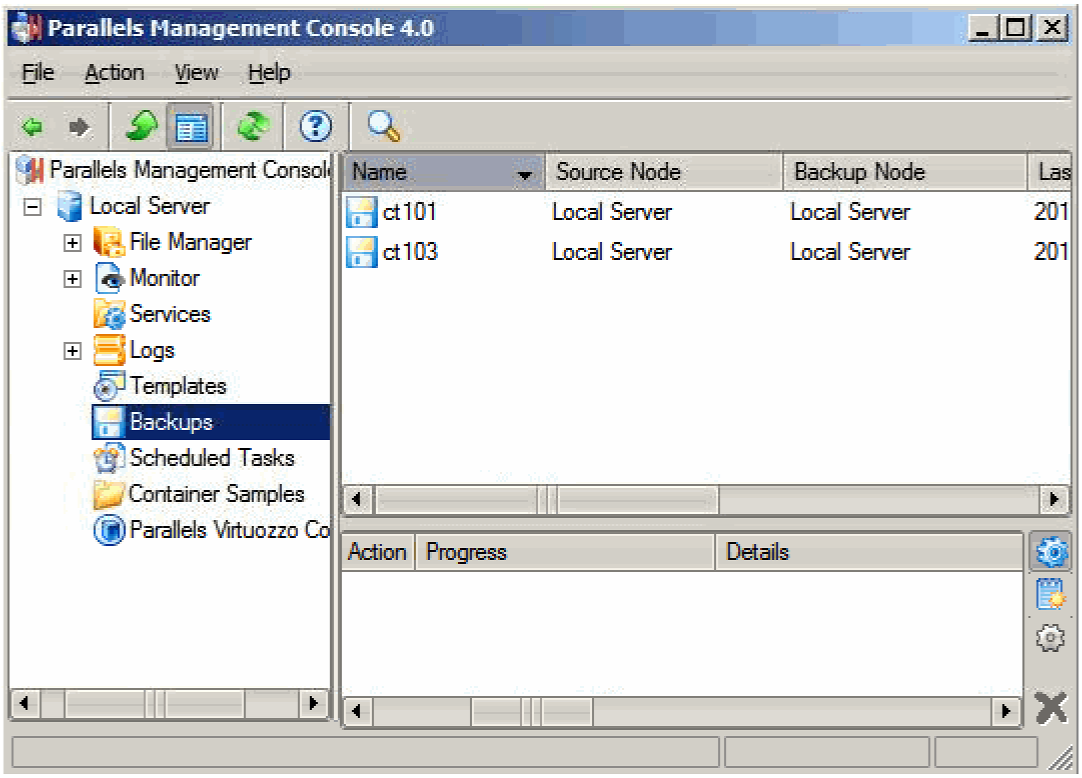
Note: During this operation, all the Containers will be restored to the Source Node—that is, to the Node for which you have invoked the wizard, irrespective of whether the backed up Containers originally belonged to this Source Node or to any other Node.

You can also restore groups of Containers using these tools:

- Parallels Virtual Automation. For more information on this web-based tool, see the *Parallels Virtual Automation Administrator's Guide* at http://www.parallels.com/products/pva46/resources.
- `vzarestore`. Detailed information on this command-line utility is provided in the *Parallels Virtuozzo Containers 4.7 Reference Guide*.

Managing the Backup Node

Any Hardware Node can perform the functions of the Backup Node—that is, store the backups of any Containers of any Hardware Nodes. To see the list of Container backups stored on a Hardware Node, expand its name in the left pane of the Parallels Management Console main window, and click the **Backups** item.



Name	Source Node	Backup Node	Las
ct101	Local Server	Local Server	201
ct103	Local Server	Local Server	201

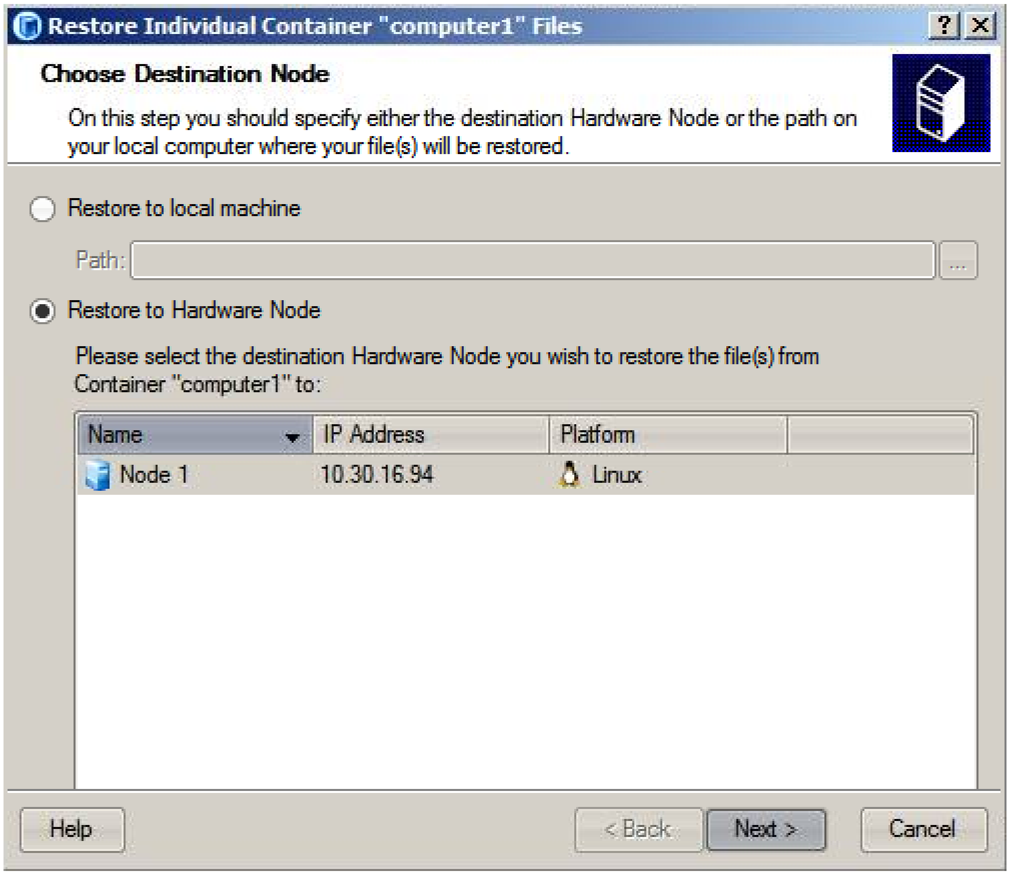
The table in the right pane shows the following information about the backups stored on the given Backup Node:

Column Name	Description
Name	The name of the backed up Container.
Source Node	The Node where the Container was hosted during its backing up.
Last Backup Date	The date and time when the last backing up of the Container took place.
Number of Backups	The number of Container backups on the Node.
Description	The backup description.

The backup manager window allows you to perform the following operations:

- Restore a single Container from its backup. To do this, right-click the needed Container backup, and choose **Restore Container** to start the **Restore Container** wizard. In this wizard, you need to select the Destination Node—the place whither to restore the Container. By default, the Container Source Node is selected. Only the Nodes registered in Parallels Management Console are shown.
- Restore one or several files and/or directories from a particular Container backup. To do this, right-click the Container backup whose files/directories you want to restore, and choose **Restore Individual Container Files** to start the **Restore Individual Container Files** wizard. In this wizard:

  1. Select the Destination Node—the place whither to restore the Container files/directories.



Restore Individual Container "computer1" Files

Choose Destination Node

On this step you should specify either the destination Hardware Node or the path on your local computer where your file(s) will be restored.

○ Restore to local machine

Path:

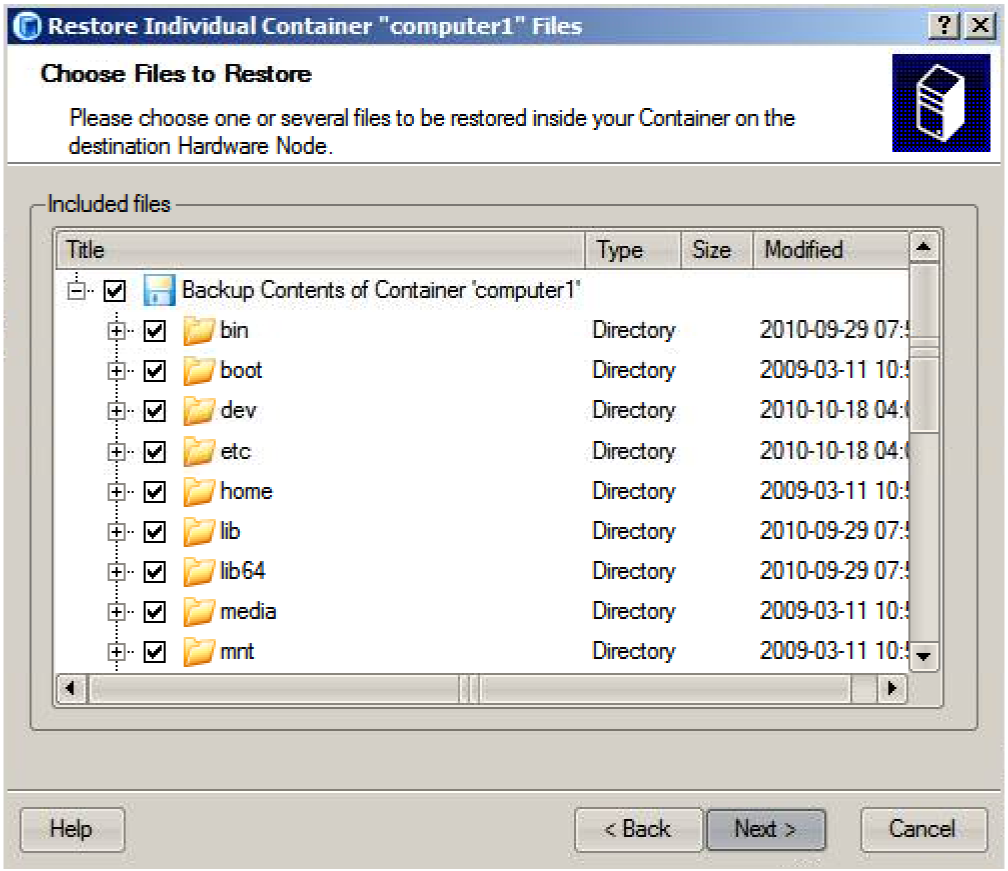
● Restore to Hardware Node

Please select the destination Hardware Node you wish to restore the file(s) from Container "computer1" to:

Name	IP Address	Platform
Node 1	10.30.16.94	Linux

By default, the Container Source Node is selected. Only the Nodes registered in Parallels Management Console are shown. You can also restore the files to your local computer, i.e. to the computer where Parallels Management Console is installed. To do this, select the **Restore to local machine** radio button and, in the **Path** field, specify the path to the folder whither to restore the files.

2. Select the Container files/directories to restore to the Destination Node:



Restore Individual Container "computer1" Files

Choose Files to Restore

Please choose one or several files to be restored inside your Container on the destination Hardware Node.

Included files

Title	Type	Size	Modified
☑ Backup Contents of Container 'computer1'			
☑ bin	Directory		2010-09-29 07:
☑ boot	Directory		2009-03-11 10:
☑ dev	Directory		2010-10-18 04:
☑ etc	Directory		2010-10-18 04:
☑ home	Directory		2009-03-11 10:
☑ lib	Directory		2010-09-29 07:
☑ lib64	Directory		2010-09-29 07:
☑ media	Directory		2009-03-11 10:
☑ mnt	Directory		2009-03-11 10:

The **Choose Files to Restore** window lists all files and directories that you have backed up. To enqueue a file/directory for being restored, select its check box. You can select the check box next to the corresponding directory to restore all the files and subdirectories from this directory.

3. In the **Review Container Restoration Settings** window, you can review the parameters entered in the previous steps of the wizard. If you are satisfied with the specified parameters, click **Finish** to start restoring the selected files/directories. Otherwise, click **Back**, and change the necessary parameters.

Right-clicking a Container backup in the backups list and choosing **Properties** brings about the **Container Backups** dialog where you can do the following:

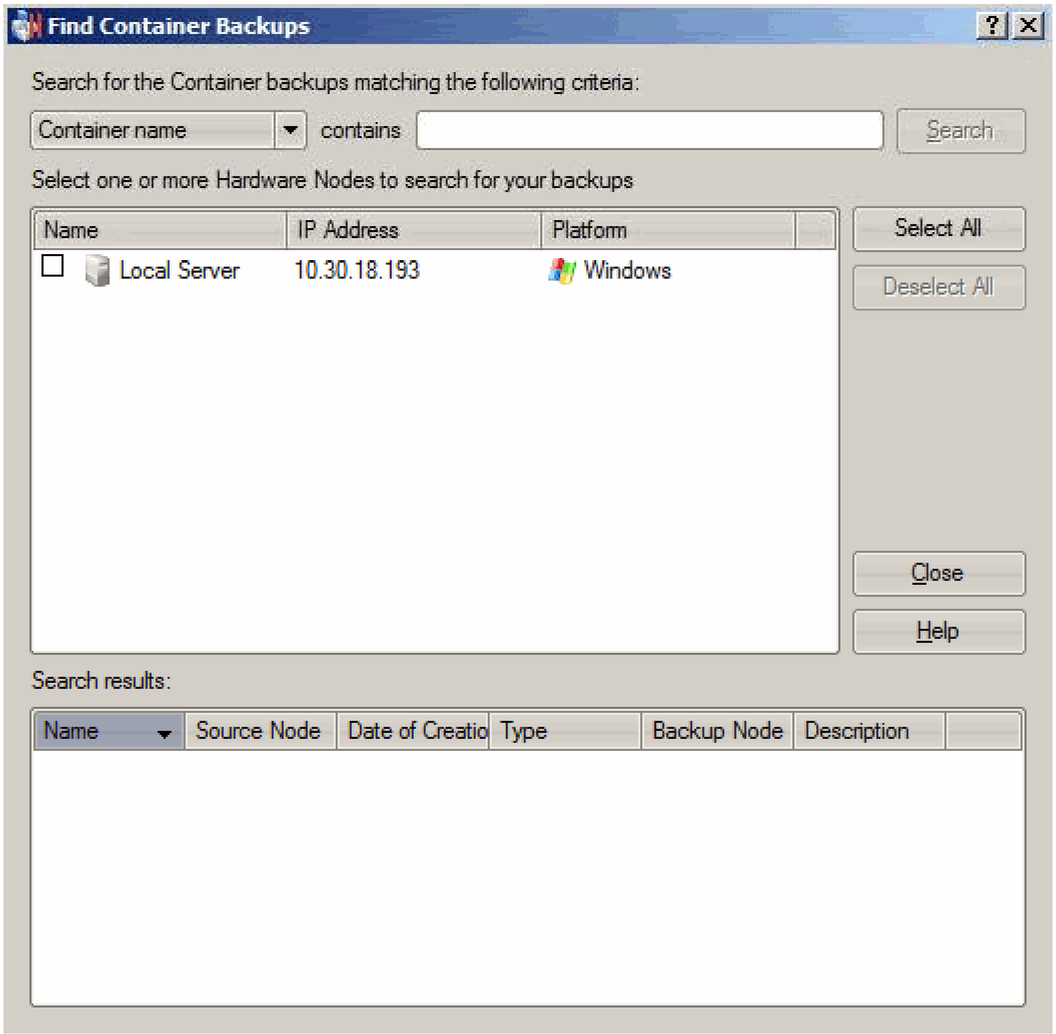
- View extensive information about the selected Container backup including all its full, incremental, and differential backups.
- Delete any of the existing backups.
- Explore the backup contents (the Container files and directories)
- Restore the Container or any of its files/directories.

Searching for Container Backups

If you do not remember the place where you are storing the backup of a particular Container (identified by its ID or its IP address or its hostname or by the date of its creation), you can search for the backup across all the Hardware Nodes registered in Parallels Management Console.

To search for a backup, do the following:

1 Right-click the **Parallels Containers** item under the corresponding Backup Node name, and choose **Backup** > **Search for Backups** to open the **Find Container Backups** dialog:



Search for the Container backups matching the following criteria:

Container name ▾ contains

Select one or more Hardware Nodes to search for your backups

Name	IP Address	Platform
□ Local Server	10.30.18.193	Windows

Search results:

Name	Source Node	Date of Creatio	Type	Backup Node	Description

2 On the upper left drop-down menu, choose the Container parameter by which you want to search for the corresponding Container backup.

3 Enter the value of the parameter in the text field on the right. All the Containers with the corresponding parameter including the specified value as its part will be found. For example, if you enter "100" as the value for Container ID, the backups of Containers 100, 1000, 1001, 1002, 2100, 3100, and so on, will be searched for.

4 Check those Nodes where you want to search for the backups.

5 Click the **Search** button.

The **Search results** table presents the following information about the found backups:

Column Name	**Description**
Name	The name of the Container whose backup has been found.
Source Node	The Node where the Container was hosted during its backing up.
Date of Creation	The date and time when the backup was created.
Type	The backup type. Detailed information on all backup types is given in **Defining Default Backup Type** (p. 71).
Backup Node	The Backup Node - the Node where the backup has been found.
Description	The backup description.

Double-clicking on a Container backup in this table brings about the **Container Backups** dialog where you can view extensive information about the current Container backup, including all its full and incremental backups, as well as delete any of these backups or restore them in the manner depicted above.

Scheduling Container Backups

Parallels Management Console allows you to automate the task of backing up Containers by setting backups to be run on a schedule. For example, you can specify specific time intervals for creating Container backups. You can set a Container to be backed up at different intervals: daily, weekly, monthly. You can also specify a particular day of month for a Container backup to be executed.

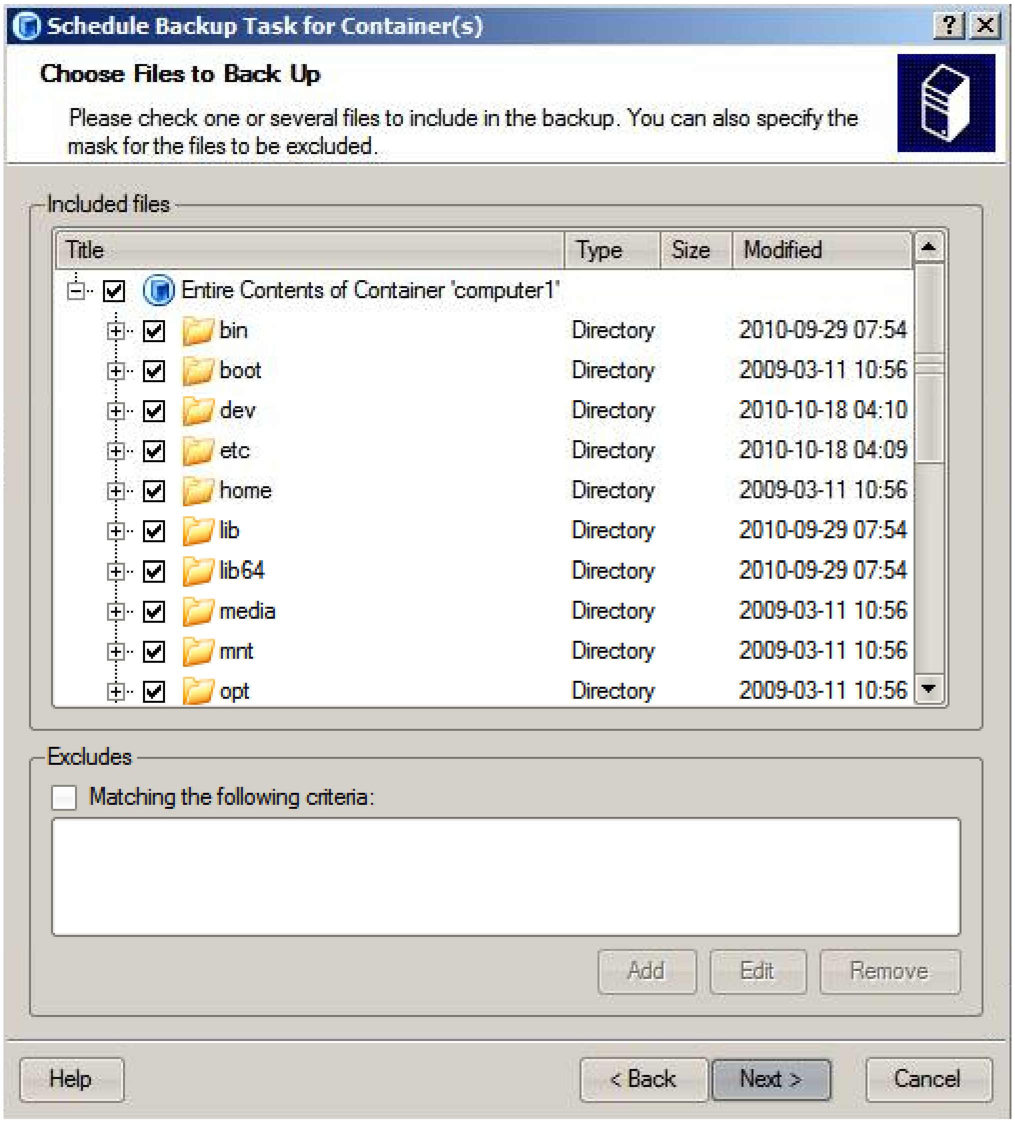
Parallels Management Console provides you with a special wizard—**Schedule Task for Backing Up Containers**—that helps schedule the time for backing up Containers. To launch the wizard, right-click the **Scheduled Tasks** item under the corresponding Hardware Node name, and choose **Schedule New Task** > **Back Up Containers**.

In this wizard, do the following:

1. Choose the Containers to be backed up on the schedule you will set in the following steps of the wizard.To do this, click the **Add** button in the top right corner of the **Choose Containers to Backup Up** window, select the names of the corresponding Containers, and click **OK**. When you are ready, click **Next** to proceed with the wizard.
2. Choose the Container backup mode:
   - **Default**: select this radio button to back up the Container using the default backup mode. When run in this mode, the default backup parameters are used for creating the Container backup. You can only set the backup description and configure the default backup policy.

   Note: Detailed information on what default backup parameters are and how to manage them is given in **Setting Default Backup Parameters** (p. 66).

   - **Custom**: select this radio button to manually set the parameters to be applied to the resulting backup archive. In this case, you will have to go through a number of additional steps (**Steps 3** and **4**) of the **Schedule Backup Task for Container(s)** wizard and set the necessary backup parameters one by one.
3. Specify the files and directories to be included in the backup.



Schedule Backup Task for Container(s)

Choose Files to Back Up

Please check one or several files to include in the backup. You can also specify the mask for the files to be excluded.

Included files

Title	Type	Size	Modified
☑ Entire Contents of Container 'computer1'			
☑ bin	Directory		2010-09-29 07:54
☑ boot	Directory		2009-03-11 10:56
☑ dev	Directory		2010-10-18 04:10
☑ etc	Directory		2010-10-18 04:09
☑ home	Directory		2009-03-11 10:56
☑ lib	Directory		2010-09-29 07:54
☑ lib64	Directory		2010-09-29 07:54
☑ media	Directory		2009-03-11 10:56
☑ mnt	Directory		2009-03-11 10:56
☑ opt	Directory		2009-03-11 10:56

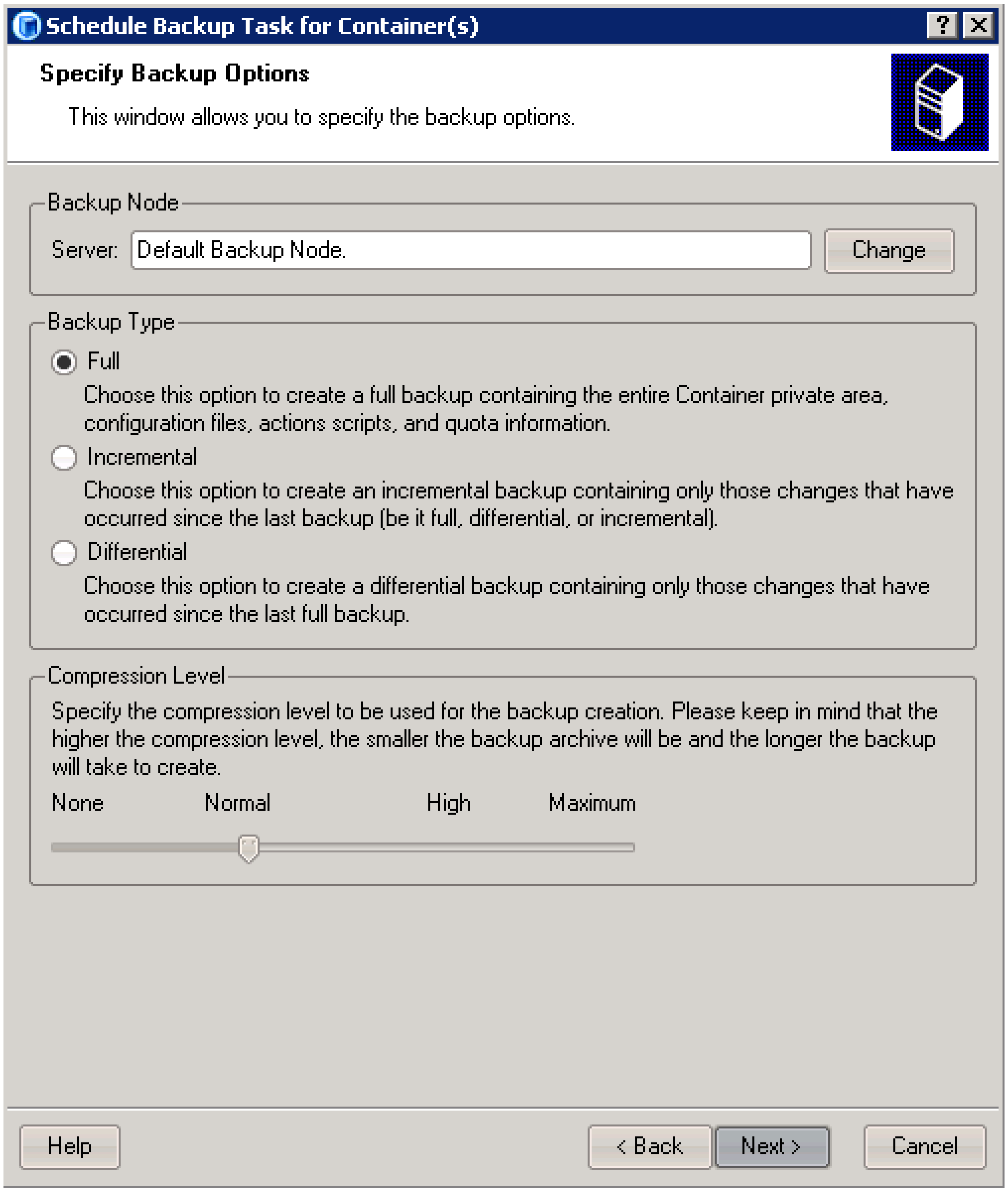
Excludes

□ Matching the following criteria:

By default, all Container files and directories are included in the backup archive. To leave out a file or directory from the backup process, clear its check box in the **Included files** table. You can also select the **Matching the following criteria** check box and use the **Add**/**Edit**/**Remove** buttons to set the parameters to be met by the file/folder to exclude it from the backup process. You can specify the full path to the corresponding file/folder, enter its name, or define any filter compatible with standard Linux masking rules (i.e. with standard globs). For example, you can indicate `/usr/MyDirectory/MyFile.txt` to exclude the `MyFile.txt` file from the backup process or type `*.bmp` to leave out all files with the `bmp` extension.

Note: The **Included files** table is not shown if you are creating a backup task for several Containers.

4 Next, you should specify the main backup parameters.



Schedule Backup Task for Container(s)

Specify Backup Options

This window allows you to specify the backup options.

Backup Node

Server: Default Backup Node.

Backup Type

● Full

Choose this option to create a full backup containing the entire Container private area, configuration files, actions scripts, and quota information.

○ Incremental

Choose this option to create an incremental backup containing only those changes that have occurred since the last backup (be it full, differential, or incremental).

○ Differential

Choose this option to create a differential backup containing only those changes that have occurred since the last full backup.

Compression Level

Specify the compression level to be used for the backup creation. Please keep in mind that the higher the compression level, the smaller the backup archive will be and the longer the backup will take to create.

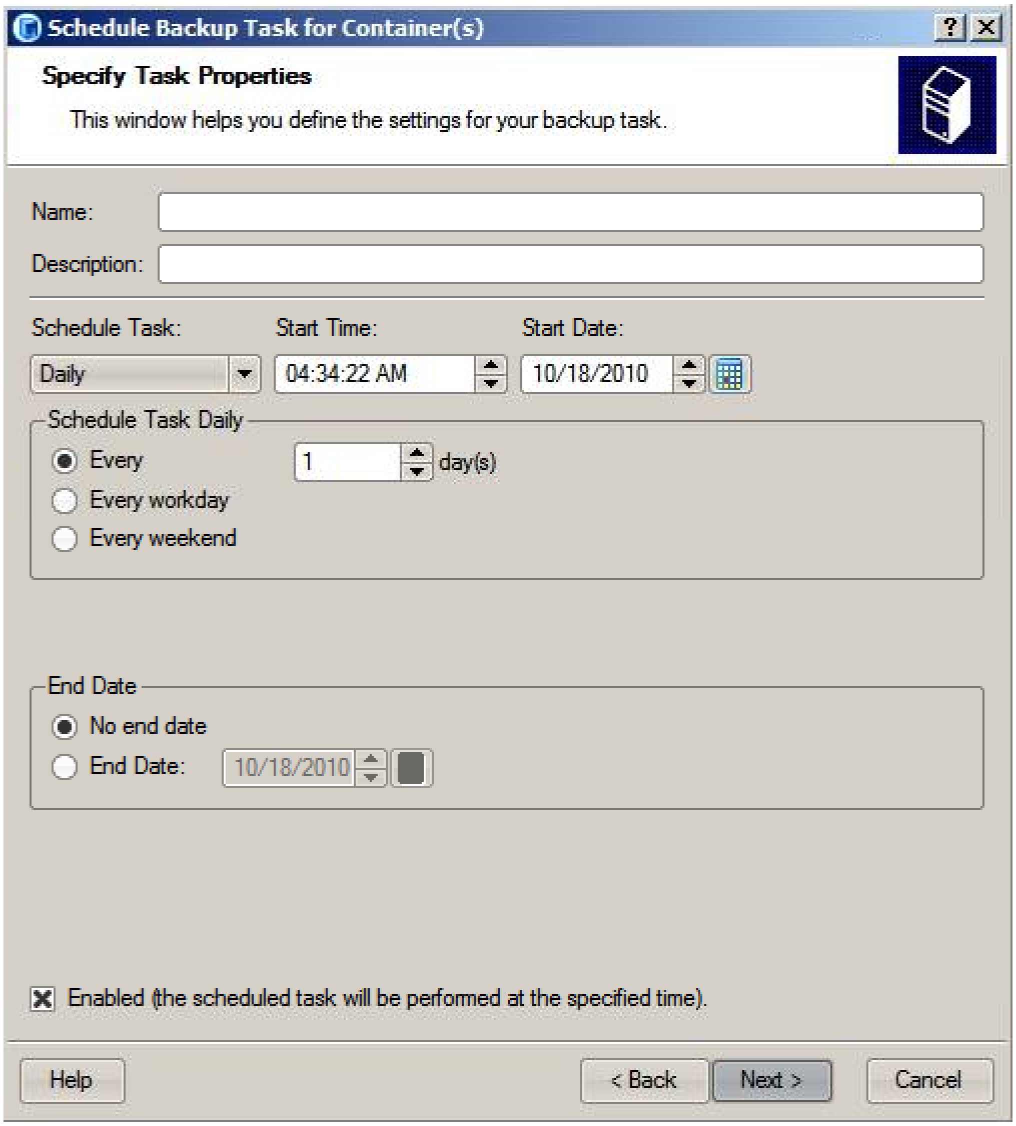
In this window you can configure the following backup parameters:

- *Backup Node*. This Node is the place where the Container backup will be stored. You can leave the Backup Node offered by Parallels Management Console by default or use the **Change** button to specify the desired Backup Node. For detailed information on Backup Nodes, see **Assigning the Default Backup Node** (p. 66).
- *Backup compression level*: 'None', 'Normal', 'High', or 'Maximum'. Detailed information on compression levels is given in **Defining the Default Compression Level** (p. 69).
- *Backup type*. It can be full, incremental, or differential. Detailed information on backup types is provided in **Specifying the Default Backup Type** (p. 71). If you are backing up a single Container, and no backup of this Container has been found on the Backup Node, the **Backup Type** group is not shown, and a full backup is automatically created.

5 In the next step of the wizard, you can set the following backup parameters:

- Provide the backup description in the **Backup description** field, if necessary. The description can be any text containing any backup-related information (for example, the backup purpose).
- Do not stop the Container backup even if any errors appear (select the **Do not stop on errors** check box) or break the backup process if any malfunction occurs (clear the check box).
- Do not stop the backup process if one or more of the Containers to be backed up do not exist on the Source Node (select the **Ignore non-existent Containers**) or break the backup process in this case (clear the check box). You can use this option when backing up several Containers at once.

6 Next, you need specify a number of parameters for the backup task being created.



Schedule Backup Task for Container(s)

Specify Task Properties

This window helps you define the settings for your backup task.

Name:

Description:

Schedule Task: Daily

Start Time: 04:34:22 AM

Start Date: 10/18/2010

Schedule Task Daily

● Every 1 day(s)

○ Every workday

○ Every weekend

End Date

● No end date

○ End Date: 10/18/2010

☒ Enabled (the scheduled task will be performed at the specified time).

In this window, do the following:

- Set the name for the backup task.
- Provide the task description, if necessary.
- Set the schedule for the Container backup (specify the task start time, set the time interval when the Container backup is to be performed, and so on)
- Define the date when the backup task is to be removed from the schedule.

You can also clear the **Enabled** check box if you want to temporarily stop running the scheduled task. At any time, you can enable the task again by right-clicking it and choosing **Enable**.

7 In the last step of the wizard, review the parameters provided in the previous steps of the wizard. If you are satisfied with all the parameters, click **Finish** to schedule the task. Otherwise, click the **Back** button to return to the previous steps and change the necessary parameters.

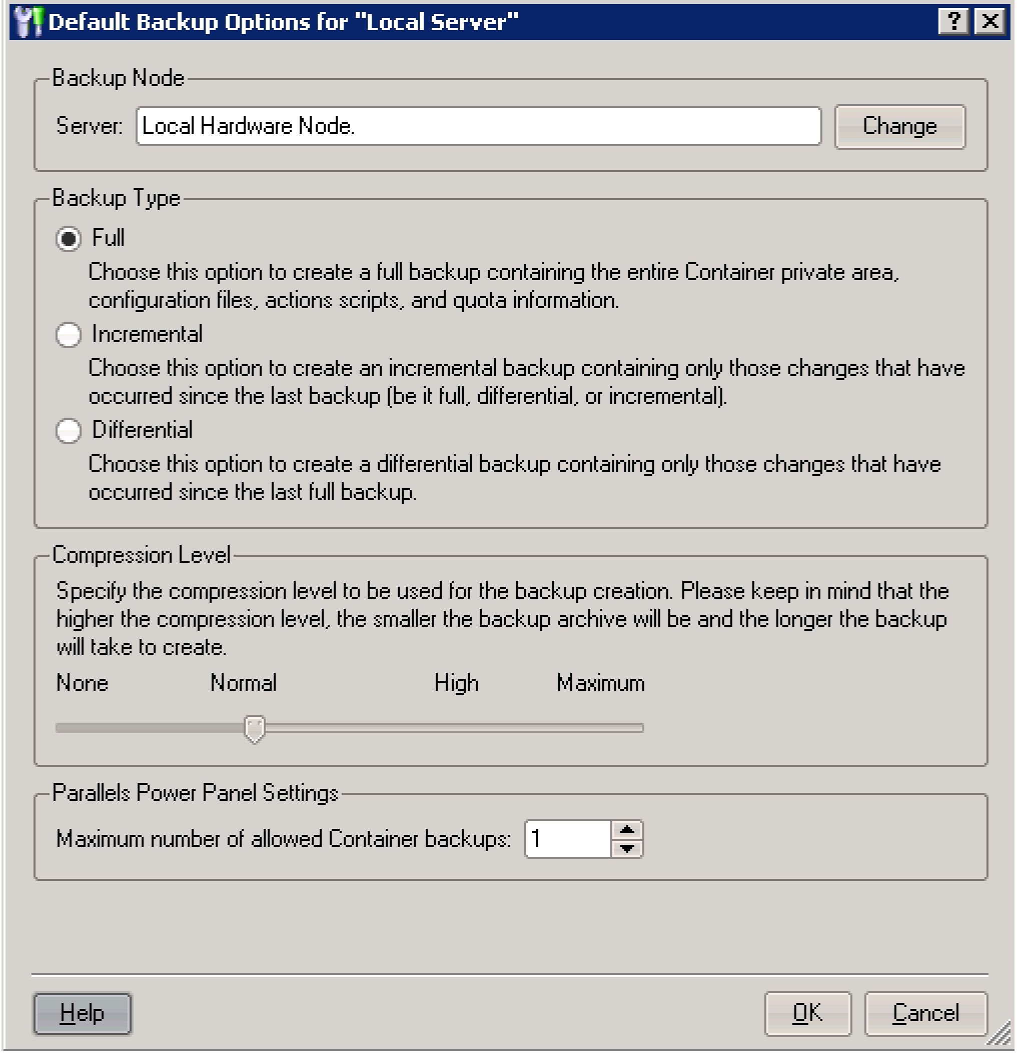
At any time, you can configure any parameters of the scheduled backup task, disable the task, or even delete it. To do this, choose the **Scheduled Tasks** item under the corresponding Hardware Node name, right-click the corresponding backup task in the Management Console right pane, and choose one of the following options:

- **Disable** to temporarily stop backing up Containers on the set schedule.
- **Delete** to permanently remove the scheduled backup task.
- **Properties** to change the settings of the backup task.

Setting the Maximum Number of Backups for Parallels Power Panel

Parallels Management Console allows you to configure the number of backups Container administrators are allowed to create on the given Hardware Node using Parallels Power Panel. By default, any Container administrator is allowed to create only one Container backup in Parallels Power Panel. However, you can increase the number of allowed backups by doing the following:

1 Right-click the Hardware Node where the Container for which you want to increase the number of allowed backups is residing, and choose **Backup** > **Set Default Backup Options**.



Default Backup Options for "Local Server"

Backup Node

Server: Local Hardware Node.

Backup Type

● Full

Choose this option to create a full backup containing the entire Container private area, configuration files, actions scripts, and quota information.

○ Incremental

Choose this option to create an incremental backup containing only those changes that have occurred since the last backup (be it full, differential, or incremental).

○ Differential

Choose this option to create a differential backup containing only those changes that have occurred since the last full backup.

Compression Level

Specify the compression level to be used for the backup creation. Please keep in mind that the higher the compression level, the smaller the backup archive will be and the longer the backup will take to create.

Parallels Power Panel Settings

Maximum number of allowed Container backups: 1

2 Specify the number of Container backups the Container administrator will be able to create with Parallels Power Panel by typing the desired number in the **Maximum number of allowed Container backups** field or using the spin button.

3 Click **OK**.

Keep in mind that the limit set on the number of Container backups concerns only the process of backing up Containers using the Parallels Power Panel tool. There are no restrictions for any users creating Container backups by means of other Parallels Virtuozzo Containers tools (for example, Parallels Virtual Automation or Parallels Management Console); they are allowed to create as many Container backups as they want to.

Reinstalling Containers

Reinstalling a Container is used if a Container administrator has inadvertently modified, replaced, or deleted any file that is part of an application or OS template, which has brought about the Container malfunction. You can reinstall the Container using one of the following commands:

vzctl recover

The `vzctl recover` command restores the original VZFS symlinks of the Container private area to the OS and/or application template(s) as they were at the time when the Container was created and/or when the application template(s) were added to the Container. This command does not deal with any user files on the Container:

```
# vzctl recover 101
Optimizing Container private area...
...
Recovering Container completed successfully
```

vzctl reinstall

The `vzctl reinstall` command creates a new private area for the problem Container from scratch using its configuration files and its OS and application templates. Thus, a clean working copy of the Container is created:

```
# vzctl reinstall 101
Optimizing Container private area...
...
Container reinstallation completed successfully
```

Note: If any of the Container application templates cannot be added to the Container in a normal way, the reinstallation process will fail. This may happen, for example, if an application template was added to the Container using the `--force` option of the `vzpkgadd` or `vzpkg install` command (for more information on these commands, see the *Parallels Virtuozzo Containers 4.7 Reference Guide*).

In order to retain the personal data inside the old Container, the utility also copies the contents of the old private area to the `/old` directory of the new private area (unless the `--skipbackup` option is given). The personal data can then be copied to the corresponding directories of the new private area and the `/old` directory eventually deleted:

```
# vzctl start 101
# vzctl exec 101 ls /
bin
boot
dev
[...other directories...]
old
[...other directories...]
tmp
usr
var
```

Both the `vzctl recover` and `vzctl reinstall` commands retain the users' credentials base, unless the `--resetpwdb` option is specified.

Customizing Container Reinstallation

The default reinstallation, as performed by the `vzctl reinstall` command, creates a new private area for the broken Container as if it were created by the `vzctl create` command and copies the private area of the broken Container to the `/old` directory in the new private area so that no file is lost. There is also a possibility of deleting the old private area altogether without copying or mounting it inside the new private area, which is done by means of the `--skipbackup` option. This way of reinstalling corrupted Containers might in certain cases not correspond exactly to your particular needs. It happens when you are accustomed to creating new Containers in some other way than just using the `vzctl create` command. For example, you may install additional software licenses into new Containers, or anything else. In this case you would naturally like to perform reinstallation in such a way so that the broken Container is reverted to its original state as determined by you, and not by the default behavior of the `vzctl create` command.

To customize reinstallation, you should write your own scripts determining what should be done with the Container when it is being reinstalled, and what should be configured inside the Container after it has been reinstalled. These scripts should be named `vps.reinstall` and `vps.configure`, respectively, and should be located in the `/etc/vz/conf` directory on the Node. To facilitate your task of creating customized scripts, the Containers software is shipped with sample scripts that you may use as the basis of your own scripts.

When the `vzctl reinstall` *`<CT_ID>`* command is called, it searches for the `vps.reinstall` and `vps.configure` scripts and launches them consecutively. When the `vps.reinstall` script is launched, the following parameters are passed to it:

`--veid` The ID of the Container.

`--ve_private_tmp` The path to the Container temporary private area. This path designates where a new private area is temporarily created for the Container. If the script runs successfully, this private area is mounted to the path of the original private area after the script has finished.

`--ve_private` The path to the Container original private area.

You may use these parameters within your `vps.reinstall` script.

If the `vps.reinstall` script finishes successfully, the Container is started, and the `vps.configure` script is called. At this moment the old private area is mounted to the `/old` directory inside the new one irrespective of the `--skipbackup` option. This is done in order to let you use the necessary files from the old private area in your script, which is to be run inside the running Container. For example, you might want to copy some files from there to regular Container directories.

After the `vps.configure` script finishes, the old private area is either dismounted and deleted or remains mounted depending on whether the `--skipbackup` option was provided.

If you do not want to run these reinstallation scripts and want to stick to the default `vzctl reinstall` behavior, you may do either of the following:

1 Remove the `vps.reinstall` and `vps.configure` scripts from the `/etc/vz/conf` directory, or at least rename them;

2 Modify the last line of the `vps.reinstall` script so that it would read

```
exit 128
```

instead of

```
exit 0
```

The `128` exit code tells the utility not to run the scripts and to reinstall the Container with the default behavior.

Deleting Containers

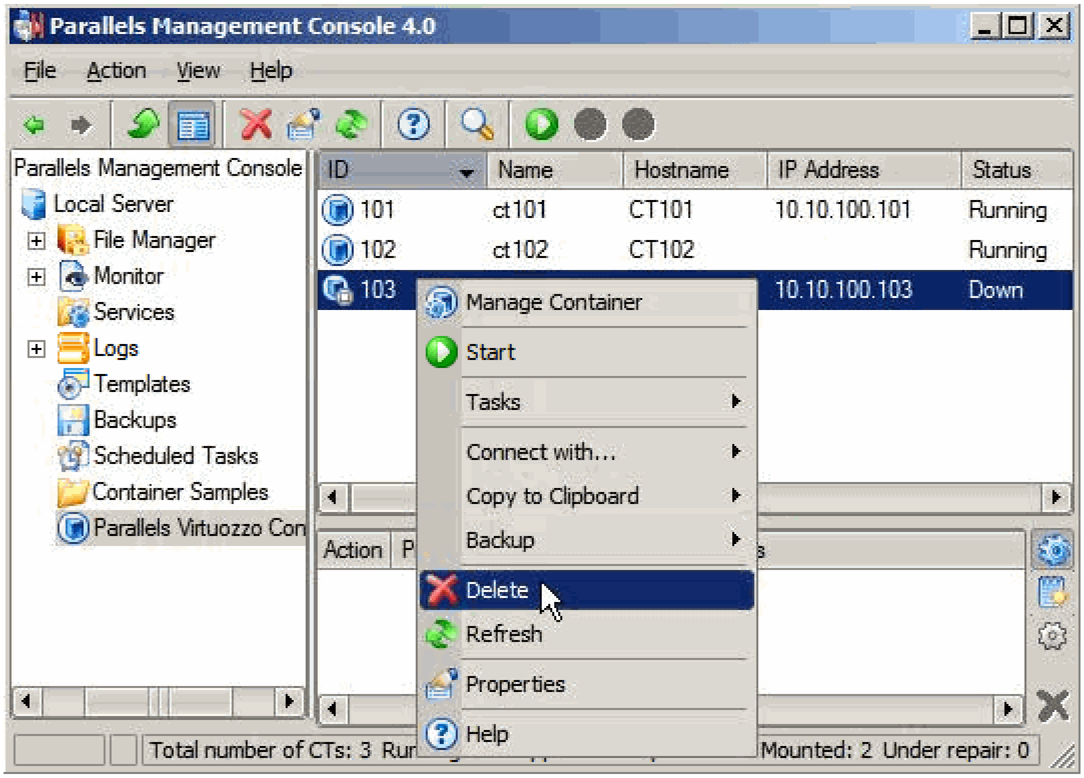
You can delete a Container that is not needed anymore with the `vzctl destroy <CT_ID>` command. This command removes the Container private area completely and renames the Container configuration file and action scripts by appending the `.destroyed` suffix to them.

Note: You can also use the `vzctl delete` command to remove Containers from the Hardware Node. This command has the syntax identical to that of `vzctl destroy`.

A running Container cannot be destroyed with the `vzctl destroy` command. The example below illustrates destroying Container 101:

```
# vzctl destroy 101
Container is currently running.
Stop it before proceeding.
# vzctl stop 101
Stopping Container ...
Container was stopped
Container is unmounted
# vzctl destroy 101
Destroying Container private area: /vz/private/101
Container private area was destroyed
# vzctl status 101
VEID 101 deleted unmounted down
```

Parallels Management Console allows you to delete Containers that are not needed anymore. To delete a Container, select it in the **Containers** table in the right pane of the Management Console main window. You can use CTRL+Click to select or deselect an entry, SHIFT+Click to select a range of Containers, CTRL+A to select all Containers. Then right-click the selected Containers, and choose **Delete**.



ID	Name	Hostname	IP Address	Status
101	ct101	CT101	10.10.100.101	Running
102	ct102	CT102		Running
103			10.10.100.103	Down

Total number of CTs: 3 Rur

Mounted: 2 Under repair: 0

You can also click the **Delete** button on the toolbar or select **Delete** on the **Action** menu. In the displayed dialog, click **Yes** to confirm your decision.

Deleting a considerable number of Containers may take some time. The progress is displayed in the **Actions** pane.

Disabling Containers

There may appear situations when you need to forbid Container owners to use their Containers. For example, it may happen if the Container owner uses it for unallowed purposes: intruding into computers of other users, participating in DoS attacks, and so on.

In such cases, you can disable a Container, thus making it impossible to start the Container once it was stopped. For example, you can execute the following command to disable Container 101:

```
# vzctl set 101 --disabled yes
```

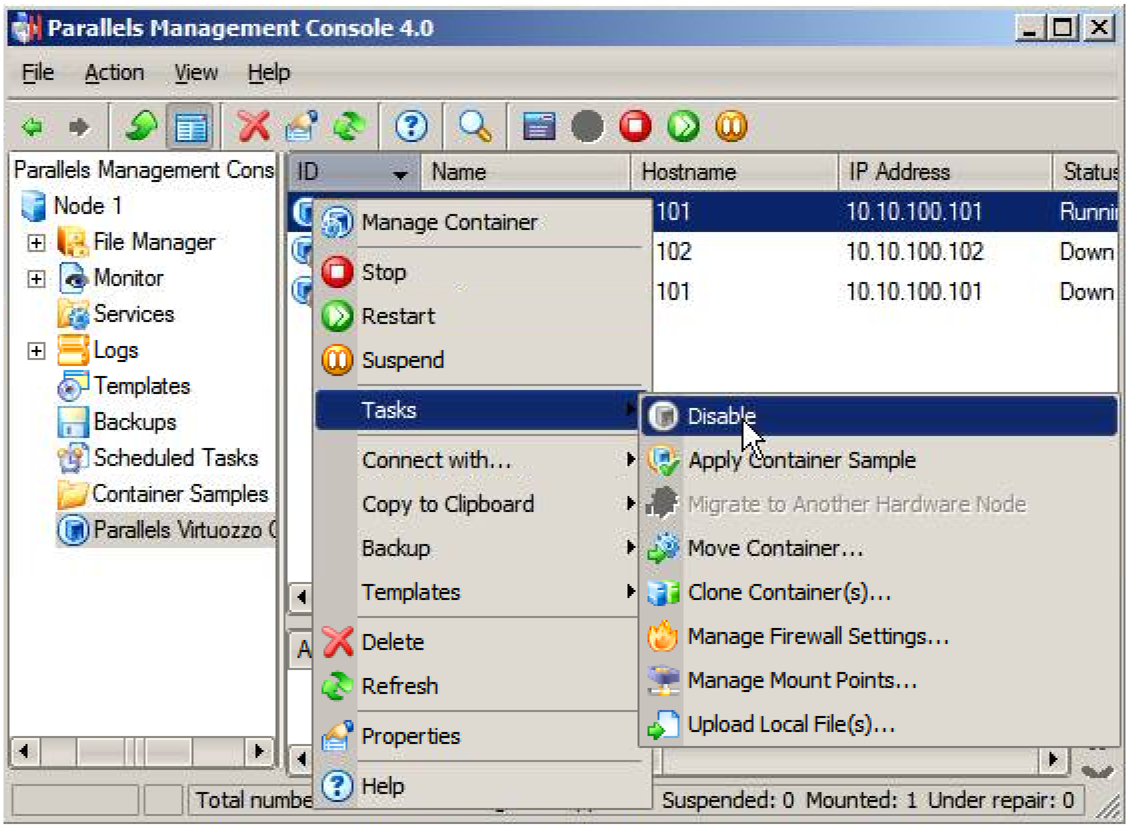
Once Container 101 is stopped, the user will not be able to start it again until you enable the Container again:

```
# vzctl set 101 --disabled no
```

You can also use the `--force` option to start a disabled Container. For example:

```
# vzctl start 101
Container start disabled
# vzctl start 101 --force
Starting Container...
Container is mounted
Adding port redirection to Container(1): 4643 8443
Adding IP address(es): 10.144.144.101
Hostname for Container set: Container_101
Container start in progress...
```

You can also disable and enable Containers using Parallels Management Console. To do this, click the **Parallels Virtuozzo Containers** item under the corresponding Hardware Node name, right-click the Container you want to disable or enable, and choose **Tasks** > **Disable** or **Tasks** > **Enable**.



ID	Name	Hostname	IP Address	Status
		101	10.10.100.101	Runnir
		102	10.10.100.102	Down
		101	10.10.100.101	Down

Suspended: 0 Mounted: 1 Under repair: 0

You can use CTRL+Click to select or deselect an entry, SHIFT+Click to select a range of Containers, CTRL+A to select all Containers.

Suspending Containers

Parallels Virtuozzo Containers allows you to suspend any running Container on the Hardware Node by saving its current state to a special dump file. Later on, you can resume the Container and get it in the same state the Container was at the time of its suspending.

In Parallels Virtuozzo Containers-based systems, you can use the `vzctl suspend` command to save the current state of a Container. For example, you can issue the following command to suspend Container 101:

```
# vzctl suspend 101
Setup checkpoint ...
Container is unmounted
Checkpointing completed successfully
```

During the command execution, the `/vz/private/101/dump/Dump` file containing the entire state of Container 101 is created and the Container itself is stopped.

> **Note:** You can set another directory to store dump files for your Containers by changing the value of the `DUMPDIR` parameter in the global file. Detailed information on the global file and the parameters you can specify in it is provided in the *Parallels Virtuozzo Containers 4.7 Reference Guide*.

In Parallels Management Console, you can suspend a running Container by doing the following:

1. Select the **Parallels Virtuozzo Containers** item under the corresponding Hardware Node name in the Parallels Management Console left pane.
2. In the right pane, right-click the Container you want to suspend, and choose **Suspend**.
3. Confirm the operation by clicking **Yes** in the displayed window.

At any time, you can resume Container 101 by executing the following command:

```
# vzctl resume 101
Starting Container ...
Container is mounted
Adding port redirection to Container(1): 4643 8443
Adding IP address(es): 10.0.10.101
Container start in progress...
```

The Container state is restored from the `/vz/private/101/dump/Dump` file on the Node. Upon the restoration completion, any applications that were running inside Container 101 at the time of its suspending will be running and the information content will be the same as it was when the Container was suspended.

To restore a suspended Container in Parallels Management Console:

1. Select the **Parallels Virtuozzo Containers** item under the corresponding Hardware Node name in the Parallels Management Console left pane.
2. In the right pane, right-click the Container you want to restore, and choose **Resume**.

When working with dump files, keep in mind the following:

- You can both restore the Container dump file on the Source Node, i.e. on the Node where this Container was running before its dumping, or transfer the dump file to another Node and restore it there.

  > **Note:** Before restoring a Container from its dump file, make sure that the file system on the Destination Node is identical to that at the moment of the Container dumping; otherwise, the Container restoration may fail.

- You can use the file manager to view the files and directories inside the suspended Container. However, you cannot change any of the files and directories since it may cause the Container to resume improperly.
- You can reinstall the suspended Container.
- You can back up the suspended Container.
- You can restore the suspended Container from its backup. After restoring the Container, it is brought to the 'suspended' state again.
- You cannot clone the suspended Container.
- You cannot change the ID of the suspended Container.
- You cannot change network settings of the suspended Container.
- You cannot perform operations on the users' accounts inside the suspended Container.
- You cannot repair the suspended Container.

Running Commands in Containers

Usually, a Container administrator logs in to the Container via network and executes any commands in the Container as on any other Linux box. However, you might need to execute commands in Containers bypassing the normal login sequence. This can happen when:

- You do not know the Container login information but need to run some diagnosis commands in a Container to verify that it is operational.
- Network access is absent for a Container. For example, the Container administrator might have accidentally applied incorrect firewalling rules or stopped the SSH daemon.

Parallels Virtuozzo Containers allows you to execute commands in a Container in both these cases. The session below illustrates the situation when the SSH daemon is not started in Container 101:

```
# vzctl exec 101 /etc/init.d/sshd status
sshd is stopped
# vzctl exec 101 /etc/init.d/sshd start
Starting sshd:[  OK  ]
# vzctl exec 101 /etc/init.d/sshd status
sshd (pid 26187) is running...
```

Now Container users can log in to Container 101 via SSH.

When executing commands in a Container from shell scripts, use the `vzctl exec2` command. It has the same syntax as `vzctl exec` but returns the exit code of the command being executed instead of the exit code of `vzctl` itself. You can check the exit code to find out whether the command has completed successfully.

If you want to execute a command in all running Containers, you can use the following script:

```
# for i in `cat /proc/vz/veinfo | awk '{print $1}'|egrep -v '^0$'`; \
do echo "Container $i"; vzctl exec $i <command>; done
```

where *`<command>`* is the command to be executed. For example:

```
# for i in `cat /proc/vz/veinfo | awk '{print $1}'|egrep -v '^0$'`; \
do echo "Container $i"; vzctl exec $i uptime; done
Container 1
  2:26pm  up 6 days,  1:28,  0 users,  load average: 0.00, 0.00, 0.00
Container 101
  2:26pm  up 6 days,  1:39,  0 users,  load average: 0.00, 0.00, 0.00
[The rest of the output is skipped...]
```

Updating Containers

In Parallels Virtuozzo Containers, you can do one of the following to keep Containers up to date:

- Update EZ templates software packages in a particular Container by means of Parallels Management Console or the `vzpkg` utility. By doing so, you can keep any of your Containers up to date.
- Update the caches of OS EZ templates installed on the Hardware Node. This facility allows you to create new Containers already having the latest software packages installed.

Updating EZ Template Packages In Containers

Parallels Virtuozzo Containers allows you to update packages of the OS EZ template a Container is based on and of any application EZ templates applied to the Container. You can do it by using the `vzpkg update` utility. Assuming that Container 101 is based on the `redhat-el5-x86` OS EZ template, you can issue the following command to update all packages included in this template:

```
# vzpkg update 101 redhat-el5-x86
...
  Updating: httpd                ####################### [1/4]
  Updating: vzdev                ####################### [2/4]
  Cleanup : vzdev                ####################### [3/4]
  Cleanup : httpd                ####################### [4/4]

Updated: httpd.i386 0:2.0.54-10.2 vzdev.noarch 0:1.0-4.swsoft
Complete!
Updated:
 httpd                   i386      0:2.0.54-10.2
 vzdev                   noarch    0:1.0-4.swsoft
```

Notes:

1. A Container has to be running in order to update EZ templates inside this Container.

2. If you are going to update the cache of a commercial OS EZ template (e.g. Red Hat Enterprise Server 5 or SLES 10), you should first update software packages in the remote repository used to handle this OS EZ template and then proceed with updating the EZ template cache. Detailed information on how to manage repositories for commercial Linux distributions is provided in the *Parallels Virtuozzo Containers 4.7 Templates Management Guide*.

As you can see from the example above, the `httpd` and `vzdev` applications have been updated for the `redhat-el5-x86` OS EZ template. If you wish to update all EZ templates (including the OS EZ template) inside Container 101 at once, you should execute the following command:

```
# vzpkg update 101
...
Running Transaction
  Updating   : hwdata              ###################### [1/2]
  Cleanup    : hwdata              ###################### [2/2]

Updated: hwdata.noarch 0:1.0-3.swsoft
Complete!
Updated:
 hwdata                  noarch    0:0.158.1-1
```

In the example above, only the `hwdata` package inside Container 101 was out of date and updated to the latest version.

In Parallels Management Console, you can do the following to update the OS EZ template a Container is based on and/or any of its application EZ templates:

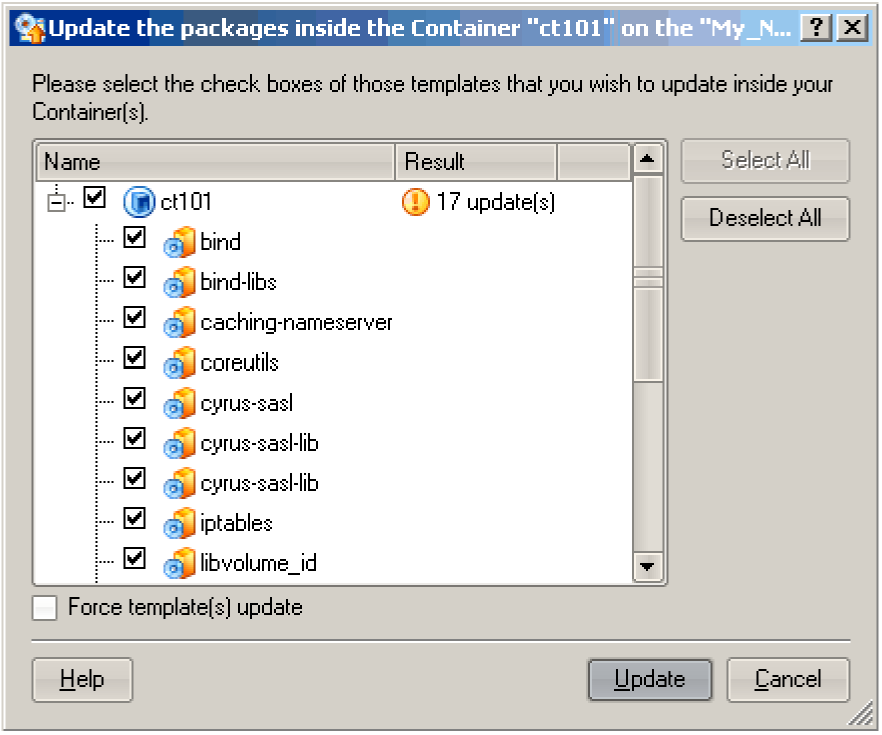
1 Open the list of Containers by selecting the **Parallels Virtuozzo Containers** item in the Hardware Node tree.

2 Double-click the Container where you want to add an EZ template. The Container Manager opens.

3 Click the **Templates** item in the main tree of the Container Manager**.**

4 In the right pane, click either the **OS Templates** or **Application Templates** tab depending on which EZ template you want to update.

5 Right-click the corresponding EZ template, and choose **Update Installed Packages**.


Update the packages inside the Container "ct101" on the "My_N...
Select All
Deselect All
Help
Update
Cancel

Please select the check boxes of those templates that you wish to update inside your Container(s).

Name	Result
☑ ct101	17 update(s)
☑ bind	
☑ bind-libs	
☑ caching-nameserver	
☑ coreutils	
☑ cyrus-sasl	
☑ cyrus-sasl-lib	
☑ cyrus-sasl-lib	
☑ iptables	
☑ libvolume_id	

□ Force template(s) update

This window displays all packages included in all EZ templates (both OS and application) that are applied to the Container.

6 Select the check boxes of the packages to update, and click the **Update** button. You can use the **Select All** and **Deselect All** buttons to select/deselect all packages included in your EZ templates. On this screen, you can also select the **Force template(s) installation** check box to force the EZ template installation inside the Container. In this case no dependencies and no available versions of the application EZ template will be checked during its installation, which may cause the application EZ template to malfunction.

Updating OS EZ Template Caches

With the release of new updates for the corresponding Linux distribution, the created OS EZ template cache can become obsolete. So, Parallels Virtuozzo Containers provides the `vzpkg update cache` command allowing you to quickly update any of the OS EZ template caches available on the Hardware Node.

> **Note:** If you are going to update the cache of a commercial OS EZ template (e.g. Red Hat Enterprise Server 5 or SLES 10), you should first update software packages in the remote repository used to handle this OS EZ template and then proceed with updating the EZ template cache. Detailed information on how to manage repositories for commercial Linux distributions is provided in the *Parallels Virtuozzo Containers 4.7 Templates Management Guide*.

When executed, `vzpkg update cache` checks the `cache` directory in the template area (by default, the template area is located in `/vz/template`) on the Hardware Node and updates all existing tarballs in this directory. However, you can explicitly indicate the tarball for what OS EZ template should be updated by specifying the OS EZ template name. For example, to update the tarball for the `fedora-core-13-x86` OS EZ template, you can run the following command:

```
# vzpkg update cache fedora-core-13-x86
Loading "rpm2vzrpm" plugin
Setting up Update Process
Setting up repositories
base0            100% |=========================|  951 B    00:00
base1            100% |=========================|  951 B    00:00
base2            100% |=========================|  951 B    00:00
base3            100% |=========================|  951 B    00:00
...
```

Upon the `vzpkg update cache` execution, the old tarball is renamed by receiving the `-old` suffix (for example, `fedora-core-13-x86.tar.gz-old`):

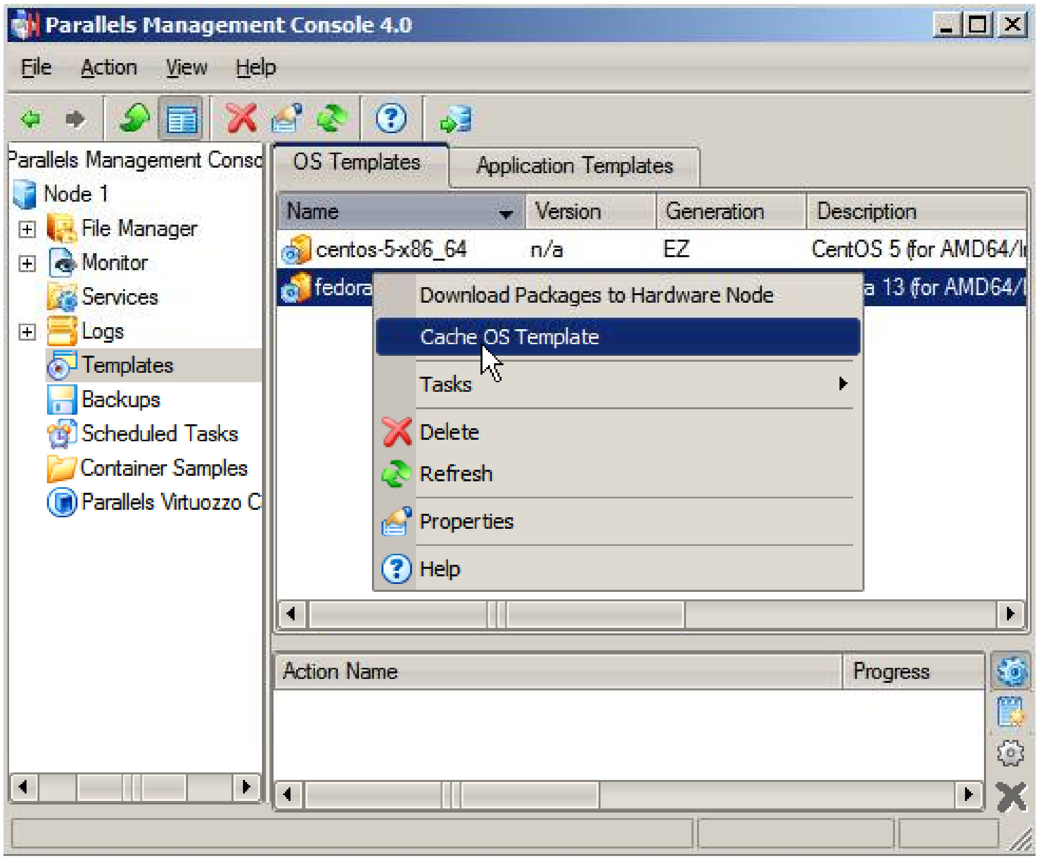
```
# ls /vz/template/cache
fedora-core-13-x86.tar.gz  fedora-core-13-x86.tar.gz-old
```

You can also pass the `-f` option to `vzpkg update cache` to remove an existing tar archive and create a new one instead of it.

If the `vzpkg update cache` command does not find a tarball for one or several OS EZ templates installed on the Node, it creates tar archives of the corresponding OS EZ templates and puts them to the `/vz/template/cache` directory.

To update an OS EZ template cache in Parallels Management Console, do the following:

1. Select the **Templates** item under the corresponding Hardware Node name in the Parallels Management Console left tree.
2. In the right pane, click the **OS Templates** tab to display the list of OS templates installed on the Node.
3. Right-click the template you want to cache in the right pane, and choose **Cache OS Template**.



Name	Version	Generation	Description
centos-5-x86_64	n/a	EZ	CentOS 5 (for AMD64/I
fedora			a 13 (for AMD64/I

CHAPTER 4

Managing Resources

The main goal of resource control in Parallels Virtuozzo Containers 4.7 is to provide Service Level Management or Quality of Service for Containers. Correctly configured resource control settings prevent serious impacts resulting from the resource overusage (accidental or malicious) of a Container on the other Containers. Using resource control parameters for resources management also allows to enforce fairness of resource usage among Containers and better service quality for preferred Containers, if necessary.

In This Chapter

What are Resource Control Parameters?

The system administrator controls the resources available to a Container through a set of resource management parameters. All these parameters are defined either in the global configuration file (`/etc/vz/vz.conf`), or in the respective Container configuration files (`/etc/vz/conf/`*`CT_ID`*), or in both. You can set them by manually editing the corresponding configuration files, by using Parallels command-line utilities, Parallels Virtual Automation, or Parallels Management Console. These parameters can be divided into the disk, network, CPU, and system categories. The table below summarizes these groups:

Group	Description	Parameter names	Explained in
Disk	This group of parameters determines disk quota and I/O operations in Parallels Virtuozzo Containers.	`DISK_QUOTA, DISKSPACE, DISKINODES, QUOTATIME, QUOTAUGIDLIMIT, IOPRIO, IOLIMIT, IOPSLIMIT`	*Managing Disk Quotas* and *Managing Disk I/O Parameters*
Network	This group of parameters determines network traffic settings for Containers.	`TRAFFIC_SHAPING, BANDWIDTH, TOTALRATE, RATE, RATEBOUND`	*Managing Network Accounting and Bandwidth*
CPU	This group of parameters defines the CPU time Containers are guaranteed to receive.	`VE0CPUUNITS, CPUUNITS, CPUS, BURST_CPULIMIT, BURST_CPU_AVERAGE_USAGE`	*Managing Container CPU Resources*
System	This group of parameters allows you to configure and control all memory-related parameters in Containers.	`PHYSPAGES, SWAPPAGES`	*Managing VSwap Parameters*

Managing Container CPU Resources

The current section explains the CPU resource parameters that you can configure and monitor for Containers. The table below provides the name and the description for the CPU parameters.

Parameter	Description
`cpuunits`	Positive integer number that determines how much time one Container can receive in comparison with the other Containers on the Node.
`cpus`	Number of CPUs to use for handling the processes running in a Container.
`cpulimit`	Positive number that indicates the CPU time, in percent, a Container is not allowed to exceed.
`burst_cpu_avg_usage`	CPU usage limit, in percent, used by the Parallels Agent software when controlling the CPU consumption of all running Containers.
`burst_cpulimit`	CPU power limit, in percent, a Container cannot exceed. The limitations set in this parameter are applied to the Container when it exceeds the limit specified in the `burst_cpu_avg_usage` parameter.
`cpumask`	CPU affinity mask that defines on which CPUs the processes in a Container may be executed.
`nodemask`	CPU mask that specifies the NUMA nodes where the processes of a Container may be executed.

Configuring CPU Units

CPU units define how much CPU time one Container can receive in comparison with the other Containers on the Hardware Node if all the CPUs of the Node are fully used. For example, if Containers 101 and 103 are set to receive 1000 CPU units each and Container 102 is configured to get 2000 CPU units, Container 102 will get twice as much CPU time as Containers 101 or 103 if all the CPUs of the Node are completely loaded.

By default, each Container on the Node gets 1000 CPU units. You can configure the default setting using the `pctl set` command. For example, you can run the following command to allocate 2000 CPU units to Container 101:

```
# pctl set 101 --cpuunits 2000 --save
Saved parameters for Container
```

Configuring Number of CPUs

If your Node has multiple CPU cores, you can configure the number of CPU cores to use for executing the processes running in Containers. The CPU cores allocated to a Container are called virtual CPUs to distinguish them from the physical CPUs installed on the Hardware Node.

By default, a Container is allowed to consume the CPU time of all CPU cores available on the Node. That means that any process in any Container can be executed on any CPU core. You can, however, configure the number of physical CPU cores that will be available to a Container using the `--cpus` option of the `vzctl set` command. For example, if your Node has 4 CPU cores installed, you can set the processes in Container 101 to be executed on 2 CPUs only by running this command:

```
# pctl set 101 --cpus 2 --save
Saved parameters for Container 101
```

Note: The number of virtual CPUs for a Container must not exceed the number of physical CPU cores installed on the Hardware Node.

To make sure that only two CPUs are now available to Container 101, run this command on the Node:

```
# pctl exec 101 cat /proc/cpuinfo
processor       : 0
vendor_id       : GenuineIntel
cpu family      : 15
model           : 4
model name      : Intel(R) Xeon(TM) CPU 2.80GHz
stepping        : 1
cpu MHz         : 2793.581
cache size      : 1024 KB
...

processor       : 1
vendor_id       : GenuineIntel
cpu family      : 15
model           : 4
model name      : Intel(R) Xeon(TM) CPU 2.80GHz
stepping        : 1
cpu MHz         : 2793.581
cache size      : 1024 KB
...
```

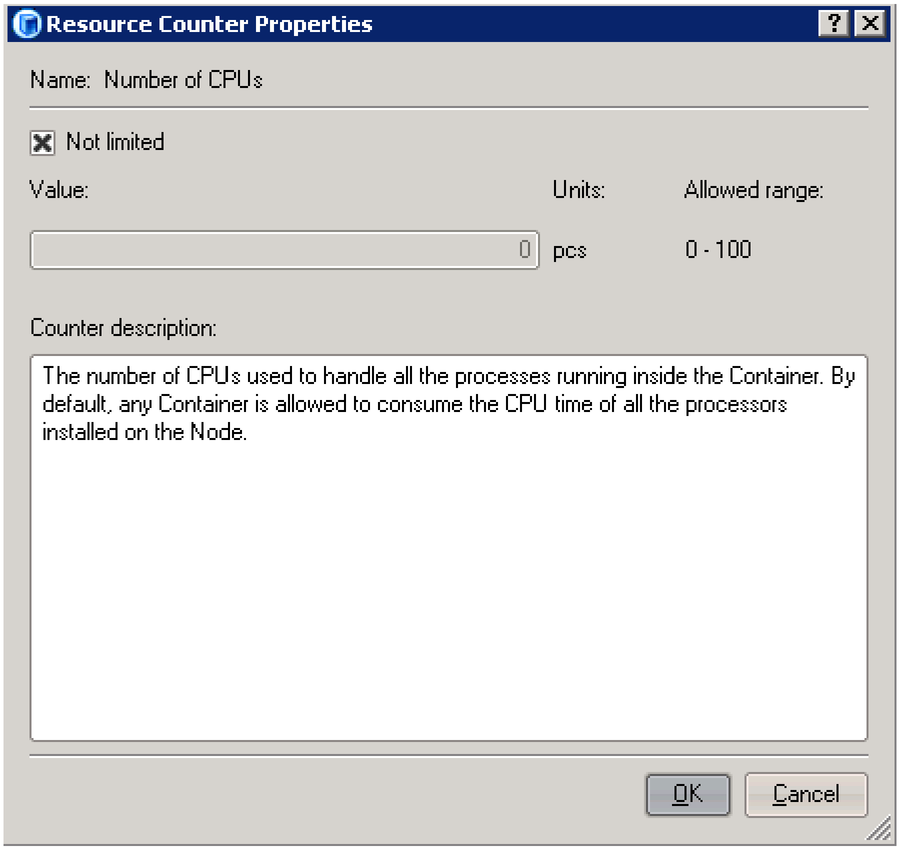
The output shows that Container 101 is currently bound to only two processors on the Node instead of 4 available for the other Containers on this Node. It means that, from now on, the processes of Container 101 will be simultaneously executed on no more than 2 physical CPU cores while the other Containers will continue consuming the CPU time of all 4 processors. Also notice that the physical CPU cores for Container 101 might not remain the same during the Container operation. They might change for load balancing reasons; the only thing that cannot be changed is their maximum number.

In Parallels Management Console, you can configure the number of CPUs to be available to a Container by doing the following:

1 Select the **Parallels Virtuozzo Containers** item under the corresponding Hardware Node name.

2 Right-click the Container for which you want to change the number of available CPUs, and choose **Properties**.

3 In the **Parameters** table on the **Resources** tab of the displayed window, double-click the `cpus` item.



Resource Counter Properties

Name: Number of CPUs

☒ Not limited

Value:	Units:	Allowed range:
0	pcs	0 - 100

Counter description:

The number of CPUs used to handle all the processes running inside the Container. By default, any Container is allowed to consume the CPU time of all the processors installed on the Node.

4 Clear the **Not limited** check box, and specify the desired number of CPUs in the **Value** field.

5 Click **OK** twice.

Configuring CPU Limits

The CPU limit parameter indicates the maximum CPU power a Container may get for its running processes. The Container is not allowed to exceed the specified limit even if the Node has enough free CPU power. By default, the CPU limit parameter is disabled for all newly created Containers. This means that any application in any Container can use all the free CPU power of the Node.

To set a CPU limit for a Container, you can use the `pctl set` command with the `--cpulimit` option. In the following example, Container 101 is set to receive no more than 25% of the CPU time of the Node even if the CPUs of the Node are not fully loaded:

```
# pctl set 101 --cpulimit 25 --save
Saved parameters for Container 101
```

In this example, you set the CPU limit for Container 101 to 25% of the whole CPU power of the Node. Assuming that if the Node total CPU power is 2000 megahertz (MHz), Container 101 can get up to 500 MHz.

Now, imagine the situation when you migrate Container 101 to another Node whose CPU power equals 4000 MHz. On this server, Container 101 will be able to get 25% of 4000 MHz—that is, 1000 MHz. To ensure that Container 101 always has the same CPU limit on all Nodes, irrespective of their total CPU power, you can set CPU limits for the Container in megahertz (MHz). For example, to make Container 101 consume no more than 500 MHz on any Node, run the following command:

```
# pctl set 101 --cpulimit 500m --save
Saved parameters for Container 101
```

For more information on setting CPU limits for Containers, see **CPU Limit Configuration Specifics** (p. 126).

CPU Limit Configuration Specifics for Containers

Internally, Parallels Virtuozzo Containers sets the CPU limit for Containers in percent. On multi-core systems, each physical CPU core is considered to have the CPU power of 100%. So if a server has 4 CPU cores, the total CPU power of the server equals 400%.

You can also set a CPU limit in megahertz (MHz). If you specify the limit in MHz, Parallels Virtuozzo Containers uses the following formula to convert the CPU power of the server from MHz into percent:

```
CPULIMIT_% = 100% * CPULIMIT_MHz / CPUFREQ
```

where

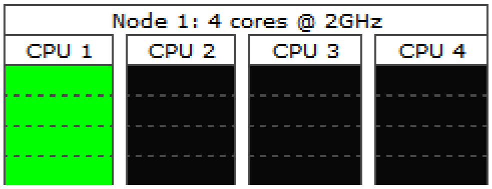
- `CPULIMIT_%` is the total CPU power of the server in percent.
- `CPULIMIT_MHz` is the total CPU power of the server in megahertz.
- `CPUFREQ` is the CPU frequency of one core on the server.

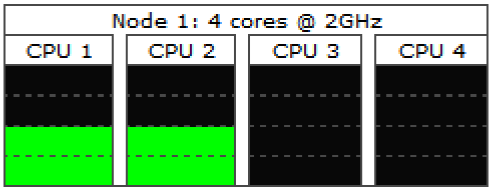
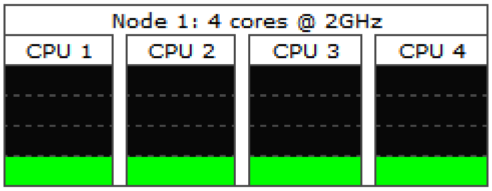
When setting a limit for a Container, note the following:

- Make sure that the CPU limit you plan to set for a Container does not exceed the total CPU power of the server.
- The processes running in a Container are scheduled for execution on all server CPUs in equal shares. For example, if a server has 4 CPUs, 1000 MHz each, and you set the CPU limit for a Container to 2000 MHz, the Container will consume 500 MHz from each CPU.
- All running Containers on a Node cannot simultaneously consume more CPU power than is physically available on the node. In other words, if the total CPU power of the Node is 4000 MHz, the Containers on the Node will not be able to consume more than 4000 MHz, irrespective of their CPU limits. It is, however, perfectly normal that the overall CPU limit of all Containers exceeds the Node total CPU power because most of the time Containers consume only part of the CPU power assigned to them.

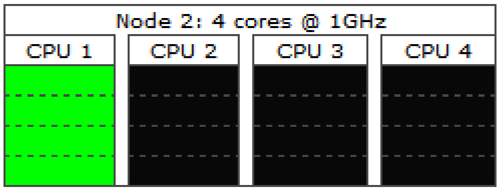
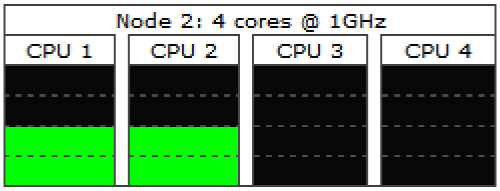
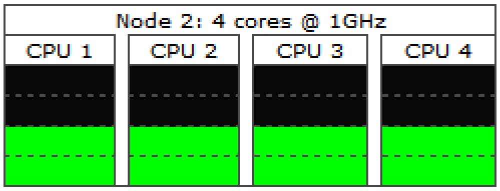
The tables below illustrate the distribution of CPU power on two nodes when you apply different CPU limits to one Container:

Node with 4 CPU cores, each core 2000 MHz

Number of virtual CPUs	Limit in %	Limit in MHz	Figure
1	100	2000	

2	100	2000	
4	100	2000	

Node with 4 CPU cores, each core 1000 MHz

Number of CPUs	Limit in %	Limit in MHz	Figure
1	100	1000	
2	100	1000	
4	200	2000	

For more examples, see the article on CPU limits in the Parallels Knowledgebase at http://kb.parallels.com/en/112588.

Controlling Container CPU Usage With VZASysD Plug-in

Parallels Virtuozzo Containers provides you with a special plug-in - `VZASysD` - allowing to automatically control the CPU consumption of any Container on the Hardware Node. This plug-in is automatically installed on your Node during the Parallels Virtuozzo Containers installation and gets started once the installation has successfully completed. When launched, the plug-in runs in the background of your system, collects the information on the Container CPU usage limits, compares the gathered information with the current CPU consumption by the corresponding Containers, and limits the Container CPU usage, if necessary.

> **Note:** `VZASysD` is an integral part of the Parallels Agent software and cannot be monitored or configured using the Parallels Virtuozzo Containers software or standard Linux tools.

By default, the `VZASysD` functionality is disabled for all Containers on the Hardware Node. To enable `VZASysD` to keep a check on the CPU consumption of a particular Container, you should open the `/etc/vz/conf/`*`CT_ID`*`.conf` file for editing (e.g. using `vi`) and set the following parameters in this file:

1. `BURST_CPU_AVG_USAGE`: the CPU usage limit, in percent, set for the Container. This limit is calculated as the ratio of the current Container CPU usage to the CPU limit (i.e to the value of the `CPULIMIT` parameter) set for the Container in its configuration file. If the limit is not specified, the full CPU power of the Hardware Node is considered as the CPU limit. Upon exceeding the `BURST_CPU_AVG_USAGE` limit, the `VZASysD` plug-in sets the Container CPU usage to the value defined in the `BURST_CPULIMIT` parameter for the given Container (see below).
2. `BURST_CPULIMIT`: the CPU power limit, in percent, the Container cannot exceed. The plug-in imposes the limitations from this parameter on a Container when this Container exceeds the limit set in the `BURST_CPU_AVG_USAGE` parameter.

> **Note:** You can also set the `BURST_CPU_AVG_USAGE` and `BURST_CPULIMIT` parameters in the global file (`/etc/vz/vz.conf`); in this case the specified limits will apply to all Containers on the Hardware Node (if not redefined in the corresponding Container configuration file).

After setting the aforementioned parameters in the Container configuration file, the `VZASysD` plug-in will carry out one of the following operations depending on the obtained results for the given Container:

- If the CPU usage consumption does not exceed the CPU limit set for the Container in the `BURST_CPU_AVG_USAGE` parameter, no actions are taken on the `VZASysD` part.
- If the processor time is currently overused by the Container, `VZASysD` places the restrictions set in the `BURST_CPULIMIT` parameter on the Container CPU usage. On the next check:
  - the set limit is removed if the CPU usage does not exceed the value calculated by the following formula: (`BURST_CPU_AVG_USAGE` x `BURST_CPULIMIT`) / 100% (the value of the `BURST_CPU_AVG_USAGE` parameter multiplied by the value of the `BURST_CPULIMIT` parameter and divided by 100%);

- the set limit is left intact if the Container CPU usage exceeds the aforementioned value.

For example, you can make the `VZASysD` plug-in control the CPU usage of Container 101 by editing the `BURST_CPU_AVG_USAGE` and `BURST_CPULIMIT` parameters in its configuration file as follows:

```
...
BURST_CPU_AVG_USAGE="80"
BURST_CPULIMIT="60"
...
```

From this moment on, `VZASysD` will regularly check Container 101 and compare its CPU usage with the value set in the `BURST_CPU_AVG_USAGE` parameter. If the CPU consumption by Container 101 exceeds the value set in `BURST_CPU_AVG_USAGE` (i.e. `80%`), the plug-in will keep the Container CPU usage under the limit specified in `BURST_CPULIMIT` (i.e. under `60%`). If during the next CPU usage check the CPU consumption by this Container:

- Becomes lower than the value calculated using the (`BURST_CPU_AVG_USAGE` x `BURST_CPULIMIT`) / 100% formula (i.e. (80% x 60%) / 100% = 48% of the CPU time), the `BURST_CPULIMIT` limit will be removed.
- Still exceeds 48% of the CPU time, the plug-in will continue keeping the Container CPU usage under the value specified in `BURST_CPULIMIT`.

In Parallels Management Console, you can perform the following operations to configure the `BURST_CPU_AVG_USAGE` and `BURST_CPULIMIT` parameters:

1. Click **Parallels Virtuozzo Containers** in the Management Console left pane, right-click the needed Container in the right pane, and choose **Properties**.
2. Click the **Resources** tab, and choose **CPU parameters**.
3. Double-click the corresponding parameter (either `burst_cpu_avg_usage` or `burst_cpulimit`) in the right part of the displayed window, and, if necessary, enter the right value.
4. Click **OK** twice.

By default, `VZASysD` checks the Container CPU usage every 5 minutes; however, you can configure the check interval by editing the `cpu_check_period` parameter in the Parallels Agent configuration file (`/var/vzagent/etc/vzagent.conf`). For example, you can do it as follows:

1. Right-click the Hardware Node name in the Parallels Management Console left pane, and choose **Tasks** > **Manage Parallels Agent Configuration**.
2. In the **Parallels Agent Configuration** window, expand the `vzasysd` key, and click the `configuration` subkey.
3. Double-click the `cpu_check_period` parameter in the right pane.
4. In the **Edit Parameter** window, enter the value you want in the **Parameter value** field.
5. Click the **OK** button and then the **Apply** button.

Configuring Containers to Run on Specific CPUs

By default, Containers can use the power of all CPUs installed on the physical server. For example, if the server has 8 CPUs installed, Containers can consume the CPU power of all 8 processors. In Parallels Virtuozzo Containers 4.7, however, you can configure Containers to run on specific CPUs only:

- You can set the CPU affinity mask for a Container. Using this mask, you can specify on which particular CPUs the Container may run.
- You can set the NUMA (Non-Uniform Memory Access) node mask. Using this mask, you can bind a Container to a specific NUMA node, allowing the processes running in the Container to be executed on all CPUs belonging to this NUMA node.

Both ways are described below in detail.

Setting the CPU Affinity Mask

In Parallels Virtuozzo Containers 4.7, you can configure Containers to use the CPU power of specific processors only. Let us assume the following:

- Your Hardware Node has 6 CPUs installed.
- You want the processes running in Container 101 to be executed on CPUs 2 and 3.

To configure Container 101 to run on CPUs 2 and 3 only, you can execute the following command:

```
# vzctl set 101 --cpumask 2,3 --save
Saved parameters for Container 101
```

To check that the affinity mask has been successfully set, use this command:

```
# vzlist 101 -o cpumask
         CPUMASK
             2-3
```

To make Container 101 use all CPUs on the Node again, run the following command:

```
# vzctl set 101 --cpumask all --save
Saved parameters for Container 101
```

Binding Containers to NUMA Nodes

On systems with a NUMA (Non-Uniform Memory Access) architecture, you can configure Containers to use CPUs from specific NUMA nodes only. Let us assume the following:

- Your physical server has 8 CPUs installed.
- The CPUs are divided into 2 NUMA nodes: NUMA node 0 and NUMA node 1. Each NUMA node has 4 CPUs.
- You want the processes in Container 101 to be executed on the processors from NUMA node 1.

To set Container 101 to use the processors from NUMA node 1, run the following command:

```
# vzclt set 101 --nodemask 1 --save
Saved parameters for Container 101
```

To check that Container 101 is now bound to NUMA node 1, use this command:

```
# vzlist 101 -o numamask
         NODEMASK
                1
```

To unbind Container 101 from NUMA node 1, execute this command:

```
# vzctl set 101 --nodemask all -save
Saved parameters for Container 101
```

Now Container 101 should be able to use all CPUs on the Node again.

Note: For more information on NUMA, visit http://lse.sourceforge.net/numa.

Managing Network Accounting and Bandwidth

This section explains how to perform the following tasks:

- setting up network classes
- viewing network traffic statistics
- turning on and off network bandwidth management
- setting up the bandwidth limit for a Container

Network Traffic Parameters

The table below summarizes the network traffic parameters that you can control. The **File** column indicates whether the parameter is defined in the global configuration file (G), in the Container configuration files (V), or it is defined in the global configuration file but can be overridden in a separate Container configuration file (GV).

Parameter	Description	File
`traffic_shaping`	If set to `yes`, traffic limitations for outgoing traffic are set for Containers. The default is `no`.	G
`bandwidth`	This parameter lists all the network adapters installed on the Node and their bandwidth.	G
`totalrate`	This parameter defines the bandwidth to be allocated for each and every network class. It is active if traffic shaping is turned on.	G
`rate`	If traffic shaping is turned on, this parameter specifies the bandwidth guarantee for any Container.	GV
`ratebound`	If this parameter is set to `yes`, the bandwidth guarantee (the global rate parameter) is also the limit for the Container, and the Container cannot borrow the bandwidth from the `TOTALRATE` bandwidth pool.	V

Configuring Network Classes

Parallels Virtuozzo Containers allows you to track the inbound and outbound network traffic as well as to shape the outgoing traffic for a Container. To provide the ability to distinguish between domestic and international traffic, a concept of network classes is introduced. It is important to fully understand this notion, because network classes IDs are used in the values of some network traffic parameters. A network class is a range of IP addresses for which Parallels Virtuozzo Containers counts and shapes the traffic.

Classes are specified in the `/etc/vz/conf/networks_classes` file. The file is in the ASCII format, and all empty lines and lines starting with the `#` sign are ignored. Other lines have the following format:

```
<class_id> <IP_address>/<prefix_length>
```

where *<class_id>* defines the network class ID, and the *<IP_address>/<prefix_length>* pair defines the range of IP addresses for this class. There may be several lines for each class.

Classes 0 and 1 have special meanings:

- Class 0 defines the IP address range for which no accounting is performed. Usually, it corresponds to the Node subnet (the Node itself and its Containers). Setting up class 0 is not required; however, its correct setup improves performance.
- Class 1 is defined by Parallels Virtuozzo Containers to match any IP address. It must be always present in the network classes definition file. Therefore, it is suggested not to change the default line in the `networks_classes` file.

```
1 0.0.0.0/0
```

If your Containers are using IPv6 addresses, you can also add the following line to this file:

```
1 ::/0
```

Other classes should be defined after class 1. They represent exceptions from the "matching-everything" rule of class 1. The example below illustrates a possible configuration of the network classes definition file containing rules for both IPv4 and IPv6 addresses:

```
# Node Containers networks
0 192.168.0.0/16
0 fe80::/64

# any IP address (all traffic)
1 0.0.0.0/0
1 ::/0

# class 2 - addresses for the "foreign" traffic
2 10.0.0.0/8
2 2001:db88::/64

# inside "foreign" network there
# is a hole belonging to "local" traffic
1 10.10.16.0/24
1 2001:db88:3333::/64
```

In this example, IPv4 addresses in the range of `192.168.0.0` to `192.168.255.255` and IPv6 addresses in the range of `fe80::` to `fe80::ffff:ffff:ffff:ffff` are treated as class 0 addresses and no accounting is done for the traffic from Containers destined to these addresses.

Class 2 matches the following IP addresses:

- IPv4 addresses from `10.0.0.0` to `10.255.255.255` with the exception of addresses in the sub-range of `10.10.16.0` to `10.10.16.255`, which are treated as class 1.
- IPv6 addresses from `2001:db88::` to `2001:db88::ffff:ffff:ffff:ffff` with the exception of addresses in the sub-range of `2001:db88:3333::` to `2001:db88:3333::ffff:ffff:ffff:ffff`, which are also treated as class 1.

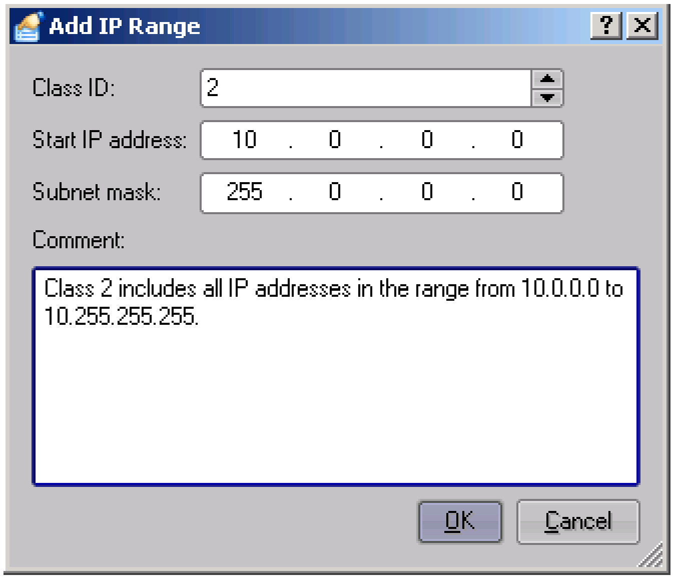
All other IP addresses (both IPv4 and IPv6) belong to class 1.

To set up network classes by means of Parallels Management Console:

1 Right-click the needed Node, and choose **Network Configuration** > **Configure Traffic Accounting and Shaping**.

2 On the **Accounting** tab of the displayed window, click the **New IP addresses range** button to display the **Add IP Range** window.

3 Fill in the fields provided (the **Class ID**, **Start IP address**, and **Subnet mask** fields are mandatory), and click **OK**. The example below illustrates how to create network class 2 matching all IP addresses in the range from `10.0.0.0` to `10.255.255.255`.



Class ID: 2

Start IP address: 10 . 0 . 0 . 0

Subnet mask: 255 . 0 . 0 . 0

Comment:

Class 2 includes all IP addresses in the range from 10.0.0.0 to 10.255.255.255.

After you click **OK** in the **Add IP Range** window, network class 2 will be created and displayed in the table on the **Traffic Accounting and Shaping** screen. To edit or delete the newly created class or any other existing classes, use the corresponding buttons on the **Accounting** tab in the **Traffic Accounting and Shaping** window.

> **Note:** After editing the `/etc/vz/conf/networks_classes` file manually (that is, without the help of Parallels Management Console), execute either the `/etc/init.d/vz accrestart` or `service vz accrestart` command for the changes made to the file to take effect.

Viewing Network Traffic Statistics

Parallels Virtuozzo Containers allows you to view the current network traffic statistics with the help of the `vznetstat` command. The session below shows the traffic statistics for Container 101:

```
# vznetstat -v 101
CTID  Net.Class  Input(bytes) Input(pkts)  Output(bytes) Output(pkts)
101       1           2202448       19527        9081832        19584
101       2                 0           0              0            0
```

In this case, around 2 MB of data were uploaded to the Container and about 9 MB were downloaded from it. All the traffic matches the definition of Class 1 and no data was exchanged with any hosts from Class 2 networks.

Without specifying a Container ID with the `–v` parameter, the command will display the statistics for all running Containers.

In Parallels Management Console, you can view the current network traffic statistics for a Container by doing the following:

1. Open the needed Container manager window by double-clicking the corresponding Container line in the right pane of the Parallels Management Console window.
2. Expand the **Monitor** item and select the **Network** folder. You can now see the network traffic statistics for the given Container in the right pane of the window.

Turning On and Off Network Bandwidth Management

Traffic shaping also known as network bandwidth management allows you to control what network bandwidth a Container receives for outgoing traffic. Traffic shaping is off by default in Parallels Virtuozzo Containers and is controlled by the `TRAFFIC_SHAPING` variable in the `/etc/vz/vz.conf` global configuration file.

Note: Container incoming traffic cannot be controlled in Parallels Virtuozzo Containers.

To turn traffic shaping on, you need to complete the following steps:

- Set the value of `TRAFFIC_SHAPING` to `yes` in the global configuration file.
- Correctly set up the `BANDWIDTH` and `TOTALRATE` parameters values.
- Start traffic shaping with the `/etc/init.d/vz shaperon` command.

The `BANDWIDTH` variable is used for specifying the network rate (in kilobits per second) of available network adapters. By default, it is set to `eth0:102400`, which corresponds to a 100Mb/s Fast Ethernet card. If your Node has more network adapters installed, you need to update this variable to list all the adapters participating in shaping. For example, in case of two Fast Ethernet cards this variable shall be set to `eth0:102400 eth1:102400`.

The `TOTALRATE` variable specifies the size of the so-called bandwidth pool for each network class being shaped. The bandwidth from the pool can be borrowed by Containers when they need more bandwidth for communicating with hosts from the corresponding network class. It is used to limit the total available outgoing traffic Containers can consume; the next section explains it in more detail. The format of this variable is `<NIC>:<network_class>:<bandwidth_in_Kbits_per_second>` and defines the pool size per network class for a given network adapter. Multiple entries for different network classes and adapters shall be separated by spaces. The default value for `TOTALRATE` is `eth0:1:4096`, which corresponds to the pool size of 4Mb/s for Network Class 1 on the first Ethernet adapter.

In the `/etc/vz/vz.conf` configuration file, you can also define the `RATE` variable whose value amounts to the number of kilobits per second any Container is guaranteed to receive for outgoing traffic with a network class on an Ethernet device. The default value of this parameter is `eth0:1:8`, which means that any Container is guaranteed to receive the bandwidth of at least 8 Kb/s for sending data to Class 1 hosts on the first Ethernet device. This bandwidth is not the limit for a Container (unless the `RATEBOUND` parameter is set to `yes` in the Container configuration file) – the Container is able to take the needed bandwidth from the `TOTALRATE` bandwidth pool if it is not used by other Containers.

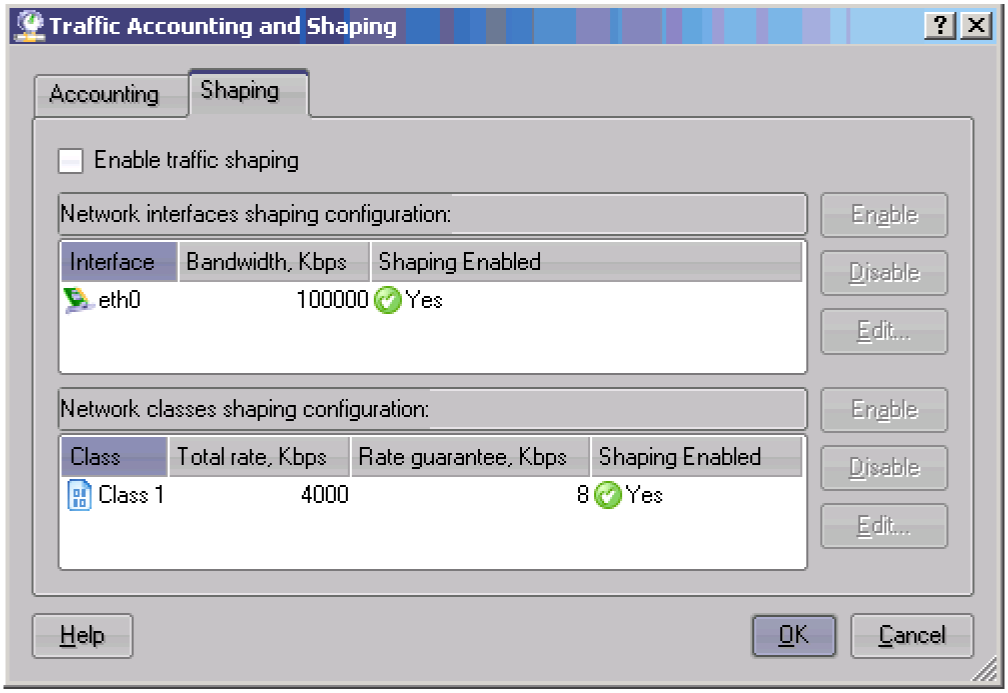
After setting up the above variables, start bandwidth management as follows:

```
# /etc/init.d/vz shaperon
Starting shaping: Ok
Set shaping on running Container :
vz WARNING: Can't get tc class for Container(101).
vz WARNING: Can't access file /var/run/vz_tc_classes. \
Creating new one.
vz WARNING: Can't get tc class for Container(1).
```

Now you have activated the network bandwidth limits. To turn traffic shaping off temporarily, use the `/etc/init.d/vz shaperoff` command. If you want to disable bandwidth management permanently, set the `TRAFFIC_SHAPING` variable to `no` in the `/etc/vz/vz.conf` configuration file.

Parallels Management Console provides a convenient means for turning on and off network bandwidth management on the **Shaping** tab of the **Traffic Accounting and Shaping** window. To open this window, do the following:

1. In the left pane of the Parallels Management Console window, right-click the needed Node, and choose **Network Configuration** > **Configure Traffic Accounting and Shaping**.
2. Go to the **Shaping** tab of the displayed window.



☐ Enable traffic shaping

Network interfaces shaping configuration:

Interface	Bandwidth, Kbps	Shaping Enabled
eth0	100000	✓ Yes

Network classes shaping configuration:

Class	Total rate, Kbps	Rate guarantee, Kbps	Shaping Enabled
Class 1	4000	8	✓ Yes

In this window, you can do the following:

- Enable/disable traffic shaping by selecting/deselecting the **Enable traffic shaping** check box.
- Add/edit/delete a network class for traffic shaping.
- Set up the `BANDWIDTH` parameter value for each Ethernet device.
- Set up the `TOTALRATE` parameter value for each network class.
- Set up the `RATE` parameter value which is the default network bandwidth guarantee for any Container sending data to the given network class.

The traffic shaping settings will take effect immediately on your clicking the **OK** button in this window.

Configuring Network Bandwidth Management for Containers

The network bandwidth for outgoing traffic a Container receives is controlled by two variables in the Container configuration file (`/etc/vz/conf/<CT_ID>.conf`): `RATE` and `RATEBOUND`.

> **Note:** Container incoming traffic cannot be controlled in the current version of Parallels Virtuozzo Containers.

The `RATE` variable has the same format as `TOTALRATE` - `<NIC>:<network_class>:<bandwidth>`. This variable specifies the guaranteed outgoing traffic rate that the corresponding Container receives. This rate can be specified differently for different network classes and network adapters; use space to separate several rate descriptions.

Bandwidth values are specified in Kb/s. It is recommended to increase this value in 8 Kb/s chunks and to set it no lower than 8 Kb/s.

The `RATEBOUND` variable specifies whether the network bandwidth available to the Container for outgoing traffic is limited by the bandwidth specified in the `RATE` variable. The possible values of the `RATEBOUND` variable are `yes` and `no`; the default is `no`. In this case the Container is allowed to take free bandwidth from the `TOTALRATE` pool.

The actual network bandwidth available to the Containers depends on the number of Containers and the total sum of the `RATE` values, and normally does not coincide with the bandwidth specified in their own `RATE` variables. If the `RATEBOUND` variable is set to `yes`, the Container bandwidth is limited by the value of the `RATE` variable.

If the Container configuration file does not specify any of these parameters, the values from the `/etc/vz/vz.conf` configuration file are taken. By default, Parallels Virtuozzo Containers does not set `RATEBOUND`, which corresponds to `no`, and `RATE` is set to `eth0:1:8`.

The network bandwidth management in Parallels Virtuozzo Containers works in the following way. The bandwidth pool for a given network class (configurable through the `TOTALRATE` variable in the global configuration file) is divided among the Containers transmitting data proportionally to their `RATE` settings. If the total value of the `RATE` variables of all Containers transmitting data does not exceed the `TOTALRATE` value, each Container gets the bandwidth equal or greater than its `RATE` value (unless this Container has the `RATEBOUND` variable set to `yes`). If the total value of the `RATE` variables of all Containers transmitting data exceeds the `TOTALRATE` value, each Container may get less than its `RATE` value.

The example below illustrates the scenario when there are two Containers, 101 and 102, which have `RATEBOUND` set to `no`, and Container 103 has `RATEBOUND` set to `yes`:

```
# grep ^RATE /etc/vz/conf/101.conf /etc/vz/conf/102.conf
RATE="eth0:1:8"
RATEBOUND=”no”
RATE="eth0:1:8"
RATEBOUND=”no”
# grep ^RATE /etc/vz/conf/103.conf
```

```
RATE="eth0:1:64"
RATEBOUND=”yes”
```

With the default `TOTALRATE` of 4096 Kb/s, bandwidth pool will be distributed according to the following table:

Container 101	Container 102	Container 103	Bandwidth consumed by Containers
transmits	idle	idle	Container101: 4096 Kb/s
idle	idle	transmits	Container103: 64 Kb/s
transmits	transmits	idle	Container101: 2048 Kb/s Container102: 2048 Kb/s
transmits	idle	transmits	Container101: 4032 Kb/s Container103: 64 Kb/s
transmits	transmits	transmits	Container101: 2016 Kb/s Container102: 2016 Kb/s Container103: 64 Kb/s

After you have set up Container bandwidth settings, activate your changes as shown below:

```
# /etc/init.d/vz shaperrestart
Stopping shaping: Ok
Starting shaping: Ok
Set shaping on running Container: Ok
```

This command clears off all existing shaping settings and sets them again using the configuration files of running Containers.

To configure the network bandwidth settings of a particular Container, do the following:

1. Click **Parallels Virtuozzo Containers** in the Parallels Management Console left pane, right-click the needed Container in the right pane, and choose **Properties**.
2. Go to the **Network** tab of the displayed window, and select the **Traffic Shaping** item.

In the displayed window, you can do the following:

- Add/edit/delete a network class for traffic shaping.
- Set up the `RATE` guarantee parameter value for the given Container for any network class.
- Set the value for the `RATEBOUND` parameter for the given Container by selecting/clearing the **Rate guarantee is also a bound** check box.
- Scale the traffic shaping configuration.

The traffic shaping settings will take effect immediately on your clicking the **OK** button in this window.

Managing Memory Parameters for Containers

This section describes the VSwap memory management system introduced in Parallels Virtuozzo Containers 4.7. You will learn to do the following:

- Configure the main VSwap parameters for Containers (p. 142).
- Set the memory allocation limit in Containers (p. 143).
- Enhance the VSwap functionality (p. 144).

If you have upgraded to Parallels Virtuozzo Containers 4.7, you will also learn how the system calculates the new VSwap values (p. 145) from the memory parameters that were applied to Containers before the upgrade.

Configuring Main VSwap Parameters

Parallels Virtuozzo Containers 4.7 introduces a new scheme for managing memory-related parameters in Containers—VSwap. Like many other memory management schemes used on standalone Linux computers, this scheme is based on two main parameters:

- *RAM*. This parameter determines the total size of RAM that can be used by the processes of a Container.
- *swap*. This parameter determines the total size of swap that can be used by a Container for swapping out memory once the RAM is exceeded.

Notes:

1. In Parallels Virtuozzo Containers 4.7, the new VSwap memory management scheme has replaced the SLM scheme.

2. You can also set memory limits for and provide memory guarantees to Containers by configuring multiple UBC (User Beancounter) parameters (numproc, numtcpsock, vmguarpages, and so on). These parameters provide you with comprehensive facilities of customizing the memory resources in respect of your Containers. However, this way of managing system resources is more complex and requires more effort to be made on your part to adopt it to your system. For detailed information on UBC parameters, refer to the *Administrator's Guide to Managing UBC Resources* available at http://www.parallels.com/products/pvcl/resources/docs.

The new memory management scheme works as follows:

1. You set for a Container a certain amount of RAM and swap space that can be used by the processes running in the Container.
2. When the Container exceeds the RAM limit set for it, the swapping process starts.

   The swapping process for Containers slightly differs from that on a standalone computer. The Container swap file is virtual and, if possible, resides in the Node RAM. In other words, when the swap-out for a Container starts and the Node has enough RAM to keep the swap file, the swap file is stored in the Node RAM rather than on the hard drive.
3. Once the Container exceeds its swap limit, the system invokes the OOM Killer for this Container.
4. The OOM Killer chooses one or more processes running in the affected Container and forcibly kills them.

By default, any newly created Container starts using the new memory management scheme. To find out the amount of RAM and swap space set for a Container, you can check the values of the PHYSPAGES and SWAPPAGES parameters in the Container configuration file, for example:

```
# grep PHYSPAGES /etc/vz/conf/101.conf
PHYSPAGES="65536:65536"
# grep SWAPPAGES /etc/vz/conf/101.conf
SWAPPAGES="65536"
```

In this example, the value of the PHYSPAGES parameter for Container 101 is set to 65536. The PHYSPAGES parameter displays the amount of RAM in 4-KB pages, so the total amount of RAM set for Container 101 equals to 256 MB. The value of the SWAPPAGES parameter is also set to 256 MB.

To configure the amounts of RAM and swap space for Container 101, use the --ram and --swap options of the vzctl set command. For example, you can execute the following command to set the amount of RAM and SWAP in Container 101 to 1 GB and 512 MB, respectively:

```
# vzctl set 101 --ram 1G --swap 512M
```

You can also use the --physpages and --swappages options to set the amount of RAM and swap space for Containers. For more information on all VSwap options, consult the *Parallels Virtuozzo Containers 4.7 Command Line Reference Guide*.

Configuring the Memory Allocation Limit

When an application starts in a Container, it allocates a certain amount of memory for its needs. Usually, the allocated memory is much more than the application actually requires for its execution. This may lead to a situation when you cannot run an application in the Container even if it has enough free memory. To deal with such situations, the VSwap memory management scheme introduces a new parameter—VM_Overcommit. Using this parameter, you can configure the amount of memory applications in a Container may allocate, irrespective of the amount of RAM and swap space assigned to the Container.

The amount of memory that can be allocated by applications of a Container is the sum of RAM and swap space set for this Container multiplied by a memory overcommit factor. In the default (basic) Container configuration file, this factor is set to 1.5. For example, if a Container is based on the default configuration file and assigned 1 GB of RAM and 512 MB of swap, the memory allocation limit for the Container will be 2304 MB. You can configure this limit and set it, for example, to 3 GB by running this command:

```
# vzctl set 101 --vm_overcommit 2
```

This command uses the factor of 2 to increase the memory allocation limit to 3 GB:

```
(1 GB of RAM + 512 MB of swap) * 2 = 3 GB
```

Now applications in Container 101 can allocate up to 3 GB of memory, if necessary.

Note: For more information on Container configuration files, see **Managing Container Resources Configurations** (p. 171).

Tuning VSwap

The new management scheme can be extended by using UBC parameters. For example, you can set the `numproc` parameter to configure the maximal number of processes and threads a Container may create or the `numfile` parameter to specify the number of files that may be opened by all processes in the Container. For detailed information on using UBC parameters, consult the *Administrator*'s *Guide to Managing UBC Resources*.

Configuring Legacy Containers

If you upgrade from an earlier version of Parallels Virtuozzo Containers 4.7, all Containers start using the new memory management scheme after the upgrade. Every time a Container is started, the system automatically calculates the values of RAM, swap, and allocation limit from the memory parameters that were applied to a Container before the upgrade and uses them for the Container while it is running. The calculation rules are described below.

SLM

If a Container uses only SLM parameters:

- The amount of RAM is set to the value of the `SLMMEMORYLIMIT` parameter.
- The amount of swap is set to 0.
- The memory allocation limit is set to the value of the `SLMMEMORYLIMIT` parameter multiplied by the value of the `VM_OVERCOMMIT` parameter. By default, the `VM_OVERCOMMIT` parameter is not limited, which means that the memory allocation limit is also unlimited. To configure the limit for a Container, you need to set the `VM_OVERCOMMIT` parameter in its configuration file.

For example, if the `SLMMEMORYLIMIT` and `VM_OVERCOMMIT` parameters for Container 101 are set to 1 GB and 1.5, respectively, the Container will have them set to the following values after the upgrade: RAM = 1 GB, swap = 0, memory allocation limit = 1.5 GB.

UBC

If a Container uses only UBC parameters:

- The amount of RAM is set to the soft limit of the `PRIVVMPAGES` parameter.
- The amount of swap is set to 0.
- The memory allocation limit is set to the hard limit of the `PRIVVMPAGES` parameter

For example, if the soft limit of `PRIVVMPAGES` for Container 101 is set to 65536 pages and the hard limit to 131072, then the Container will have the following parameters: RAM = 256 MB, swap = 0, memory allocation limit = 2.

SLM and UBC

If a Container uses both SLM and UBC parameters:

- The amount of RAM is set to the value of the `SLMMEMORYLIMIT` parameter.
- The amount of swap is set to 0.
- The memory allocation limit is set to the value of the `SLMMEMORYLIMIT` parameter multiplied by multiplied by the value of the `VM_OVERCOMMIT` parameter. By default, the `VM_OVERCOMMIT` parameter is not limited, which means that the memory allocation limit is also unlimited. To configure the limit for a Container, you need to set the `VM_OVERCOMMIT` parameter in its configuration file.

For example, if the SLMMEMORYLIMIT and VM_OVERCOMMIT parameters for Container 101 are set to 1 GB and 1.5, respectively, the Container will have them set to the following values after the upgrade: RAM = 1 GB, swap = 0, memory allocation limit = 1.5 GB.

Managing Disk Quotas

This section explains the basics of disk quotas, describes disk quota parameters as well as the following operations:

- enabling and disabling per-Container quotas
- setting Per-Container quotas
- enabling and disabling per-user and per-group quotas
- setting per-user and per-group quotas
- checking quota status

What are Disk Quotas?

In the current version of Parallels Virtuozzo Containers, system administrators can limit the amount of disk space Containers can use. Such quotas are known as per-Container or first-level quotas. In addition, administrators can limit disk space that individual users and groups in that Container can use. These quotas are called per-user and per-group quotas or second-level quotas.

By default, first-level quotas on your Node are enabled (which is defined in the /etc/vz/vz.conf configuration file), whereas second-level quotas must be turned on for each Container separately (in the corresponding Container configuration files). It is impossible to turn on second-level disk quotas for a Container if first-level disk quotas are off for that Container.

Parallels Virtuozzo Containers keeps quota usage statistics and limits in /var/vzquota/quota.<*CT_ID*>, a special quota file. The quota file has a special flag indicating whether the file is "dirty". The file becomes dirty when its contents become inconsistent with the real Container usage. This means that when the disk space or inodes usage changes during the Container operation, these statistics are not automatically synchronized with the quota file, the file just gets the "dirty" flag. They are synchronized only when the Container is stopped or when the Node is shut down. After synchronization, the "dirty" flag is removed. If the Node has been incorrectly brought down (for example, the power switch was hit), the file remains "dirty", and the quota is re-initialized on the next Container startup. This operation may noticeably increase the Node startup time. Thus, it is highly recommended to shut down the Node properly.

Disk Quota Parameters

The table below summarizes the disk quota parameters that you can control. The **File** column indicates that the parameter is defined in a Container configuration file (V) or in the global configuration file but can be overridden in a Container configuration file (GV).

Parameter	Description	File
`DISK_QUOTA`	Enables or disables per-Container quotas for all or particular Containers.	GV
`DISKSPACE`	The total disk space a Container may consume, in 1-KB blocks.	V
`QUOTAUGIDLIMIT`	Enables (if set to `1`) and disables (if set to `0`) per-user and per-group quotas.	V

Turning On and Off Per-Container Disk Quotas

The parameter that defines whether to use first-level disk quotas is `DISK_QUOTA` in the global configuration file (`/etc/vz/vz.conf`). By setting it to “no”, you will disable Parallels Virtuozzo Containers quotas completely.

This parameter can be specified in the Container configuration file (`/etc/vz/conf/<CT_ID>.conf`) as well. In this case its value will take precedence of the one specified in the global configuration file. If you intend to have a mixture of Containers with quotas turned on and off, it is recommended to set the `DISK_QUOTA` value to “yes” in the global configuration file and to “no” in the configuration file of that Container which does not need quotas.

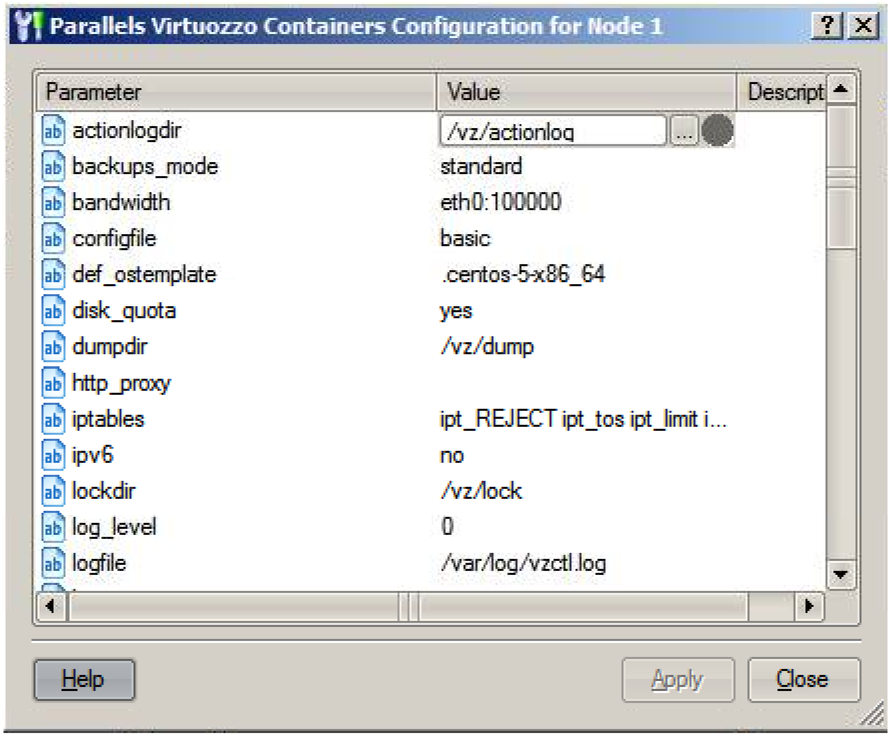
The session below illustrates a scenario when first-level quotas are on by default and are turned off for Container 101:

```
[checking that quota is on]
# grep DISK_QUOTA /etc/vz/vz.conf
DISK_QUOTA=yes

[checking available space on /vz partition]
# df /vz
Filesystem           1k-blocks      Used Available Use% Mounted on
/dev/sda2              8957295   1421982   7023242  17% /vz

[editing Container configuration file to add DISK_QUOTA=no]
# vi /etc/vz/conf/101.conf

[checking that quota is off for Container 101]
# grep DISK_QUOTA /etc/vz/conf/101.conf
DISK_QUOTA=no

# vzctl start 101
Starting Container ...
Container is mounted
Adding IP address(es): 10.0.16.101
Hostname for Container set: ct101
Container start in progress...
# vzctl exec 101 df
Filesystem           1k-blocks      Used Available Use% Mounted on
vzfs                   8282373    747060   7023242  10% /
```

As the above example shows, the only disk space limit a Container with the quotas turned off has is the available space and inodes on the partition where the Container private area resides.

To view or configure the `DISK_QUOTA` parameter status in the global file using Parallels Management Console, do the following:

1. In the Parallels Management Console left pane, right-click the needed Node, and choose **Tasks** > **Manage Parallels Virtuozzo Containers Configuration**.



Parallels Virtuozzo Containers Configuration for Node 1

Parameter	Value	Descript
actionlogdir	/vz/actionlog	
backups_mode	standard	
bandwidth	eth0:100000	
configfile	basic	
def_ostemplate	.centos-5-x86_64	
disk_quota	yes	
dumpdir	/vz/dump	
http_proxy		
iptables	ipt_REJECT ipt_tos ipt_limit i...	
ipv6	no	
lockdir	/vz/lock	
log_level	0	
logfile	/var/log/vzctl.log	

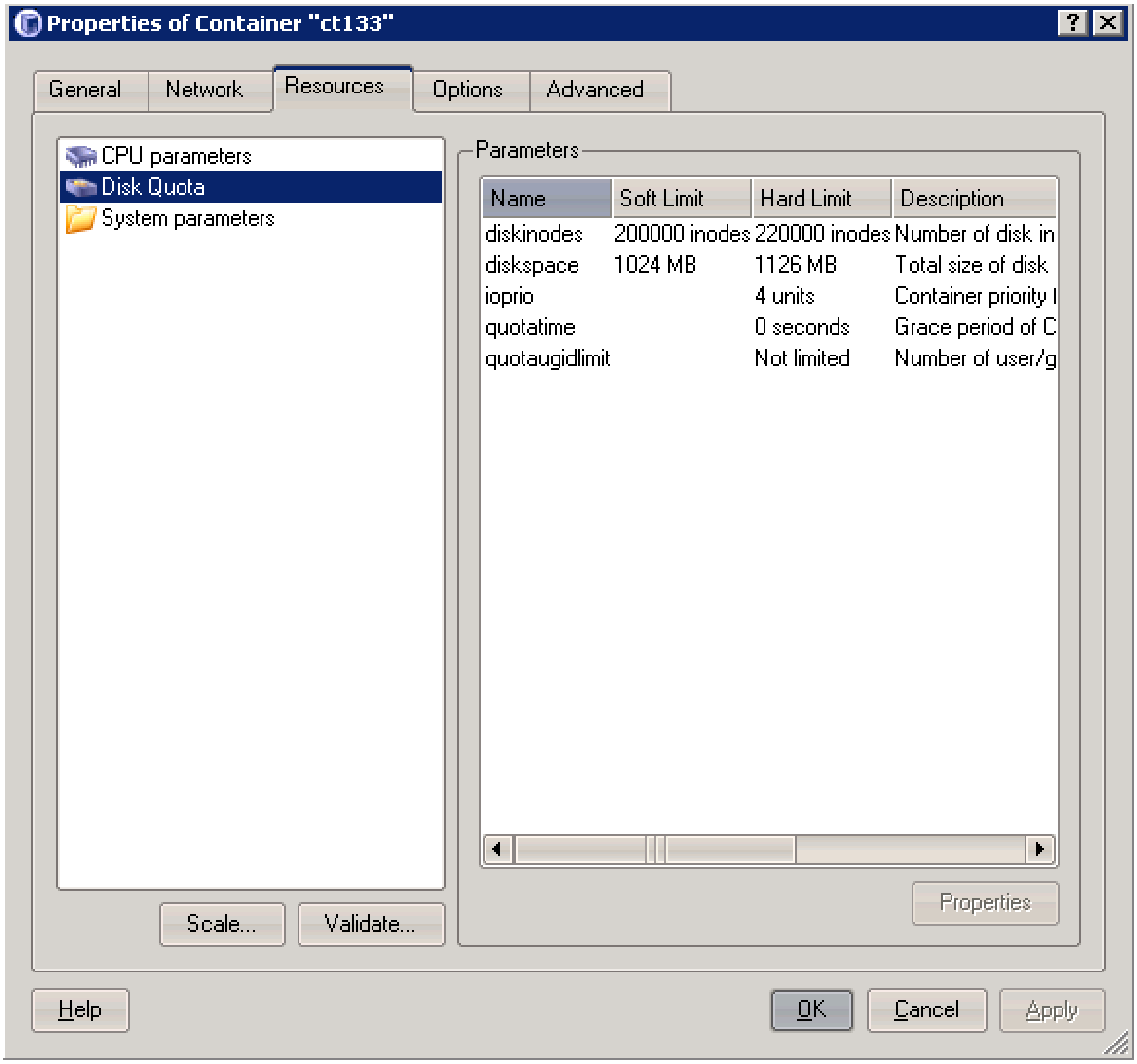
2 In the displayed window, you can view the current status of the `disk_quota` parameter, and modify it, if necessary.

3 Click the **Apply** button.

Parallels Management Console does not let you enable or disable disk quotas for individual Containers, thus overriding the global setting. If the first-level quotas are on by default, there is no way to rescind the calculation of quota data for a Container by means of Parallels Management Console. However, you can allow the Container to have an almost unlimited disk space and the number of inodes by doing the following:

1 Click **Parallels Virtuozzo Containers** in the Parallels Management Console left pane, right-click the needed Container in the right pane, and choose **Properties**.

2 Click the **Resources** tab and select the **Disk Quota** item.



Name	Soft Limit	Hard Limit	Description
diskinodes	200000 inodes	220000 inodes	Number of disk in
diskspace	1024 MB	1126 MB	Total size of disk
ioprio		4 units	Container priority I
quotatime		0 seconds	Grace period of C
quotaugidlimit		Not limited	Number of user/g

3 Double-click the `diskinodes` parameter, and select the **Not limited** check box to remove any limits on the number of disk inodes for the Container.

4 Click **OK** twice.

5 If necessary, repeat **Steps 3** and **4** for the `diskspace` parameter to allow the Container to have unlimited disk space.

Note: You must change the `DISK_QUOTA` parameter in the global configuration file only when all Containers are stopped, and in the Container configuration file—only when the corresponding Container is stopped. Otherwise, the configuration may prove inconsistent with the real quota usage, which may interfere with the normal Hardware Node operation.

Setting Up Per-Container Disk Quota Parameters

Three parameters determine how much disk space and inodes a Container can use. These parameters are specified in the Container configuration file:

`DISKSPACE` The total size of disk space that can be consumed by the Container, in 1-Kb blocks. When the space used by the Container hits the soft limit, the Container can allocate additional disk space up to the hard limit during the grace period specified by the `QUOTATIME` parameter.

`DISKINODES` The total number of disk inodes (files, directories, and symbolic links) the Container can allocate. When the number of inodes used by the Container hits the soft limit, the Container can create additional file entries up to the hard limit during the grace period specified by the `QUOTATIME` parameter.

`QUOTATIME` The grace period of the disk quota, in seconds. The Container is allowed to temporarily exceed the soft limit values for the disk space and disk inodes quotas for no more than the period specified by this parameter.

The first two parameters have both soft and hard limits (or, simply, barriers and limits). The hard limit is the limit that cannot be exceeded under any circumstances. The soft limit can be exceeded up to the hard limit, but as soon as the grace period expires, the additional disk space or inodes allocations will fail. Barriers and limits are separated by colons (":") in Container configuration files and in the command line.

The following session sets the disk space available to Container 101 to approximately 1 GB and allows the Container to allocate up to 90,000 inodes. The grace period for the quotas is set to 10 minutes:

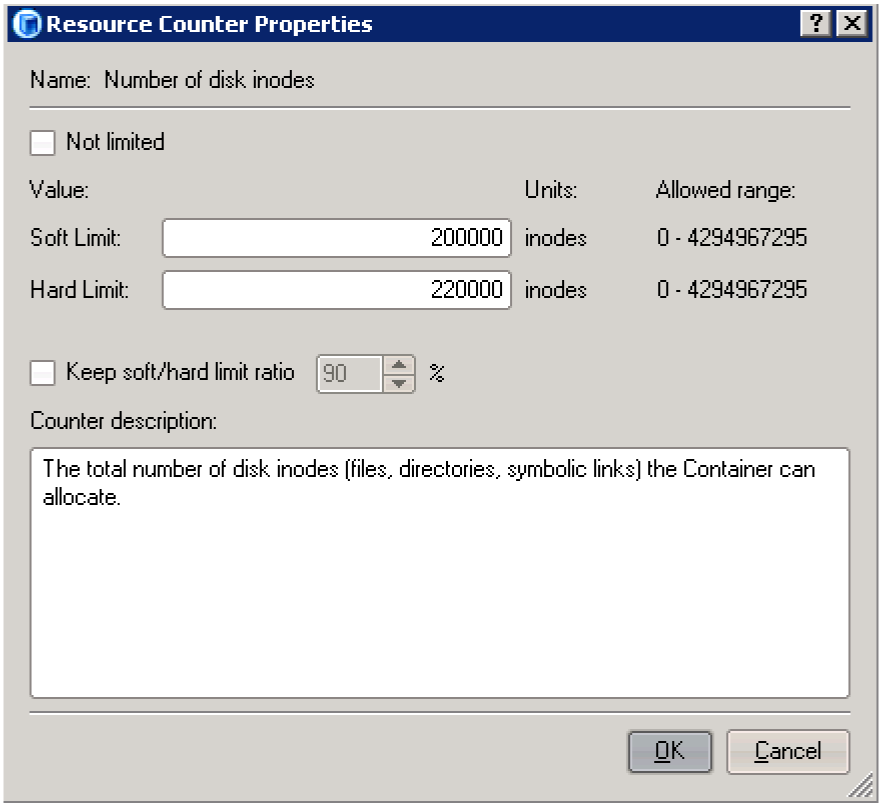
```
# vzctl set 101 --diskspace 1000000:1100000 --save
Saved parameters for Container 101
# vzctl set 101 --diskinodes 90000:91000 --save
Saved parameters for Container 101
# vzctl set 101 --quotatime 600 --save
Saved parameters for Container 101
# vzctl exec 101 df
Filesystem           1k-blocks      Used Available Use% Mounted on
vzfs                   1000000    747066    252934  75% /
# vzctl exec 101 stat -f /
  File: "/"
    ID: 0        0        Namelen: 255     Type: UNKNOWN (0x565a4653)
Blocks: Total: 1000000    Free: 252934     Available: 252934   Size: 1024
Inodes: Total: 90000      Free: 9594
```

It is possible to change the first-level disk quota parameters for a running Container. The changes will take effect immediately. If you do not want your changes to persist till the next Container startup, do not use the `--save` switch.

To set up per-Container disk quota parameters using Parallels Management Console, do the following:

1. Click **Parallels Virtuozzo Containers** in the Parallels Management Console left pane, right-click the needed Container in the right pane, and choose **Properties**.
2. Click the **Resources** tab, and select **Disk Quota**.

3 Double-click the `diskinodes` parameter in the right part of the displayed window, and enter the soft limit and hard limit values for this parameter in the fields provided. For example:



Name: Number of disk inodes

□ Not limited

Value:		Units:	Allowed range:
Soft Limit:	200000	inodes	0 - 4294967295
Hard Limit:	220000	inodes	0 - 4294967295

□ Keep soft/hard limit ratio 90 %

Counter description:

The total number of disk inodes (files, directories, symbolic links) the Container can allocate.

The hard limit is the limit that cannot be exceeded under any circumstances. The soft limit can be exceeded up to the hard limit, but as soon as the grace period expires, the additional disk space or inodes allocations will fail.

4 Click **OK**.

5 If necessary, repeat **Steps 3** and **4** for the `diskspace` and `quotatime` parameters to define the disk space quota and its grace period for the Container.

Turning On and Off Second-Level Quotas for Containers

The parameter that controls the second-level disk quotas is `QUOTAUGIDLIMIT` in the Container configuration file. By default, the value of this parameter is zero and this corresponds to disabled per-user and per-group quotas.

If you assign a non-zero value to the `QUOTAUGIDLIMIT` parameter, this action brings about the two following results:

1. Second-level (per-user and per-group) disk quotas are enabled for the given Container.
2. The value that you assign to this parameter will be the limit for the number of file owners and groups of this Container, including Linux system users. Notice that you will theoretically be able to create extra users of this Container, but if the number of file owners inside the Container has already reached the limit, these users will not be able to own files.

Enabling per-user and per-group quotas for a Container requires restarting the Container. The value for it should be carefully chosen; the bigger value you set, the bigger kernel memory overhead this Container creates. This value must be greater than or equal to the number of entries in the Container `/etc/passwd` and `/etc/group` files. Taking into account that a newly created Red Hat Linux-based Container has about 80 entries in total, the typical value would be 100. However, for Containers with a large number of users, this value should be increased.

When managing the `QUOTAUGIDLIMIT` parameter, keep in mind the following:

- If you delete a registered user but some files with their ID continue residing inside your Container, the current number of ugids (user and group identities) inside the Container will not decrease.
- If you copy an archive containing files with user and group IDs not registered inside your Container, the number of ugids inside the Container will increase by the number of these new IDs.

The session below turns on second-level quotas for Container 101:

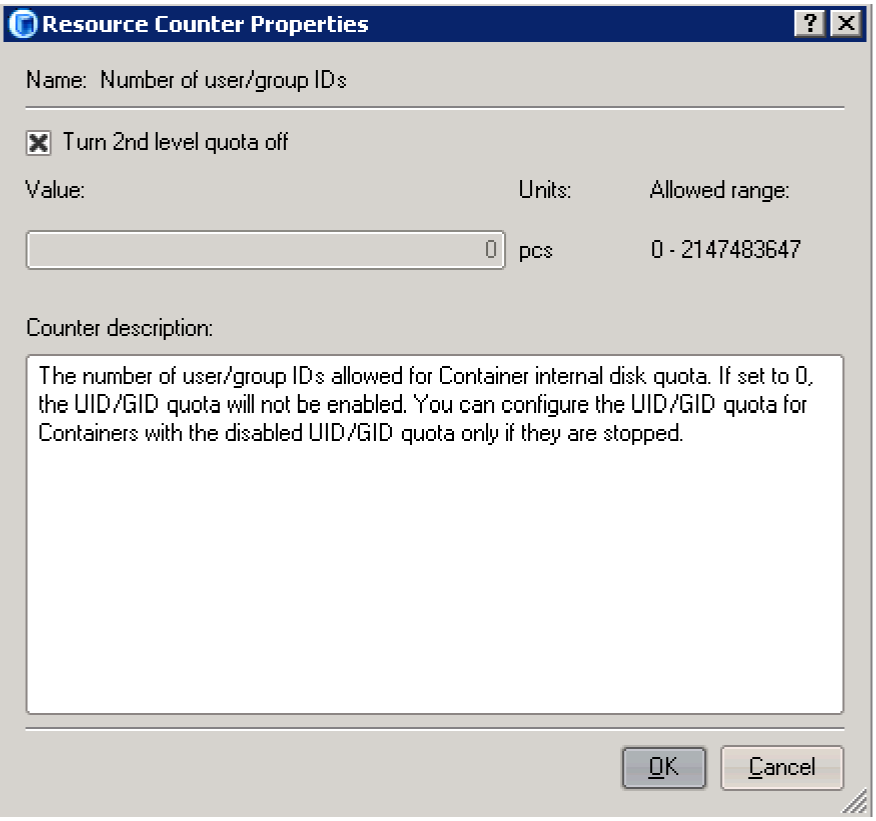
```
# vzctl set 101 --quotaugidlimit 100 --save
Unable to apply new quota values: ugid quota not initialized
Saved parameters for Container 101
# vzctl restart 101
Stopping Container ...
Container was stopped
Container is unmounted
Starting Container ...
Container is mounted
Adding IP address(es): 192.168.1.101
Hostname for Container set: ct101
Container start in progress...
```

In Parallels Management Console, you can manage second-level disk quotas by doing the following:

1. Click **Parallels Virtuozzo Containers** in the Parallels Management Console left pane, right-click the needed Container in the right pane, and choose **Properties**.

2 Click the **Resources** tab and the **Disk Quota** item.

3 Double-click the `quotaugidlimit` parameter.



Name: Number of user/group IDs

☒ Turn 2nd level quota off

Value:	Units:	Allowed range:
0	pcs	0 - 2147483647

Counter description:

The number of user/group IDs allowed for Container internal disk quota. If set to 0, the UID/GID quota will not be enabled. You can configure the UID/GID quota for Containers with the disabled UID/GID quota only if they are stopped.

4 Clear the **Turn 2nd level quota off** check box, enter the desired value in the **Value** field, and click **OK**.

5 Restart the Container, if it is running, for the changes to take effect.

Setting Up Second-Level Disk Quota Parameters

Parallels Virtuozzo Containers provides the standard Linux `quota` package for working inside Containers:

```
# vzctl exec 101 rpm -q quota
quota-3.03-1.1.parallels
```

This command shows that the `quota` package installed in the Container is built and shipped by Parallels. Use the utilities from this package (as is prescribed in your Linux manual) to set second-level quotas for the given Container. For example:

```
# ssh ct101
root@ct101's password:
Last login: Sat Jul 5 00:37:07 2009 from 10.100.40.18
[root@ct101 root]# edquota root
Disk quotas for user root (uid 0):
  Filesystem   blocks      soft      hard     inodes     soft    hard
  /dev/vzfs    38216       50000     60000    45454      70000   70000
[root@ct101 root]# repquota -a
*** Report for user quotas on device /dev/vzfs
Block grace time: 00:00; Inode grace time: 00:00
                        Block limits                File limits
User            used    soft    hard  grace    used  soft  hard  grace
----------------------------------------------------------------------
root      --   38218   50000   60000          45453 70000 70000
[the rest of repquota output is skipped]

[root@ct101 root]# dd if=/dev/zero of=test
dd: writing to `test': Disk quota exceeded
23473+0 records in
23472+0 records out
[root@ct101 root]# repquota -a
*** Report for user quotas on device /dev/vzfs
Block grace time: 00:00; Inode grace time: 00:00
                        Block limits                File limits
User            used    soft    hard  grace    used  soft  hard  grace
----------------------------------------------------------------------
root      +-   50001   50000   60000   none   45454 70000 70000
[the rest of repquota output is skipped]
```

The above example shows the session when the `root` user has the disk space quota set to the hard limit of 60,000 1KB blocks and to the soft limit of 50,000 1-KB blocks; both hard and soft limits for the number of inodes are set to 70,000.

It is also possible to set the grace period separately for block limits and inodes limits with the help of the `/usr/sbin/setquota` command. For more information on using the utilities from the `quota` package, consult the system administration guide shipped with your Linux distribution or online manual pages included in the package.

Parallels Management Console also provides means for setting up second-level disk quotas in Parallels Virtuozzo Containers. Do the following:

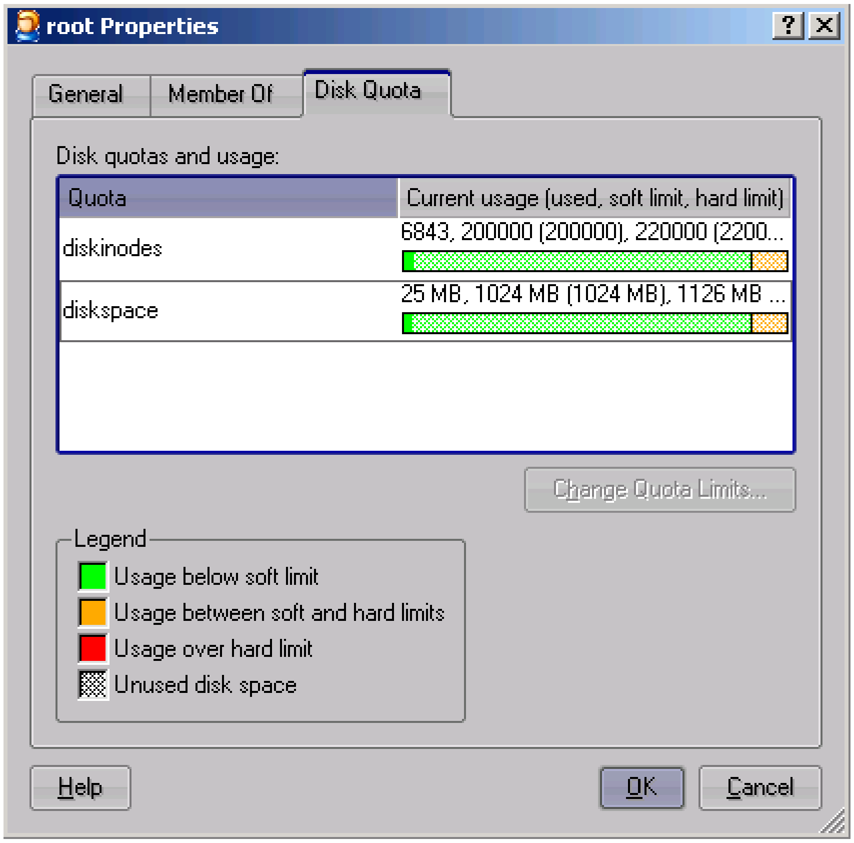
1 Open the needed Container manager window by double-clicking the corresponding Container line in the right pane of the Parallels Management Console window.

2 Select the **Users and Groups** item in the left pane of the Container manager window.

3 In the right pane, select either the **Groups** or **Users** tab to see the list of Container registered groups or users, respectively.

4 Double-click the name of the group/user for whom you want to set up the quota parameters. The group/user **Properties** window appears.

5 Click the **Disk Quota** tab in this window.



Disk quotas and usage:

Quota	Current usage (used, soft limit, hard limit)
diskinodes	6843, 200000 (200000), 220000 (2200...
diskspace	25 MB, 1024 MB (1024 MB), 1126 MB ...

Legend

- 🟩 Usage below soft limit
- 🟧 Usage between soft and hard limits
- 🟥 Usage over hard limit
- ▨ Unused disk space

6 Select the needed quota parameter (either `diskinodes` or `diskspace`), and click the **Change Quota Limits** button.

7 In the displayed window, enter the quota settings of your choice for the current group/user.

8 Click **OK** to close the **Second Level Disk Quota** window; then click **OK** to close the group/user **Properties** window.

Checking Quota Status

As the Node administrator, you can check the quota status for any Container with the `vzquota stat` and `vzquota show` commands. The first command reports the status from the kernel and shall be used for running Containers. The second command reports the status from the quota file (located at `/var/vzquota/quota.<CT_ID>`) and shall be used for stopped Containers. Both commands have the same output format.

The session below shows a partial output of Container 101 quota statistics:

```
# vzquota stat 101 -t
   resource          usage       softlimit      hardlimit    grace
  1k-blocks          38281         1000000        1100000
     inodes          45703           90000          91000
User/group quota: on,active
Ugids: loaded 34, total 34, limit 100
Ugid limit was exceeded: no

User/group grace times and quotafile flags:
 type block_exp_time inode_exp_time  dqi_flags
 user                                       0h
group                                       0h

User/group objects:
ID    type   resource    usage   softlimit   hardlimit    grace status
0     user  1k-blocks    38220       50000       60000          loaded
0     user     inodes    45453       70000       70000          loaded
[the rest is skipped]
```

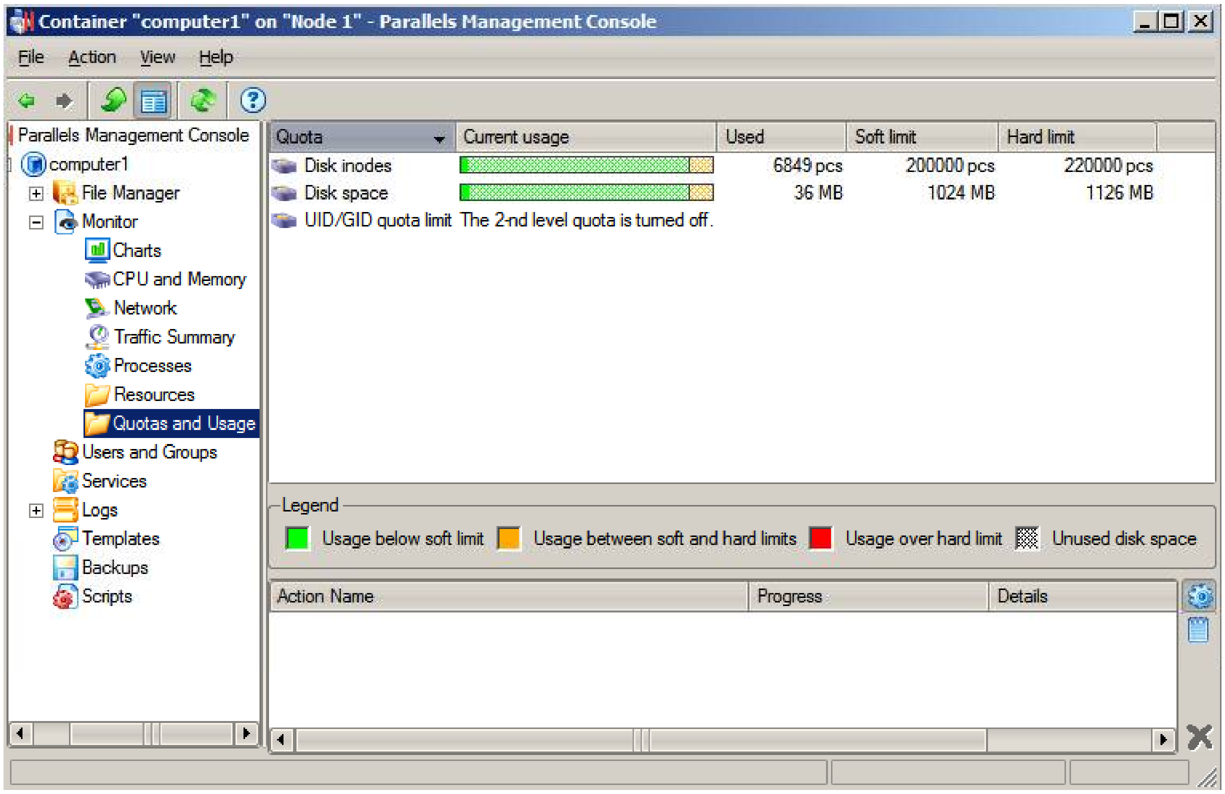
The first three lines of the output show the status of first-level disk quotas for the Container. The rest of the output displays statistics for user/group quotas and has separate lines for each user and group ID existing in the system.

If you do not need the second-level quota statistics, you can omit the `-t` switch from the `vzquota` command line.

To check the first-level quota status for a Container in Parallels Management Console:

1. Open the Container manager window by double-clicking the corresponding Container in the right pane of the Parallels Management Console window.
2. Expand the **Monitor** item, and select the **Quotas and Usage** folder.

You can see the first-level quota statistics for the current Container in the right pane of the window.



Quota	Current usage	Used	Soft limit	Hard limit
Disk inodes		6849 pcs	200000 pcs	220000 pcs
Disk space		36 MB	1024 MB	1126 MB
UID/GID quota limit	The 2-nd level quota is turned off.			

Legend

🟩 Usage below soft limit 🟧 Usage between soft and hard limits 🟥 Usage over hard limit ▨ Unused disk space

To check the second-level disk quota parameters for any group or user of the given Container, perform Steps 1 through 5 as is indicated in the previous section.

Cleaning Up Containers

The first-level quota assigned to this or that Container essentially shows how much space may be occupied by the Container *private* files, i.e. not by the OS or common applications files. The real OS and application files reside in the `/vz/template` directory on the Node and practically do not add up to the Container quota (except for the symlinks to them located inside the Container and occupying insignificant space).

However, there are situations when one and the same application or application update is installed not as a template, but separately inside each and every Container. A good example of this is the CPanel application with its robust auto-update features. If a certain version of CPanel is installed in a number of Containers, and then an update is released, CPanel automatically updates itself in all these Containers, thus creating a vast amount of identical files (not symlinks already) throughout the Containers. These files tell dramatically on the Container quotas, which may be avoided by putting all the identical files to the Node template area and creating symlinks instead of real files inside the affected Containers.

The problem like the one described above can be solved in two ways:

1. A special subarea is created inside the Node template area—`/vz/template/vc`—for housing the files identical among multiple Containers with the help of the `vzcache` utility.
2. If the application or application update installed directly into one or more Containers has a corresponding application template or template update installed on the Node, the real files inside the Containers are replaced with symlinks to the template files on the Node with the help of the `vzpkg link` utility. This utility is used to create symlinks to application EZ templates.

Moving Container Files to the Cache Area

We will illustrate the effect produced by `vzcache` by copying one and the same huge dummy file into two Containers. First, let us learn the disk space occupied by the whole `/vz` partition and by the two Containers—Container 101 and Container 102:

```
# df /vz
Filesystem           1K-blocks      Used Available Use% Mounted on
/dev/hda3             13756796   1348292  11622123  11% /vz
# vzctl exec 101 df
Filesystem           1K-blocks      Used Available Use% Mounted on
vzfs                   1048576     22311   1026265   3% /
# vzctl exec 102 df
Filesystem           1K-blocks      Used Available Use% Mounted on
vzfs                   1048576     22311   1026265   3% /
```

After that, we copy the dummy file, which is around 600 MB in size, to the root of these Containers:

```
# cp foo /vz/root/101
# cp foo /vz/root/102
```

Now check the disk space once again:

```
# df /vz
Filesystem           1K-blocks      Used Available Use% Mounted on
/dev/hda3             13756796   2569060  10401355  20% /vz
# vzctl exec 101 df
Filesystem           1K-blocks      Used Available Use% Mounted on
vzfs                   1048576    632430    416146  61% /
# vzctl exec 102 df
Filesystem           1K-blocks      Used Available Use% Mounted on
vzfs                   1048576    632430    416146  61% /
```

We see that around 600 MB has been added to the space occupied by each Container and, consequently, around 1.2 GB has been added to the space used on the `/vz` partition. Now it's time to resort to `vzcache` to get rid of identical files inside the Containers:

```
# vzcache -v 101 102
Processing VZFSv2 Container 101
VZFSv2 Container 101         78 regular files
Processing VZFSv2 Container 102
VZFSv2 Container 102         78 regular files
```

During the command execution, `vzcache` does the following:

- Looks for identical files inside Container 101 and Container 102.
- Creates the *`CT_UUID`* subdirectory (where *`CT_UUID`* denotes the Container unique identifier and can be determined by viewing the `UUID` parameters in the Container configuration file) within the Node template area (`/vz/template/vc` by default) for each Container.
- Moves the identical files to the created subdirectories in the Node template area.

Let us now take the final look at the disk space usage:

```
# df /vz
Filesystem           1K-blocks      Used Available Use% Mounted on
/dev/hda3             13756796   1953053  11017362  16% /vz
# vzctl exec 101 df
Filesystem           1K-blocks      Used Available Use% Mounted on
```

```
vzfs                   1048576     15105   1033471   2% /
# vzctl exec 102 df
Filesystem           1K-blocks      Used Available Use% Mounted on
vzfs                   1048576     15138   1033438   2% /
```

As you can see, both the Node and the Containers have each gained more than 600 MB of disk space. In real life, the disk space is gained by caching not one huge file in two Containers but a number of identical files across many Containers.

The operation of the `vzcache` utility may be customized to a certain extent by using `vzcache` command line switches (see the *Parallels Virtuozzo Containers 4.7 Reference Guide* for details).

Associating Container Files With Application Templates

It may often happen that a security update should immediately be applied to a package installed as a template on the Node and added to a number of Containers hosted there. However, it takes certain time to prepare a template update, so the Node and/or Container administrators are not inclined to wait for it and they install the original security update directly inside the Containers. As to the template update, it becomes available a few days afterwards. In other cases, a Container administrator might not know that there is a certain template installed on the Node, so they install the corresponding application directly inside their Container.

To eliminate cluttering up the Container disk space with application files that are present as part of an application template on the Node, the `vzpkg link` utility is used. This utility links a Container to the application EZ templates installed on the Node. For example, you can use the following command to replace the `openssl` files inside Container 101 running Fedora 13 with symlinks to these files in the `/vz/template/fedora-core/13/x86/config/app/openssl` directory on the Node:

```
# vzpkg link 101
```

Managing Disk I/O Parameters

This section explains how to manage disk input and output (I/O) parameters in Parallels Virtuozzo Containers systems.

Configuring Container Disk I/O Priority Levels

Parallels Virtuozzo Containers provides you with the capability of configuring the Container disk I/O (input/output) priority level. The higher the Container I/O priority level, the more time the Container will get for its disk I/O activities as compared to the other Containers on the Node. By default, any Container on the Node has the I/O priority level set to 4. However, you can change the current Container I/O priority level in the range from 0 to 7 using the `--ioprio` option of the `vzctl set` command. For example, you can issue the following command to set the I/O priority of Container 101 to 6:

```
# vzctl set 101 --ioprio 6 --save
Saved parameters for Container 101
```

To check the I/O priority level currently applied to Container 101, you can execute the following command:

```
# grep IOPRIO /etc/vz/conf/101.conf
IOPRIO="6"
```

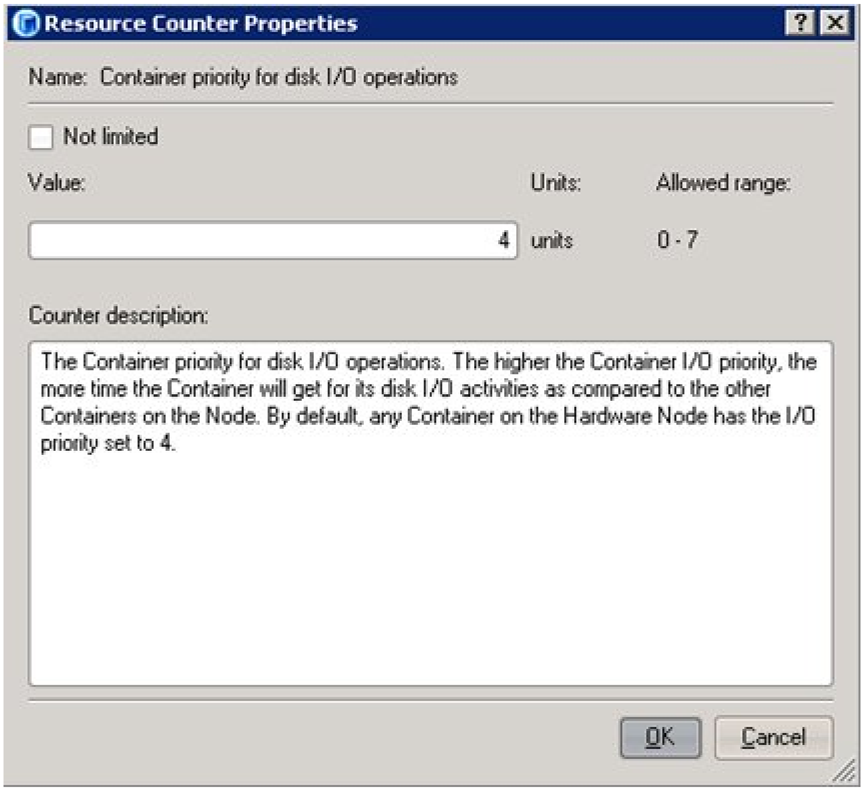
The command output shows that the current I/O priority level is set to 6.

To configure the I/O priority level of a particular Container in Parallels Management Console, do the following:

1 Click **Parallels Virtuozzo Containers** in the Parallels Management Console left pane, right-click the needed Container in the right pane, and choose **Properties**.

2 Click the **Resources** tab and then the **Disk Quota** item.

3 Double-click the `ioprio` parameter.



Name: Container priority for disk I/O operations

□ Not limited

Value:	Units:	Allowed range:
4	units	0 - 7

Counter description:

The Container priority for disk I/O operations. The higher the Container I/O priority, the more time the Container will get for its disk I/O activities as compared to the other Containers on the Node. By default, any Container on the Hardware Node has the I/O priority set to 4.

4 In the **Resource Counter Properties** window, you can view the disk I/O priority level currently set for the Container and change it, if necessary, by entering the desired value (from 0 to 7) in the field provided and clicking **OK**.

Configuring the Disk I/O Bandwidth for Containers

In Parallels Virtuozzo Containers 4.7, you can configure the bandwidth a Container is allowed to use for its disk input and output (I/O) operations. Limiting the disk I/O bandwidth can help you prevent the situations when high disk activities in one Container (generated, for example, by transferring huge amounts of data to/from the Container) can slow down the performance of other Containers on the Hardware Node.

By default, the I/O bandwidth limit for all newly created Containers is set to 0, which means that no limits are applied to the Containers. To limit the disk I/O bandwidth for a Container, you can use the `--iolimit` option of the `vzctl set` utility. For example, the following command sets the I/O bandwidth limit for Container 101 to 10 megabytes per second (MB/s):

```
# vzctl set 101 --iolimit 10 --save
Set up iolimit: 10485760
Saved parameters for Container 101
```

By default, the limit is set in megabytes per second. However, you can use the following suffixes to use other measurement units:

- `G`: sets the limit in gigabytes per second (1G).
- `K`: sets the limit in kilobytes per second (10K).
- `B`: sets the limit in bytes per second (10B).

Note: In the current version of Parallels Virtuozzo Containers, the maximum I/O bandwidth limit you can set for a Container is 2 GB per second.

To ensure that the I/O speed limit has been successfully applied to Container 101, use the `vzlist` utility:

```
# vzlist 101 -o iolimit
IOLIMIT
10485760
```

At any time, you can remove the I/O bandwidth limit set for Container 101 by running this command:

```
# vzctl set 101 --iolimit 0 --save
Set up iolimit: 0
Saved parameters for Container 101
```

Configuring the Number of I/O Operations Per Second

In Parallels Virtuozzo Containers 4.7, you can limit the maximum number of disk input and output operations per second a Container is allowed to perform (known as the IOPS limit). You may consider setting the IOPS limit for Containers with high disk activities to ensure that they do not affect the performance of other Containers on the Node.

By default, any newly created Container does not have the IOPS limit set and can perform so many disk I/O operations per second as necessary. To set the IOPS limit for a Container, you can use the `--iopslimit` option of the `vzctl set` command. For example, to allow Container 101 to perform no more than 100 disk I/O operations per second, you can run the following command:

```
# vzctl set 101 --iopslimit 100 --save
Set up iopslimit: 100
Saved parameters for Container 101
```

To ensure that the IOPS limit has been successfully applied to Container 101, use the `vzlist` utility:

```
# vzlist 101 -o iopslimit
IOPSLIMIT
      100
```

At any time, you can remove the IOPS limit set for Container 101 by running this command:

```
# vzctl set 101 --iopslimit 0 --save
Set up iopslimit: 0
Saved parameters for Container 101
```

Viewing Disk I/O Statistics for Containers

In Parallels Virtuozzo Containers 4.7, you can view disk input and output (I/O) statistics for Containers. To display the I/O statistics for all Containers on the Hardware (both running and stopped), you can run the `vzstat` utility with the `-a` option. For example:

```
# vzstat -a
 7:18pm, up 1 day,  1:29,  2 users, load average: 0.00, 0.01, 0.00
CTNum 2, procs  127: R   2, S  125, D   0, Z   0, T   0, X   0
CPU [ OK ]: CTs   0%, CT0   1%, user   0%, sys   2%, idle  98%, lat(ms)  12/0
Mem [ OK ]: total 1560MB, free 627MB/402MB (low/high), lat(ms) 0/0
  ZONE0 (DMA): size 16MB, act 0MB, inact 0MB, free 11MB (0/0/0)
  ZONE1 (Normal): size 880MB, act 76MB, inact 104MB, free 616MB (3/4/5)
  ZONE2 (HighMem): size 684MB, act 116MB, inact 153MB, free 402MB (0/1/1)
  Mem lat (ms): A0 0, K0 0, U0 0, K1 0, U1 0
  Slab pages: 65MB/65MB (ino 43MB, de 9MB, bh 2MB, pb 0MB)
Swap [ OK ]: tot 2502MB, free 2502MB, in 0.000MB/s, out 0.000MB/s
Net [ OK ]: tot: in  0.005MB/s   45pkt/s, out  0.000MB/s    1pkt/s
             lo: in  0.000MB/s    0pkt/s, out  0.000MB/s    0pkt/s
           eth0: in  0.005MB/s   45pkt/s, out  0.000MB/s    1pkt/s
           sit0: in  0.000MB/s    0pkt/s, out  0.000MB/s    0pkt/s
Disks [ OK ]: in 0.000MB/s, out 0.000MB/s

  CTID ST  IOUSED% IOWAIT% IOSPEED      IP
     0 OK     7.96   0.00  458/...KB/s 10.10.11.101
   101 OK     0.00   0.00    2/100MB/s  10.10.11.101
   111 OK     0.00   0.00  0.0/---KB/s  10.10.11.111
```

The information related to the Containers disk I/O statistics is at the end of the command output. The table below explains the displayed I/O parameters:

Parameter	**Description**
`IOUSED%`	The percentage of time the disks are used by the Container.
`IOWAIT%`	The percentage of time when at least one I/O transaction in the Container is waiting for being served.
`IOSPEED`	The current speed of disk I/O operations in the Container and the I/O limit set for this Container, if any. The value can be displayed in bytes, kilobytes, megabytes, or gigabytes per second, depending on the units you used to set the I/O limit.

Note: For more information on `vzstat` and its options, see the **Monitoring Resources in Text Console** section (p. 181) and the *Parallels Virtuozzo Containers 4.7 Reference Guide*.

Detecting Disk I/O Bottlenecks

Like in an ordinary data center, the disk I/O subsystem may also become a bottleneck in a Parallels Virtuozzo Containers system (that is, on the Hardware Node) and slow down the performance of other Containers or of the Node itself. Such I/O bottlenecks can often be caused by high disk activities in some of your Containers. For example, a high disk I/O load can be generated when you transfer huge amounts of data to/from a Container.

In Parallels Virtuozzo Containers 4.7, you can use the following procedure to find Containers that generate the highest disk I/O load:

1 Run the `cat /proc/bc/iostat` command on the Node. This command shows input/output statistics for all Containers on the Node, for example:

```
cat /proc/bc/iostat
sda 111 A 60 0 699   873   3584   10143  42776   0
sda 101 I 3  0 556   2819  5007   1135   40440   0
sda 0   I 0  0 30314 11473 879107 70227  2186282 0
hdc 0   I 0  0 0     0     0      0      0       0
```

2 In the command output, examine columns 4 (the number of queues with I/O requests) and 9 (the number of completed I/O requests). The Container that has the highest values in these columns generates the highest I/O load. In our example, this is Container 111.

   For the description of all columns in the `/proc/bc/iostat` output, see **iostat Output Parameters** below.

3 Enter the problem Container (for example, using the `vzctl enter` command or via SSH):

```
# vzctl enter 111
entered into Container 111
CT-101-bash-3.2#
```

4 Run the `ps axf` command inside the Container, and look for the processes that are in the `B` state for a long time, for example:

```
CT-111-bash-3.2# ps axf
  PID TTY      STAT   TIME COMMAND
    1 ?        Ss     0:00 init [3]
 7911 ?        S<s    0:00 /sbin/udevd -d
 9301 ?        B      6:30 syslogd -m 0
 9314 ?        Ss     0:00 /usr/sbin/sshd
 9323 ?        Ss     0:00 xinetd -stayalive -pidfile /var/run/xinetd.pid
 9340 ?        Ss     0:05 sendmail: accepting connections
 9348 ?        Ss     0:00 sendmail: Queue runner@01:00:00 for /var/spool/clientmqueue
 9358 ?        B      0:03 /usr/sbin/httpd
 9360 ?        S      0:00  \_ /usr/sbin/httpd
 9367 ?        Ss     0:00 crond
 9375 ?        Ss     0:00 /usr/sbin/saslauthd -m /var/run/saslauthd -a pam -n 2
 9376 ?        B      3:20  \_ /usr/sbin/saslauthd -m /var/run/saslauthd -a pam -n 2
 9382 ?        Ss+    0:00 /sbin/mingetty console
20396 ?        Ss     0:00 vzctl: pts/0
20397 pts/0    Ss     0:00  \_ -bash
20415 pts/0    R+     0:00      \_ ps axf
```

   In our example, two processes with PIDs 9301 and 9376 have spent the highest time in the `B` state.

5 Run the `cat /proc/PID/io` command, and check the `read_bytes` and `write_bytes` values in the command output:

```
CT-111-bash-3.2# cat /proc/9301/io
rchar: 3266
wchar: 854
syscr: 14
syscw: 8
read_bytes: 228672665
write_bytes: 426542387
cancelled_write_bytes: 0

CT-111-bash-3.2# cat /proc/9376/io
rchar: 3266
wchar: 854
syscr: 14
syscw: 8
read_bytes: 286672
write_bytes: 347606
cancelled_write_bytes: 0
```

As the commands output shows, process 9301 running in Container 111 can most probably cause performance problems on the Node.

iostat Output Parameters

The table below describes all columns in the `cat /proc/bc/iostat` command output:

Column Number	Description
1	Name of the disk device.
2	Container ID. The Hardware Node is marked with ID 0.
3	Device state. The following states are possible: • `I` (idle): there are no requests from the Container to the disk device. • `W` (wait): there are one or more queues with requests from the Container to the disk device, but no request is being processed at the moment. • `A` (active): a request from the Container to the device is being processed. • `D` (delayed): the processing of requests from the Container was delayed by the I/O scheduler.
4	Number of queues with I/O requests from the Container to the disk device.
5	Number of requests that have been delivered from the Container to the disk device.
6	Number of times when the disk device has been switched to the active state.
7	Total amount of time that the Container has spent in the waiting state, in milliseconds.
8	Total amount of time that the Container has spent in the active state, in milliseconds.
9	Number of completed I/O requests from the Container.
10	Number of transferred 512 bytes sectors (both read and write sectors).
11	Total amount of time that the Container has spent in the delayed state, in milliseconds.

Setting Disk I/O Limits for Backups and Migrations

The operations of backing up, restoring, and migrating Containers can generate a high disk I/O load on the Node, thus slowing down the performance of other Containers or of the Node itself. You can avoid such situations by setting disk I/O limits for these operations.

To set a disk I/O limit, do the following:

1 Open the Parallels Virtuozzo Containers global configuration file for editing, for example:

```
# vi /etc/vz/vz.conf
```

2 Locate the following section in the file:

```
# VZ Tools IO limit
# To enable - uncomment next line, check the value - there should not be CT with the
same ID
# VZ_TOOLS_BCID=2
# Uncomment next line to specify required disk IO bandwidth in Bps (10485760 - 10MBps)
# VZ_TOOLS_IOLIMIT=10485760
```

3 Edit this section as follows:

a Uncomment the `VZ_TOOLS_BCID` parameter to enable disk I/O limits for backup, restore, and migration operations. When defining the parameter, make sure that no Container with the specified ID exists on the Node.

b Uncomment the `VZ_TOOLS_IOLIMIT` parameter, and set the disk I/O limit for backup, restore, and migration operations. The value is set in bytes per second.

4 Save the file.

When setting disk I/O limits, pay attention to the following:

- `VZ_TOOLS_BCID` and `VZ_TOOLS_IOLIMIT` are global parameters—that is, once these parameters are set, they have effect on all Containers on the Node.
- The `VZ_TOOLS_BCID` and `VZ_TOOLS_IOLIMIT` parameters control the disk I/O load only for backup, restore, and migration operations.

Managing Container Resources Configurations

Any Container is configured by means of its own configuration file. You can manage Container configurations in a number of ways:

1. Using configuration sample files shipped with Parallels Virtuozzo Containers. These files are used when a new Container is being created; for details, see **Creating and Configuring New Containers** (p. 29). When you install Parallels Virtuozzo Containers on your Node, the default Container samples are put to the `/etc/vz/conf` directory. They have the following format: `ve-<`*`name`*`>.conf-sample` (for example, `ve-basic.conf-sample`).

   Currently, the following configuration sample files are provided:

   - `basic`. Use it for creating standard Containers.
   - `confixx`. Use it for creating Containers that are to run the Confixx control panel.
   - `vswap.plesk`. Use it for creating Containers with the Plesk control panel.
   - `vswap.256MB`. Use it for creating Containers with 256 MB of main memory.
   - `vswap.512Mb`. Use it for creating Containers with 512 MB of main memory.
   - `vswap.1024Mb`. Use it for creating Containers with 1024 MB of main memory.
   - `vswap.2048Mb`. Use it for creating Containers with 2048 MB of main memory.

   Notes:

   1. Configuration sample files cannot contain spaces in their names.

   2. There is a number of `slm*` configuration files in the `/etc/vz/conf` directory. These files are left for compatibility reasons only. In Parallels Virtuozzo Containers 4.7, the SLM memory management scheme was replaced by the new VSwap scheme. See **Managing VSwap Parameters** (p. 141) for details.

2. Using specific utilities for preparing configuration files in their entirety. The tasks these utilities perform are described in the following subsections of this section.
3. The direct creating and editing of the corresponding Container configuration file (`/etc/vz/conf/<`*`CT_ID`*`>.conf`). This can be performed either with the help of any text editor or through Parallels Virtual Automation or Parallels Management Console.

Any sample configuration file can also be applied to an existing Container. You would do this if, for example, you want to upgrade or downgrade the overall resources configuration of a particular Container:

```
# vzctl set 101 --applyconfig basic --save
```

This command applies all the parameters from the `ve-basic.conf-sample` file to Container 101.

Splitting the Hardware Node Into Equal Pieces

It is possible to create a Container configuration roughly representing a given fraction of the Node. If you want to create such a configuration that up to 20 fully loaded Containers would be able to be simultaneously running on the given Node, you can do it as follows:

```
# cd /etc/vz/conf
# vzsplit -n 20 -f mytest
Config /etc/vz/conf/ve-mytest.conf-sample was created
```

Notice that the configuration produced depends on the given Node resources. Therefore, it is important to validate the resulted configuration file before trying to use it, which is done with the help of the `vzcfgvalidate` utility. For example:

```
# vzcfgvalidate ve-mytest.conf-sample
Recommendation: kmemsize.lim-kmemsize.bar should be > 253952 \
(currently, 126391)
Recommendation: dgramrcvbuf.bar should be > 132096 (currently, 93622)
```

The number of Containers you can run on the Node is actually several times greater than the value specified in the command line because Containers normally do not consume all the resources that are guaranteed to them. To illustrate this idea, let us look at the Container created from the configuration produced above:

```
# vzctl create 101 --ostemplate redhat-el5-x86 --config mytest
Creating Container private area (redhat-el5-x86)
Container is mounted
Postcreate action done
Container is unmounted
Container private area created
Container registered successfully
# vzctl set 101 --ipadd 192.168.1.101 --save
Saved parameters for Container 101
# vzctl start 101
Starting Container ...
Container is mounted
...
# vzcalc 101
Resource     Current(%)  Promised(%)  Max(%)
Memory           0.53        1.90        6.44
```

As is seen, if Containers use all the resources guaranteed to them, then around 20 Containers can be simultaneously running. However, taking into account the **Promised** column output, it is safe to run 40-50 such Containers on this Node.

There is a possibility to create a suchlike configuration sample file using Parallels Management Console:

1 Right-click the **Container Samples** item in the Hardware Node main tree, and choose **"Slice" Hardware Node**.

2 Follow the instructions of the wizard.

Notes:

1. If you generate a Container configuration sample using the `vzsplit` command line utility, the resulting Container sample is put to the `/etc/vz/conf` directory. This sample can then be used by `vzctl create` when creating a new Container on its basis.

2. If you generate a Container sample by splitting Hardware Node resources via Parallels Virtuozzo Containers tools, the resulting Container sample can be used only by Parallels Virtual Automation and Parallels Management Console to create a new Container on its basis.

Scaling Container Configuration

Any configuration or configuration sample file can prove insufficient for your needs. You might have an application which does not fit into existing configurations. The easiest way of producing a Container configuration is to scale an existing one.

Scaling produces a “heavier” or “lighter” configuration in comparison with an existing one. All the parameters of the existing configuration are multiplied by a given number. A heavier configuration is produced with a factor greater than 1, and a lighter one – with a factor between 0 and 1.

> **Note:** If you create a new sample on the basis of an existing sample using the `vzcfgscale` command line utility, the resulting Container sample is put to the `/etc/vz/conf` directory. This sample can then be used by `vzctl create` when creating a new Container on its basis.

The session below shows how to produce a configuration sample 50% heavier than the `basic` configuration shipped with Parallels Virtuozzo Containers:

```
# cd /etc/vz/conf
# vzcfgscale -a 1.5 -o ve-improved.conf-sample ve-basic.conf-sample
# vzcfgvalidate ve-improved.conf-sample
Recommendation: kmemsize.lim-kmemsize.bar should be > 245760 \
(currently, 221184)
Recommendation: dgramrcvbuf.bar should be > 132096 (currently, 98304)
Validation completed: success
```

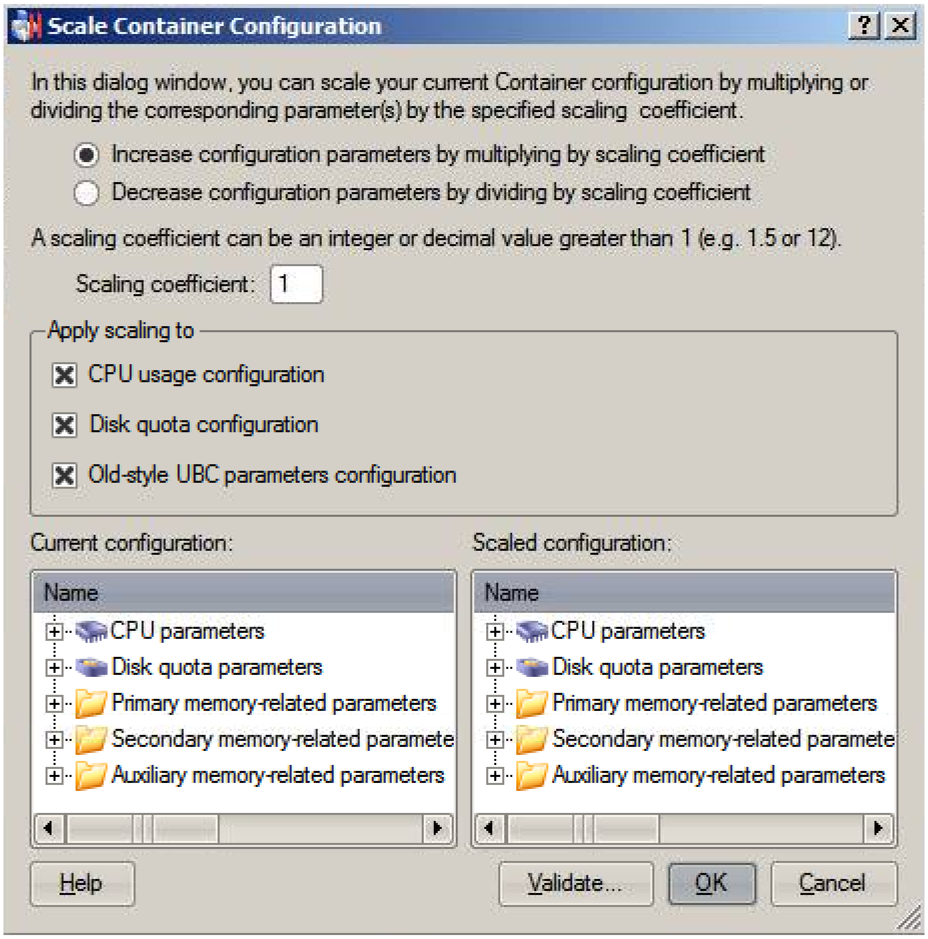
Now `improved` can be used in the `vzctl create` command for creating new Containers.

It is possible to use the same technique for scaling configurations of the existing Containers. Notice that the output file cannot be the same as the file being scaled. You have to save the scaling results into an intermediate file.

In Parallels Management Console, on the contrary, the scaling results are not written into a new file. If you scale the configuration of a Container, its configuration file is changed without saving the original file. If you scale a configuration sample file, it is correspondingly modified. That is why, it is recommended to create a copy of the configuration sample file you are going to scale before scaling it.

To scale an existing configuration using Parallels Management Console, do the following:

1. Select the **Container Configuration Samples** or **Parallels Virtuozzo Containers** item in the Hardware Node main tree.
2. Right-click the sample configuration file or the Container whose configuration you are going to scale, and choose **Properties**.
3. Go to the **Resources** tab, and click the **Scale** button.



In this dialog window, you can scale your current Container configuration by multiplying or dividing the corresponding parameter(s) by the specified scaling coefficient.

● Increase configuration parameters by multiplying by scaling coefficient

○ Decrease configuration parameters by dividing by scaling coefficient

A scaling coefficient can be an integer or decimal value greater than 1 (e.g. 1.5 or 12).

Scaling coefficient: 1

Apply scaling to

☒ CPU usage configuration

☒ Disk quota configuration

☒ Old-style UBC parameters configuration

Current configuration:
Name
CPU parameters
Disk quota parameters
Primary memory-related parameters
Secondary memory-related paramete
Auxiliary memory-related parameters
Scaled configuration:
Name
CPU parameters
Disk quota parameters
Primary memory-related parameters
Secondary memory-related paramete
Auxiliary memory-related parameters
Help
Validate...
OK
Cancel

4 Determine whether you want to enhance or attenuate the current configuration, and specify the factor.

5 Choose the groups of parameters to scale in the **Apply scaling to** section.

6 Validate the resulting configuration by clicking the **Validate** button.

7 Click **OK** to save the changes.

Validating Container Configuration

The system resource control parameters have complex interdependencies. The violation of these interdependencies can be catastrophic for the Container. In order to ensure that a Container does not break them, it is important to validate the Container configuration file before creating Containers on its basis.

The typical validation scenario is shown below:

```
# vzcfgvalidate  /etc/vz/conf/101.conf
Error: kmemsize.bar should be > 1835008 (currently, 25000)
Recommendation: dgramrcvbuf.bar should be > 132096 (currently, 65536)
Recommendation: othersockbuf.bar should be > 132096 \
(currently, 122880)
# vzctl set 101 --kmemsize 2211840:2359296 --save
Saved parameters for Container 101
# vzcfgvalidate  /etc/vz/conf/101.conf
Recommendation: kmemsize.lim-kmemsize.bar should be > 163840 \
(currently, 147456)
Recommendation: dgramrcvbuf.bar should be > 132096 (currently, 65536)
Recommendation: othersockbuf.bar should be > 132096 \
(currently, 122880)
Validation completed: success
```

The utility checks constraints on the resource management parameters and displays all the constraint violations found. There can be three levels of violation severity:

Recommendation This is a suggestion, which is not critical for Container or Node operations. The configuration is valid in general; however, if the system has enough memory, it is better to increase the settings as advised.

Warning A constraint is not satisfied, and the configuration is invalid. The Container applications may not have optimal performance or may fail in an ungraceful way.

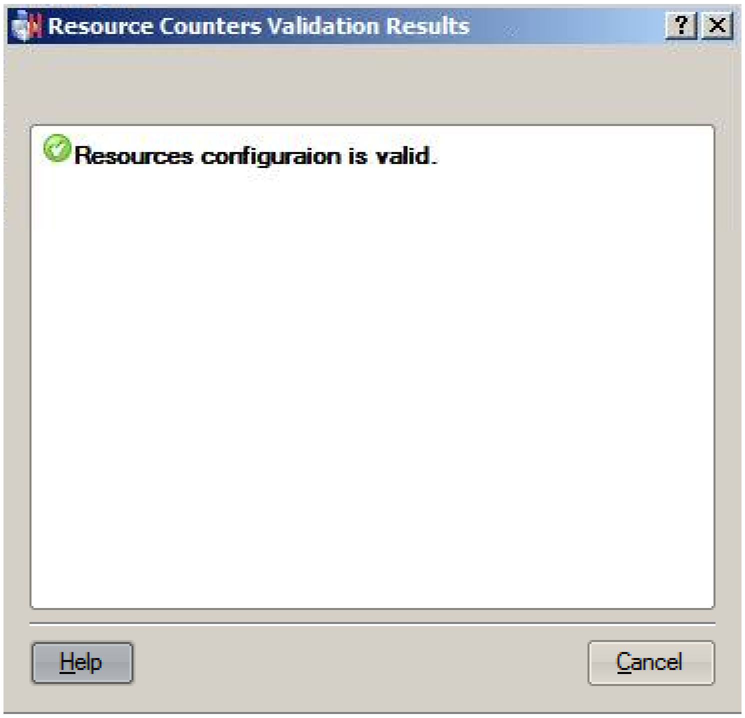
Error An important constraint is not satisfied, and the configuration is invalid. The Container applications have increased chances to fail unexpectedly, to be terminated, or to hang.

In the scenario above, the first run of the `vzcfgvalidate` utility found a critical error for the `kmemsize` parameter value. After setting reasonable values for `kmemsize`, the resulting configuration produced only recommendations, and the Container can be safely run with this configuration.

You can also validate any configuration sample file the given Hardware Node has by means of Parallels Management Console. For this, do the following:

1 Click the **Container Sample** item in the Hardware Node name, right-click the needed sample configuration file in the right pane, and select **Properties**.

2 Select the **Resources** tab and click the **Validate** button. A window appears informing you of the results. For example:



✓ Resources configuraion is valid.

In this example the configuration sample verification has passed successfully.

Applying New Configuration Samples to Containers

Parallels Virtuozzo Containers allows you to change the configuration sample file a Container is based on and, thus, to modify all the resources the Container may consume and/or allocate at once. For example, if Container 101 is currently based on the `basic` configuration sample and you are planning to run the Plesk application inside the Container, you may wish to apply the `ve-vswap.plesk` sample to it instead of `basic`, which will automatically adjust the necessary Container resource parameters for running the Plesk application inside Container 101. To do this, you can execute the following command on the Node:

```
# vzctl set 101 --applyconfig vswap.plesk --save
Saved parameters for Container 101
```

This command reads the resource parameters from the `ve-vswap.plesk.conf-sample` file located in the `/etc/vz/conf` directory and applies them one by one to Container 101.

When applying new configuration samples to Containers, keep in mind the following:

- All Container sample files are located in the `/etc/vz/conf` directory on the Node and are named according to the following pattern: `ve-<name>.conf-sample`. You should specify only the `<name>` part of the corresponding sample name after the `--applyconfig` option (`vswap.plesk` in the example above).
- The `--applyconfig` option applies all the parameters from the specified sample file to the given Container, except for the `OSTEMPLATE`, `TEMPLATES`, `VE_ROOT`, `VE_PRIVATE`, `HOSTNAME`, `IP_ADDRESS`, `TEMPLATE`, `NETIF` parameters (if they exist in the sample file).

You may need to restart your Container depending on the fact whether the changes for the selected parameters can be set on the fly or not. If some parameters could not be configured on the fly, you will be presented with the corresponding message informing you of this fact.

To apply a new Container configuration sample to a Container in Parallels Management Console, do the following:

1. Select the **Parallels Virtuozzo Containers** item in the Hardware Node main tree.
2. Right-click the corresponding Container, and choose **Tasks** > **Apply Container Sample**. The **Apply Container Configuration Sample** window appears.
3. In this window, select a new sample file to base the Container on and the parameters to be changed in accordance with this configuration sample. If you want to change all the parameters for the Container, select the check box near the **Applicable parameters** item or click the **Select All** button to the right of the table. Otherwise, expand the **Applicable parameters** item and select the check boxes near the parameters to be configured.
4. Click **OK**.

Once you select a new configuration sample and click **OK**, you may need to restart the Container depending on whether the changes for the selected parameters can be set on the fly or need a restart.

Note: Before applying a new Container sample to a Container, make sure you are aware of the resource values defined in this sample. Detailed information on Container samples is provided in the **Managing Container Resources Configurations** section (p. 171).

CHAPTER 5

Real-Time Monitoring in Parallels Virtuozzo Containers

This chapter describes the way to keep track of the resources consumption by running Containers and the Hardware Node itself.

In This Chapter

Monitoring Resources with Console

Parallels Virtuozzo Containers includes quite a number of means to monitor the Hardware Node and Containers resources. One of the most powerful features in Parallels Virtuozzo Containers is the ability to monitor resources in real time. To do this, you can run the `vzstat` utility on the Hardware Node, for example, with the following options:

```
# vzstat -d 5 -v
 12:34pm, up 14 days, 18:31,  1 user, load average: 1.00, 1.00, 1.00
CTNum 1, procs  245: R   3, S  228, D   0, Z   0, T  14, X   0
CPU [ OK ]: CTs   0%, CT0  50%, user  31%, sys  19%, idle  50%, lat(ms)  10/0
Mem [CRIT]: total 3940MB, free 962MB/0MB (low/high), lat(ms) 1/0
  ZONE0 (DMA): size 10MB, act 0MB, inact 0MB, free 2MB (0/0/0)
    fragm 5*1 7*2 5*4 4*8 5*16 5*32 4*64 3*128 1*256 1*512 1*1024
  ZONE1 (DMA32): size 2992MB, act 1631MB, inact 179MB, free 957MB (5/7/8)
    fragm 1*1 1*2 5*4 2*8 0*16 0*32 2*64 15*128 11*256 3*512 233*1024
  ZONE2 (Normal): size 1008MB, act 603MB, inact 258MB, free 2MB (1/2/2)
    fragm 1*1 9*2 3*4 3*8 2*16 1*32 2*64 1*128 1*256 2*512 1*1024
  Mem lat (ms): A0 0, K0 0, U0 0, K1 1, U1 0
  Slab pages: 243MB/243MB (ino 84MB, de 53MB, bh 49MB, pb 8MB)
Swap [ OK ]: tot 1992MB, free 1992MB, in 0.000MB/s, out 0.000MB/s
  Swap lat: si 0, 0/0 ms, so 0, 0/0 ms, 0/0 cpu ms
  Swap cache: add 0, del 0, find 0/0
Net [ OK ]: tot: in  0.002MB/s   22pkt/s, out  0.000MB/s    1pkt/s
             lo: in  0.000MB/s    0pkt/s, out  0.000MB/s    0pkt/s
           eth0: in  0.002MB/s   22pkt/s, out  0.000MB/s    1pkt/s
           eth1: in  0.000MB/s    0pkt/s, out  0.000MB/s    0pkt/s
           sit0: in  0.000MB/s    0pkt/s, out  0.000MB/s    0pkt/s
Disks [ OK ]: in 0.000MB/s, out 0.012MB/s
  root(/) free: 1964MB(50%), 972837ino(94%)
  vz(/vz) free: 174234MB(97%), 47117046ino(99%)
  sda1(/boot) free: 146MB(76%), 50155ino(99%)

  CTID ST    %VM     %KM        PROC    CPU     SOCK FCNT MLAT IP
     1 OK 3.0/-   0.2/-     0/78/256 0.0/100 42/1256    0    1 192.168.118.207
```

This screen will be updated with the time interval equal to the value specified after the `-d` (delay) option measured in seconds. In the session above, the statistics displayed will be renewed every five seconds. If the `-d` option is not specified, the default interval equals 1 second.

As you can see, the utility provides real-time information on the number of Containers and processes (in each and every state) on the Hardware Node, as well as on all the main resources subsystems pertaining both to the Hardware Node and to its Containers – the disk, network, CPU, and memory subsystems. You may want to shrink the output of the utility by specifying the `-b` (brief) option instead of the `-v` (verbose) one, or to do without any options to use the "normal" mode of displaying.

The following information is displayed per each Container:

Column Name	Description
`CTID`	Container ID.
`ST`	Container status. If there are no failed counters and the latency values are normal, the status is "OK". Otherwise, it is displayed in red as "!!". You can sort Containers by their

status to see the problem Containers first.

`%VM` Memory usage in a Container, in per cent to the total memory on the Node. The first number denotes how much memory the Container is currently using, and the second one is the total amount of RAM set for the Container.

`%KM` Kernel memory usage in a Container, in per cent to the normal zone size. The first number is how much `kmemsize` is being used, and the second one is the `kmemsize` barrier.

`PROC` Running/total/maximal processes number. The maximal number of processes represents the Container barrier. You can sort the Containers by the number of running or total processes.

`CPU` CPU usage in per cent to all available CPUs. The first number is how much of the CPU power is being used by the Container, and the second one is its guaranteed share judging by the `cpuunits` parameter. Note that the actual CPU usage may be higher than the guaranteed one.

`SOCK` Sockets usage, corresponding to the sum of the `numtcpsock` and `numothersock` parameters set in the Container configuration file. The first number is how many sockets are opened, the second one is the sockets barrier.

`FCNT` The number of Container failed counters for all the resource parameters. In the standard mode of displaying, this number represents the increase of failed counters since the previous screen update, whereas in the average mode of displaying, it represents an absolute failed counters sum for the given Container.

`MLAT` Maximal scheduling latency for the Container, in ms. This parameter shows the maximal scheduling latency inside the given Container, i.e. for how long (at the utmost) a process inside the Container awaits for the CPU.

`IP/HOSTNAME` The IP address or the hostname of the given Container. You may switch between them by pressing the `e` key on the keyboard while `vzstat` is running.

The **%VM**, **%KM**, **CPU**, and **SOCK** columns provide two values per column separated by a slash for each Container. The first value indicates the real usage of the corresponding parameter by the Container, and the second one – the maximal value allowed for the Container. The **PROC** column shows the number of processes in the corresponding Container in the following format: running/total/maximal number of processes.

The great thing about the `vzstat` utility is its interactivity. You can set the time interval, manage the mode of displaying, sort the Containers by a number of parameters, and all this on-the-fly. For example:

- While `vzstat` is running, press `t` on the keyboard, enter the new timeout (say, `180`), and press ENTER.
- Press `b` to switch to the brief details level.
- Press `w` to toggle the display of the swap information on the screen.
- Press `o`, and then `r` to sort the displayed Containers by the number of running processes.

Now your screen must look something like the following:

```
  1:20pm, up 14 days, 19:17,  1 user, load average: 1.00, 1.00, 1.00
CTNum 1, procs  249: R   2, S  229, D   0, Z   0, T  18, X   0
CPU [ OK ]: CTs   0%, CT0  50%, user  30%, sys  20%, idle  50%, lat(ms)   3/0
Mem [CRIT]: total 3940MB, free 958MB/0MB (low/high), lat(ms) 1/0
```

```
Net [ OK ]: tot: in  0.001MB/s   16pkt/s, out  0.000MB/s    1pkt/s
Disks [ OK ]: in 0.000MB/s, out 0.000MB/s

  CTID ST    %VM     %KM        PROC    CPU     SOCK FCNT MLAT IP
     1 OK 3.0/-   0.2/-     0/78/256 0.0/100 42/1256    0     1 192.168.118.207
```

The `vzstat` utility has a configuration file where you can set the values of different parameters indicating the warning and/or the error levels for them. If a parameter hits the warning level, it will be displayed in yellow by the utility, if it hits the error level – in red. Moreover, if a parameter has hit the error level, the **CRIT** warning is displayed instead of **OK** after the name of the corresponding subsystem (CPU, Memory, Swap, Net, or Disks). Thus, for example, if you see `Swap [ CRIT ]` on the screen, it means that one or more of the Hardware Node swap-related parameters (the total size of swap memory used, the swap in/out activity, etc.) has hit the error level. The offending parameter(s) will be displayed in red.

Consult the *Parallels Virtuozzo Containers 4.7 Reference Guide* for a complete list of command line options, interactive keys, and configuration file parameters of the `vzstat` utility.

Monitoring Resources with Parallels Management Console

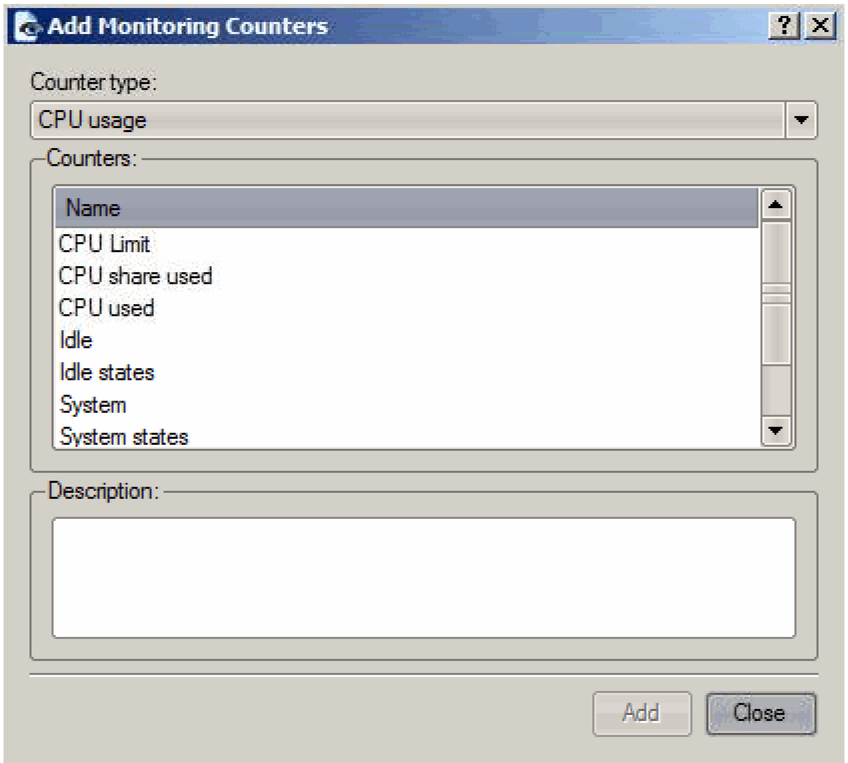
You can exploit the **Monitor** feature of Parallels Management Console for monitoring resources. This feature provides either the whole Hardware Node resources monitoring or the monitoring of resources consumption by a single Container, depending on whether you use the Management Console main window or a particular Container manager window. To open the latter, it is enough to double-click the necessary Container in the Container table in the right pane of the Management Console main window. The principles of working with these two kinds of monitors are essentially the same (only the set of the parameters that can be displayed is slightly different); therefore, they can be described together. You can access the Management Console **Monitor** feature by selecting the **Monitor** item in the left pane of the window you are working with.

Using Charts Representation

The charts section of Parallels Management Console lets you display a number of charts for monitoring various kinds of resources on a single grid. It offers means for better visualization of charts, like assigning colors and line styles to all the elements of the grid and charts or choosing a peculiar representation scale for each chart. You can save and load a set of counters you would usually monitor, thus avoiding the necessity of adding the counters one by one each time you start Management Console. You also have the possibility to replay the charts for any specified period of time by using logs.

The sequence of your actions can be the following:

1. To display the chart, expand the **Monitor** item in the window you are working with (either the Parallels Management Console main window or a Container manager window), and click **Charts** to see the monitor grid in the right pane.
2. Click the **Add Counters** button on the **Charts** toolbar.
3. In the **Add Monitoring Counters** dialog window, select the set of counters from which you want to add one(s) by selecting the desired group on the **Counter type** drop-down menu.
4. Select the needed counters, and click **Add**. You can use the Ctrl and Shift keys to add a number of counters from a group. When you select a certain counter with your mouse, the counter description is provided in the lower part of the **Add Monitoring Counters** dialog window. For example:



Counter type: CPU usage

5 Click **Close** after you have added all the desired counters.

Now that you have a number of counters on the grid, you can see a red line indicating the current moment of time moving from left to right as time passes and new values of monitored parameters appear on the grid. Now it's time to customize your view and learn the other opportunities. You may want to perform the following tasks:

- Adjust the periodicity of refreshing the information on the grid.
- Adjust the representation scale for each counter.
- Adjust colors and line styles for the visual elements.
- Highlight a certain counter.
- Save the current configuration of counters to be able to open it at any moment of time.
- Use the grid to replay some past real-time information about a set of parameters.

Adjusting Periodicity of Refreshing Information

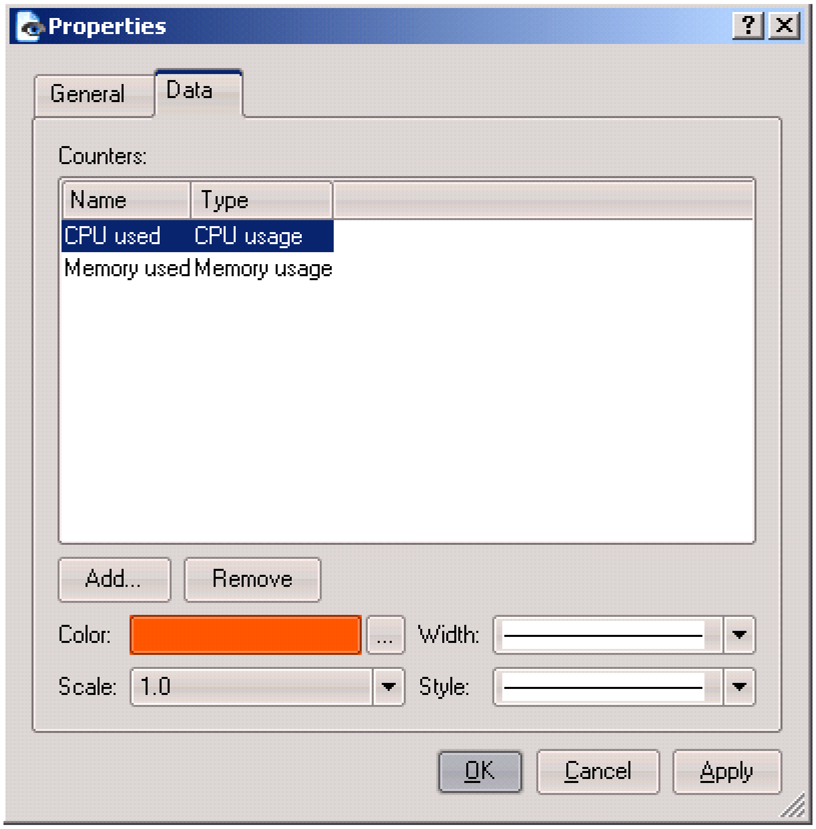
To set the time interval at which the information is refreshed for all the charts, right-click the **Charts** item in the Hardware Node or Container main tree and choose one of the following options:

- **Update Speed** > **High**. Choose this option to set the time interval to 1 second.
- **Update Speed** > **Normal**. Choose this option to set the time interval to 5 seconds.
- **Update Speed** > **Low**. Choose this option to set the time interval to 15 seconds.
- **Update Speed** > **Paused**. Choose this option to stop refreshing the information for the charts.

Adjusting Representation Scale

The value of any counter on the grid may vary from 0 to 100. These numbers are marked on the left of the grid. But the "weight" of these numbers is different for each counter. It is difficult to use one and the same scale, for example, for memory usage which can amount to hundreds of thousands of KBs and for CPU usage in percent. You can adjust the scale for each parameter separately for their better visualization on the grid:

1 Right-click the name of the corresponding counter in the table of displayed counters below the grid, and choose **Properties**.



Counters:

Name	Type
CPU used	CPU usage
Memory used	Memory usage

Color: 🟧

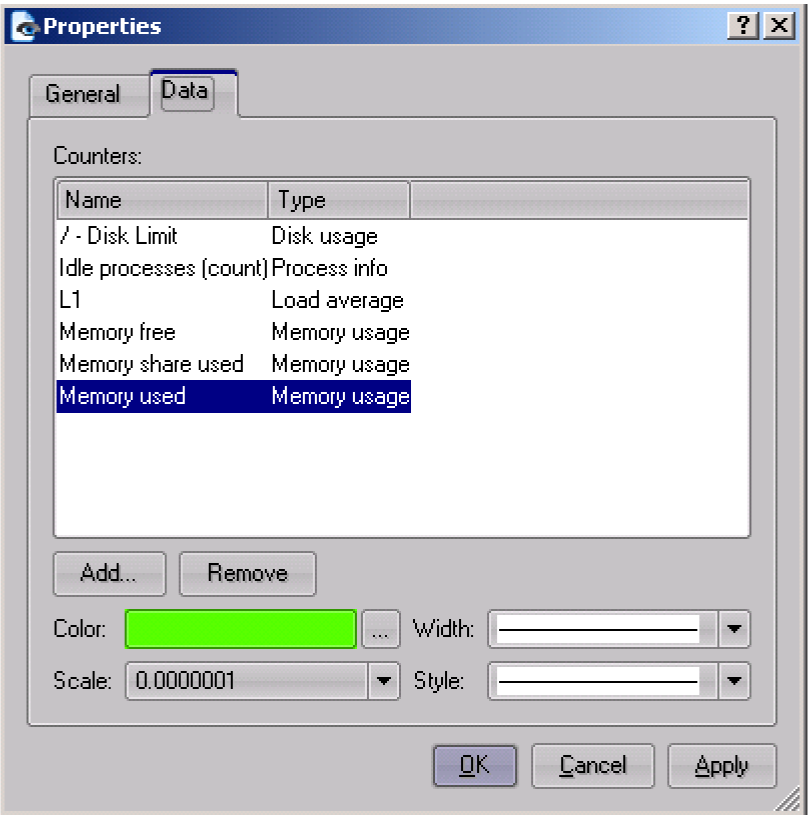
Scale: 1.0

2 Select the necessary scale on the **Scale** drop-down menu on top of the grid, and click **Apply**.

Adjusting Colors and Styles

You can define the way a counter is displayed on the grid:

1 Right-click the name of the corresponding counter in the table of displayed counters below the grid, and choose **Properties**.



Counters:

Name	Type
/ - Disk Limit	Disk usage
Idle processes (count)	Process info
L1	Load average
Memory free	Memory usage
Memory share used	Memory usage
Memory used	Memory usage

Color: 🟩

Width: ———

Scale: 0.0000001

Style: ———

2 In the corresponding boxes, adjust the color of the counter line, its width and style as desired.

3 Click the **General** tab, and adjust the view of the grid elements. The options on that tab are self-explaining.

4 Click **OK**.

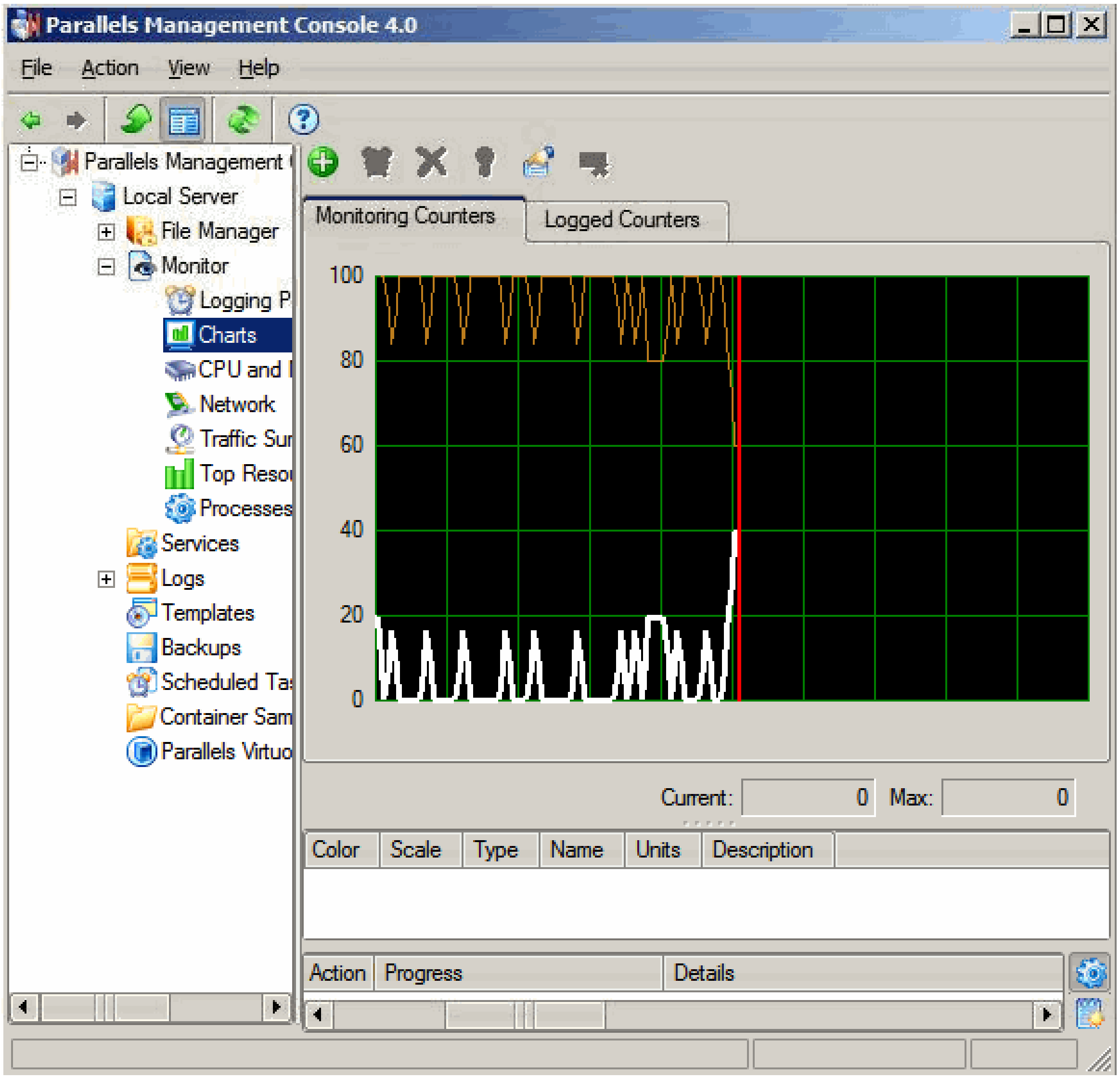
Highlighting Counters

In case there are many counters being simultaneously displayed on the grid, it might be difficult to quickly single out the needed one. Parallels Management Console provides a means for highlighting any one of the counters at a time:

1 Click the name of the corresponding counter in the table of displayed counters below the grid.

2 Click the **Highlight Counter** button on the toolbar.

The selected counter will be highlighted on the grid with a broad white line.



Saving Counters Configuration

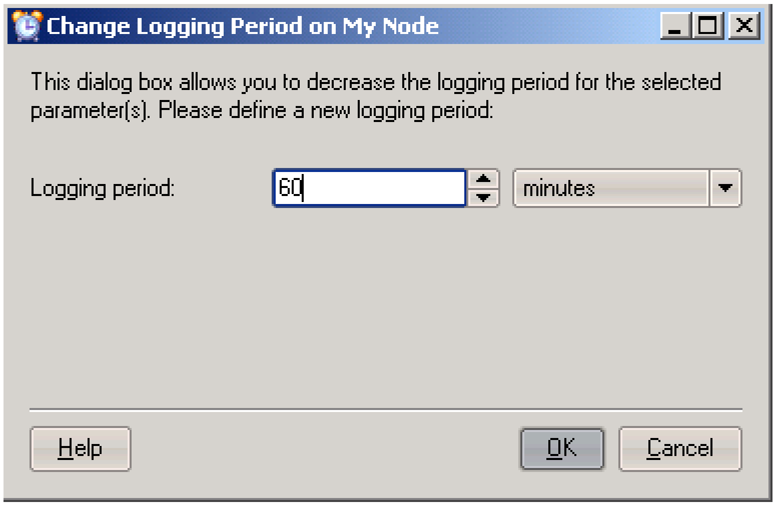
You can save the information about the current set of counters in the Parallels Management Console configuration file to call this information the next time it is needed, sparing the labor of adding the counters one by one again. Only one set of counters can thus be saved. Just right-click the counter you want to save, and choose **Save Counters**. When you alter the counters configuration (for example, when you restart Parallels Management Console, all the counters are erased) and want to restore the saved configuration, click the **Load Counters** button. The saved set of counters will be loaded from the configuration file.

Replaying Information From Logs

The function of replaying the resources consumption information over a specified time span in the past is ensured by the background logging of all the parameters in Parallels Virtuozzo Containers. The default periodicity of refreshing the resources consumption information in the logs is set to be 1 (one) hour. You can have the logs collect the resources consumption information more frequently by "accelerating" the necessary logs with the help of the **Log Setup** folder under the **Monitor** item. For example:

1 Click **Logging Period Setup** under the **Monitor** item.

2 In the right of the Management Console window, double-click the necessary log group in the **Parameters** table, or right-click it, and choose **Properties**.



Change Logging Period on My Node

This dialog box allows you to decrease the logging period for the selected parameter(s). Please define a new logging period:

Logging period: 60 minutes

3 In the **Change Logging Period** window, set the update period for the given group of logs.

4 Click **OK** for the changes to take effect.

Logs are replayed using the same grid of the **Charts** function as for real-time monitoring. The counters are also displayed and configured in the same way as for real-time monitoring. The principal difference is that when replaying the counters, the information for the charts is taken from the logs (both the default logs and the logs accelerated in the **Logging Period Setup** section are used), and not from real-time monitoring.

To switch to the charts replaying mode:

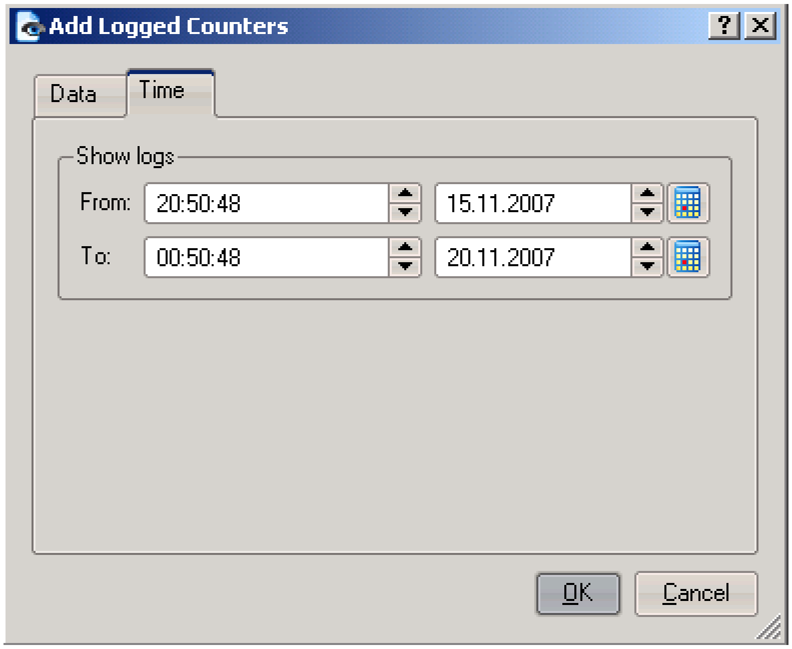
1 Click **Charts** under the **Monitor** item.

2 On the **Logged Counters** tab, click the **Add Counters** button on the toolbar to display the **Add Logged Counters** window.

3 On the **Data** tab of the **Add Logged Counters** window, click the **Add** button to add any of the available counters in the same way as they are added for real-time monitoring.

4 After adding the desired counters, adjust the style of their visualization with the help of the corresponding options on the **Data** tab.

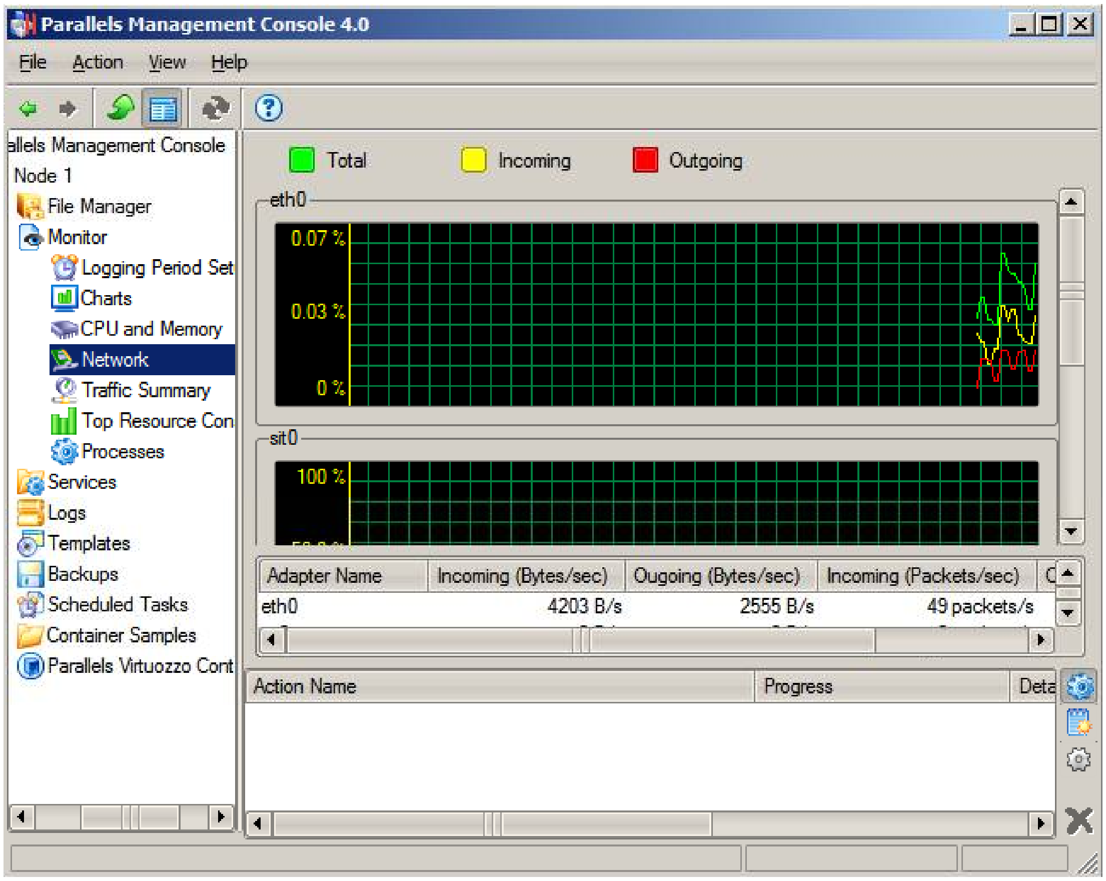
5 Go to the **Time** tab of the **Add Logged Counters** window, define the update period, and the time span for which you wish to view the logs for the specified counters. For example:



Show logs
From: 20:50:48 15.11.2007
To: 00:50:48 20.11.2007

Using Table Representation

Besides charts, it is possible to monitor many of the Hardware Node or Container parameters in real time as a list of lines each of which reflects the name and the value of a parameter, as well as the attributes specific for this or that kind of parameters. In such a way, you can view the **Network** and **Processes** groups for a particular Hardware Node, and the **Network**, **Processes**, **Resources**, and **Quotas and Usage** groups for a particular Container. Choose any of these groups either in the Management Console main window or in a Container manager window to see the real-time information about the selected parameters in the form of a table. For example, if you choose **Network** under a Hardware Node tree, you may see the following window:



Adapter Name	Incoming (Bytes/sec)	Ougoing (Bytes/sec)	Incoming (Packets/sec)	C
eth0	4203 B/s	2555 B/s	49 packets/s	

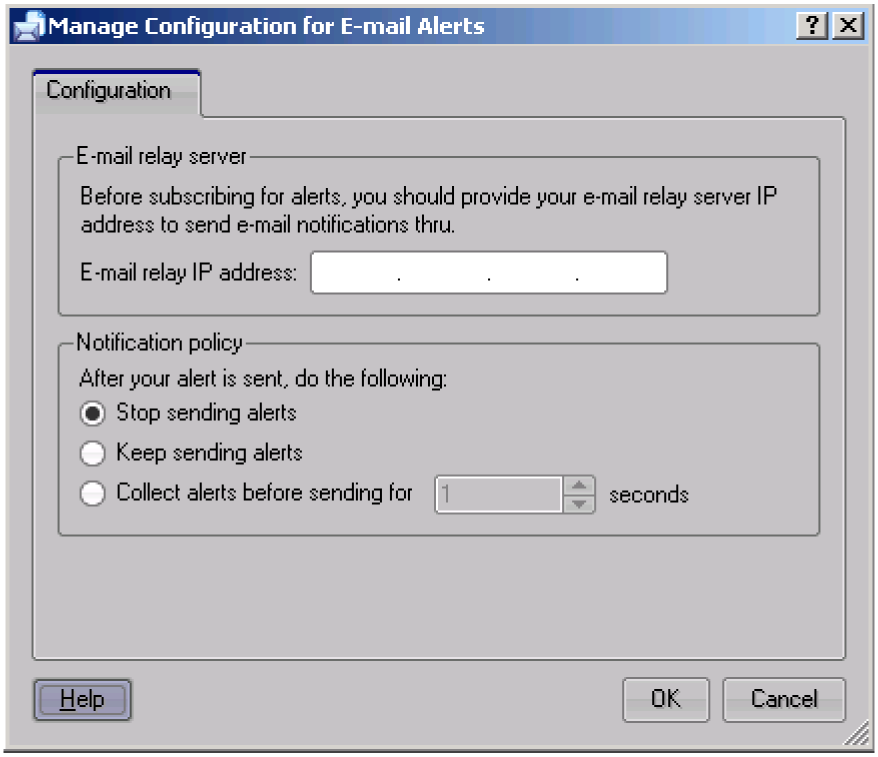
The graphic chart in the Management Console right pane shows the values for the incoming and outcoming traffic rate in bytes per second and packets per second for all the network interfaces present on the Hardware Node.

Subscribing to Parallels Management Console Alerts

Parallels Management Console allows you to subscribe to e-mail notifications about resource-overusage system alerts. The subscription to this kind of alerts consists in specifying the e-mail address to send notification to. However, prior to subscribing to alerts, you need to provide your e-mail relay server IP address to send e-mail notifications through. To do this:

1 In Parallels Management Console, click the **Manage E-mail Alert Subscription** link at the Hardware Node dashboard.

2 In the **Manage E-mail Alert Subscription** window, click the **Configure** button.



Before subscribing for alerts, you should provide your e-mail relay server IP address to send e-mail notifications thru.

E-mail relay IP address: . . .

After your alert is sent, do the following:

● Stop sending alerts

○ Keep sending alerts

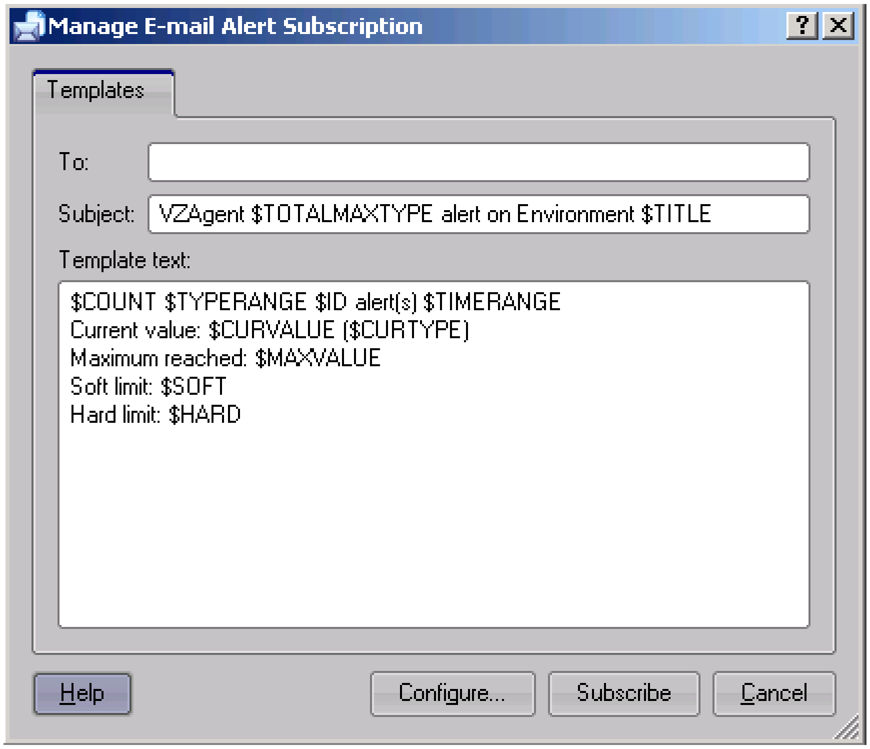
○ Collect alerts before sending for 1 seconds

3 In the displayed window, enter the IP address of the mail relay server in the **E-mail relay IP address** field.

4 Click **OK**.

Now that you have set the e-mail relay server IP address, you can subscribe to an alert:

1 Click the **Manage E-mail Alert Subscription** link at the Hardware Node dashboard.



Subject: VZAgent $TOTALMAXTYPE alert on Environment $TITLE

```
$COUNT $TYPERANGE $ID alert(s) $TIMERANGE
Current value: $CURVALUE ($CURTYPE)
Maximum reached: $MAXVALUE
Soft limit: $SOFT
Hard limit: $HARD
```

2 Type the e-mail address where the alert notification is to be sent in the **To** field.

3 Click the **Subscribe** button.

Parallels Management Console uses a pre-configured notification template. This template includes special placeholders representing special symbols that will be substituted for in the actual message by the actual Container name, parameter name, etc. The list of main placeholders is given below:

- `$TITLE`: the name assigned to the Container. If there is no name set for the Container, its hostname is used.
- `$ID`: the name of the resource parameter (in the actual message, it will be "diskspace", etc.).
- `$CURTYPE`: the alert type (at the alert generation moment). The "yellow" alert means that the barrier value lies in the range from 90% to 100% and the "red" alert indicates that the limit value has been hit.
- `$TOTALMAXTYPE`: the maximal alert type ("yellow" or "red") registered during the time when alerts were collected.
- `$COUNT`: the number of registered alerts from the time when the last e-mail notification was sent.
- `$TYPERANGE`: the range of alert types registered during the time when alerts were collected (e.g. if all types of alerts were registered, the value of this parameter in the e-mail notification will be set to "yellow" or "red").
- `$TIMERANGE`: the alert time (the server time).
- `$CURVALUE`: the current value of the parameter (at the alert generation moment).
- `$MAXVALUE`: the maximal value of the parameter during the time when alerts were collected.
- `$SOFT`: the parameter value barrier.
- `$HARD`: the parameter value limit.

By default, only one alert is sent per subscription and you have to resubscribe to an alert each time after its receiving. However, you can configure the default alert policy by doing the following:

1 Click the **Manage E-mail Alert Subscription** link at the Hardware Node dashboard.

2 In the **Manage E-mail Alert Subscription** window, click the **Configure** button.

3 In the displayed window, choose one of the following options:

   - **Stop sending alerts**. In this case, after having received an alert, you have to resubscribe to it again. This option is selected by default.
   - **Keep sending alerts**. In this case, you will get alerts on a permanent basis without having to resubscribe to them each time after their receiving.

- **Collect alerts before sending for...**. In this case, alerts will be permanently collected by Parallels Management Console in a special database. This database will be periodically, i.e. with the period specified in the field opposite the option name, checked and if there were any alerts gathered during the set time, the corresponding notification will be sent to your e-mail address. The alert checking time is measured in seconds and can be set either by using the spin button or entering the needed period by hand.

4 After you have chosen the right option, click **OK** to save the settings.

CHAPTER 6

Managing Services and Processes

This chapter provides information on what services and processes are, how they influence the operation and performance of your system, and what tasks they perform in the system.

You will learn how to use the command line utilities and Parallels Management Console to manage services and processes in Parallels Virtuozzo Containers. In particular, you will learn how to monitor active processes in your system, change the mode of the `xinetd`-dependent services, identify the Container ID where a process is running by the process ID, start, stop, or restart services and processes, and edit the service run levels.

In This Chapter

What Are Services and Processes

Instances of any programs currently running in the system are referred to as processes. A process can be regarded as the virtual address space and the control information necessary for the execution of a program. A typical example of a process is the *vi* program (on a Linux Node) running on your Hardware Node or inside your Container(s). Along with common processes, there are a great number of processes that provide an interface for other processes to call. They are called services. In many cases, services act as the brains behind many crucial system processes; they typically spend most of their time waiting for an event to occur or for a period when they are scheduled to perform some task. Many services provide the possibility for other servers on the network to connect to the given one via various network protocols.For example, the `nfs` service provides the NFS server functionality allowing file sharing in TCP/IP networks.

You may also come across the term "daemon" that is widely used in connection with processes and services. This term refers to a software program used for performing a specific function on the server system and is usually used as a synonym for "service". It can be easily identified by "d" at the end of its name. For example, `httpd` (short for HTTP daemon) represents a software program that runs in the background of your system and waits for incoming requests to a web server. The daemon answers the requests automatically and serves the hypertext and multimedia documents over the Internet using HTTP.

When working with services, you should keep in mind the following. During the lifetime of a service, it uses many system resources. It uses the CPUs in the system to run its instructions and the system's physical memory to hold itself and its data. It opens and uses files within the filesystems and may directly or indirectly use certain physical devices in the system. Therefore, in order not to damage your system performance you should run only those services on the Hardware Node that are really needed at the moment.

Besides, you should always remember that running services in the Host OS is much more dangerous than running them in Containers. In case violators get access to one of the Containers through any running service, they will be able to damage only the Container where this service is running, but not the other Containers on your Hardware Node. The Hardware Node itself will also remain unhurt. And if the service were running on the Hardware Node it would damage both the Hardware Node and all the Containers residing on it. Thus, you should make sure that you run only those services on the Hardware Node that are really necessary for its proper functioning. Please launch all additional services you need at the moment inside separate Containers. It will significantly improve your system safety.

Main Operations on Services and Processes

The ability to monitor and control processes and services in your Parallels Virtuozzo Containers system is essential because of the profound influence they have on the operation and performance of your whole system. The more you know about what each process or service is up to, the easier it will be to pinpoint and solve problems when they creep in.

The most common tasks associated with managing services in the Host Operating System of the Hardware Node or inside a Container are starting, stopping, enabling, and disabling a service. For example, you might need to start a service in order to use certain server-based applications, or you might need to stop or pause a service in order to perform testing or to troubleshoot a problem.

For `xinetd`-dependent services, you do not start and stop but enable and disable services. The services enabled in this way are started and stopped on the basis of the corresponding state of the `xinetd` daemon. Disabled services are not started whatever the `xinetd` state.

The services management is mostly disabled for the Hardware Node. Practically all the services are read-only, you are able to view the information but you cannot perform any operation on them. The reason is that many Red Hat packages determine a successful stop by looking up all the processes with a specified name. If such processes exist elsewhere, they are killed with the terminate signal. Thus, all the like services in all the Hardware Node Containers might be accidentally shut down because of this.

However, there are some services that can be managed by a number of administrative tools offered in Parallels Virtuozzo Containers. These tools allow a service to be managed and configured either by means of special Linux command-line utilities or via Parallels Management Console. You can do it either locally or from any server connected on the network. Besides, you can manage all the processes and services through Parallels Power Panel. All the necessary information on managing services and operations in Parallels Power Panel is provided in the comprehensive online help system and the user's manual Parallels Power Panel is supplied with.

As for processes, such utilities as `vzps`, `vztop`, `vzpid` enable you to see what a process is doing and to control it. Sometimes, your system may experience problems such as slowness or instability, and using these utilities should help you improve your ability to track down the causes. It goes without saying that in Parallels Virtuozzo Containers you can perform all those operations on processes you can do in the common Linux system, for example, kill a process by sending a terminate signal to it.

In Parallels Virtuozzo Containers, you can manage services and processes using both the command line and Parallels Management Console. Further in this chapter, both methods are described.

Managing Processes and Services

In Parallels Virtuozzo Containers, services and processes can be managed by using both the command line and Parallels Management Console. In the command line, you can manage the corresponding processes and services by using the following utilities:

- `vzps`
- `vzpid`
- `vztop`
- `vzsetxinetd`

With their help, you can perform the following tasks:

- Print information about active processes on your Hardware Node.
- Display the processes activity in real time.
- Change the mode of the services that can be either `xinetd`-dependent or standalone.
- Identify the Container ID where a process is running by the process ID.

Parallels Management Console allows you to manage the services present in the Host Operating System of the Hardware Node or in a Container. It allows you to monitor (and partially configure) the services of the Host operating system at the Hardware Node. By using Management Console, you can start, stop, restart a service, or edit its run levels.

Below in this chapter detailed information on all those tasks that can be performed by means of the command line utilities and Parallels Management Console is given.

Viewing Active Processes and Services

The vzps utility can be run on the Hardware Node just as the standard Linux ps utility. It provides certain additional functionality related to monitoring separate Containers running on the Node, namely, you can use the -E switch with the vzps utility to:

- display the Container IDs where the processes are running
- view the processes running inside a particular Container

vzps prints information about active processes on your Hardware Node. When run without any options, vzps lists only those processes that are running on the current terminal. Below is an example output of the vzps run:

```
# vzps
  PID TTY          TIME CMD
 4684 pts/1   00:00:00 bash
27107 pts/1   00:00:00 vzps
```

Currently, the only processes assigned to the user/terminal are the bash shell and the vzps command itself. In the output, the PID (Process ID), TTY, TIME, and CMD fields are contained. TTY denotes which terminal the process is running on, TIME shows how much CPU time the process has used, and CMD is the name of the command that started the process.

Note: Starting with Virtuozzo 3.0, the IDs of the processes running inside Containers and displayed by running the vzps command on the Hardware Node does not coincide with the IDs of the same processes shown by running the ps command inside these Containers.

As you can see, the standard vzps command just lists the basics. To get more details about the processes running on your Hardware Node, you will need to pass some command line arguments to vzps. For example, using the aux arguments with this command displays processes started by other users (a), processes with no terminal or one different from yours (x), the user who started the process and when it began (u). Besides, you can pass vzps the -E switch, which is specific for Parallels Virtuozzo Containers, to sort the processes by the Container IDs where they are running.

```
# vzps aux -E
USER  PID %CPU %MEM   VSZ  RSS TTY   STAT START  TIME COMMAND
root    1  0.0  0.0  1516  128 ?     S   Jul14   0:37 init
root    5  0.0  0.0     0    0 ?     S   Jul14   0:03 [ubstatd]
root    6  0.0  0.0     0    0 ?     S   Jul14   3:20 [kswapd]
#27     7  0.0  0.0     0    0 ?     S   Jul14   0:00 [bdflush]
root    9  0.0  0.0     0    0 ?     S   Jul14   0:00 [kinoded]
root 1574  0.0  0.1   218  140 pts/4 S   09:30   0:00 -bash
```

There is a lot more information now. The fields USER, %CPU, %MEM, VSZ, RSS, STAT, and START have been added. Let us take a quick look at what they tell us.

The USER field shows you which user initiated the command. Many processes begin at system start time and often list root or some system account as the USER. Other processes are, of course, run by individuals.

The %CPU, %MEM, VSZ, and RSS fields all deal with system resources. First, you can see what percentage of the CPU the process is currently utilizing. Along with CPU utilization, you can see the current memory utilization and its VSZ (virtual memory size) and RSS (resident set size). VSZ is the amount of memory the program would take up if it were all in memory; RSS is the actual amount currently in memory. Knowing how much a process is currently eating will help determine if it is acting normally or has spun out of control.

You will notice a question mark in most of the TTY fields in the `vzps aux` output. This is because most of these programs were started at boot time and/or by initialization scripts. The controlling terminal does not exist for these processes; thus, the question mark. On the other hand, the `bash` command has a TTY value of pts/4. This is a command being run from a remote connection and has a terminal associated with it. This information is helpful for you when you have more than one connection open to the machine and want to determine which window a command is running in.

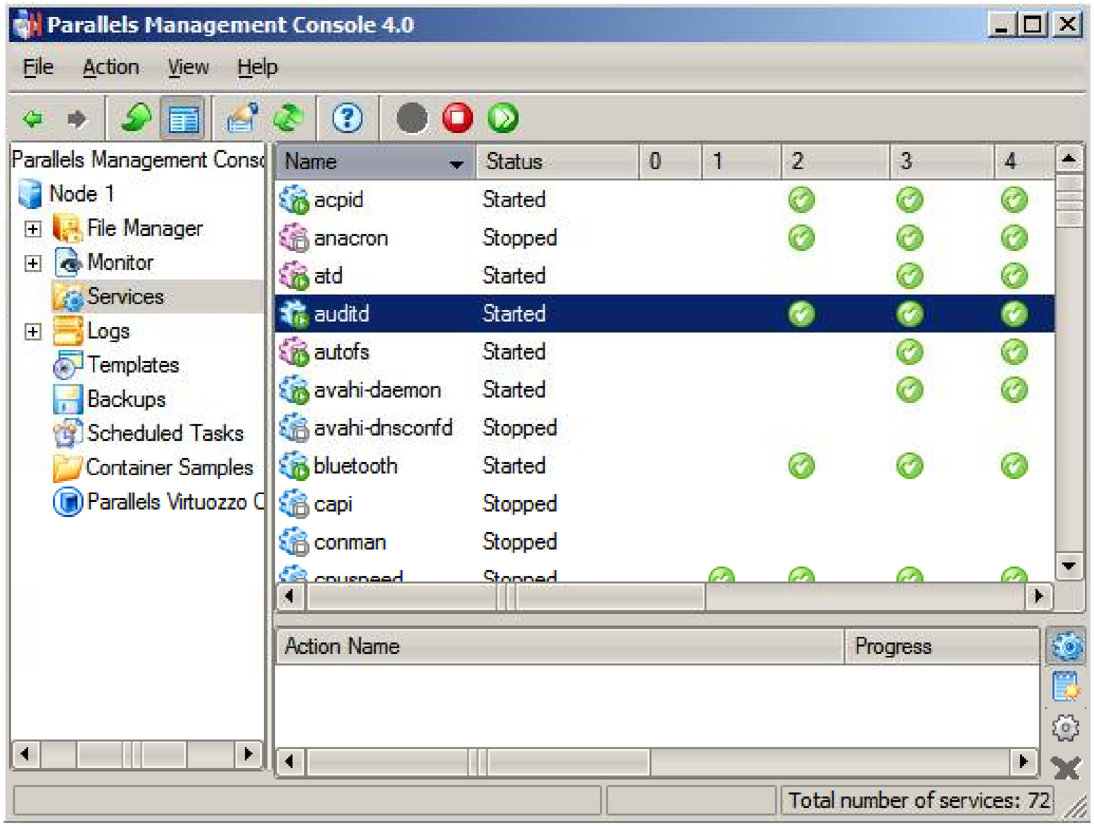
STAT shows the current status of a process. In our example, many are sleeping, indicated by an S in the STAT field. This simply means that they are waiting for something. It could be user input or the availability of system resources. The other most common status is R, meaning that it is currently running.

Note: For detailed information on all `vzps` parameters, output fields, states of processes, etc., please consult the `vzps` manual pages.

In the current version of Parallels Virtuozzo Containers, you can also use the `vzps` command to view the processes currently running inside any Containers on the Hardware Node. The example below shows you how to display all active processes inside Container 101:

```
# vzps -E 101
CTID   PID TTY          TIME CMD
101 27173 ?        00:00:01 init
101 27545 ?        00:00:00 syslogd
101 27555 ?        00:00:00 sshd
101 27565 ?        00:00:00 xinetd
101 27576 ?        00:00:03 httpd
101 27583 ?        00:00:00 httpd
101 27584 ?        00:00:00 httpd
101 27587 ?        00:00:00 crond
101 27596 ?        00:00:00 saslauthd
```

In Parallels Management Console, you can monitor the services running in the Host operating system of the Hardware Node or inside a Container. Click on the **Services** item in the tree below the Hardware Node name. The list of the Host OS or Container OS services appears in the right pane.



Name	Status	0	1	2	3	4
acpid	Started			✓	✓	✓
anacron	Stopped			✓	✓	✓
atd	Started				✓	✓
auditd	Started			✓	✓	✓
autofs	Started				✓	✓
avahi-daemon	Started				✓	✓
avahi-dnsconfd	Stopped					
bluetooth	Started			✓	✓	✓
capi	Stopped					
conman	Stopped					
cpuspeed	Stopped		✓	✓	✓	✓

Total number of services: 72

You can also view the services running in a Container by clicking the **Services** item in the Container manager window.

The way the services are colored reflects the importance of a service for Parallels Virtuozzo Containers: pink icons are for services that are critical for Parallels Virtuozzo Containers and yellow icons are for services that are not that critical.

Running services are indicated with bright icons. Stopped services have shaded icons. The **Status** column of the table duplicates this information in the text form. The default run levels of services are ticked off in the corresponding table columns.

To facilitate working with services, you can sort them by different parameters: their name, status, and so on. Just click the column with the appropriate name to put services in the desired order.

Monitoring Processes in Real Time

The vztop utility is rather similar to vzps but is usually started full-screen and updates continuously with process information. This can help with programs that may infrequently cause problems and can be hard to see with vzps. Overall system information is also presented, which makes a nice place to start looking for problems.

The vztop utility can be run on the Node just as the standard Linux top utility. The only features that distinguish the vztop utility from top are the following:

- vztop allows you to use the -E option that monitors only the processes belonging to the Container whose processes you want to display.
- You can use the e interactive command to temporarily view/hide the CTIDs where the processes are running.
- You can use the E interactive command to set the filter on the CTID field that helps you display only the processes belonging to the given Container.

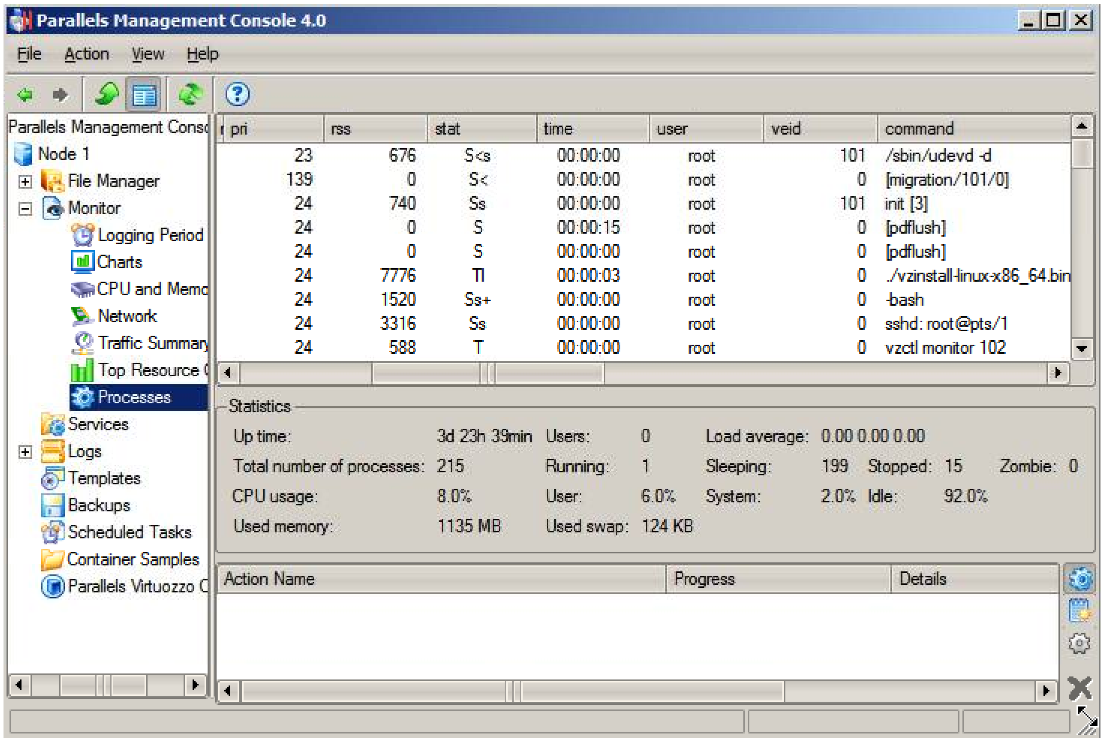
The vztop utility usually has an output like the following:

```
# vztop -E 101
17:54:03  up 20 days, 23:37, 4 users, load average: 2.13, 1.89, 1.75
305 processes: 299 sleeping, 3 running, 3 zombie, 0 stopped
CPU0 states: 20.1% user 51.2% system  0.0% nice 0.0% iowait 28.1% idle
CPU1 states: 21.2% user 50.0% system  0.0% nice 0.0% iowait 28.1% idle
Mem: 1031088k av, 969340k used, 61748k free, 0k shrd, 256516k buff
       509264k active,            330948k inactive
Swap: 4056360k av,   17156k used, 4039204k free      192292k cached
CTID   PID USER PR  NI  VIRT  RES  SHR S %CPU %MEM  TIME+  COMMAND
101  27173 root 16  0   1616  604 1420 S  0.0  0.1  0:01.86 init
101  27545 root 16  0   1520  624 1356 S  0.0  0.1  0:00.34 syslogd
101  27555 root 25  0   4008 1700 3632 S  0.0  0.4  0:00.04 sshd
101  27565 root 25  0   2068  860 1740 S  0.0  0.2  0:00.05 xinetd
101  27576 root 16  0   7560 3180 6332 S  0.0  0.7  0:03.78 httpd
101  27587 root 16  0   2452 1036 1528 S  0.0  0.2  0:00.34 crond
101  27596 root 25  0   4048 1184 3704 S  0.0  0.2  0:00.01 saslauthd
```

As you can see, vztop provides an ongoing look at the processor activity in real time (the display is updated every 5 seconds by default, but you can change that with the d command-line option or the s interactive command). It displays a list of the most CPU-intensive tasks on the system and can provide an interactive interface for manipulating processes. It can sort the tasks by CPU usage, memory usage, and runtime. Specifying 101 after the –E option allows you to display only those processes that are running inside Container 101 only. Besides, most features can be selected by an interactive command, for example, the e and E commands described above.

Note: For more information on all vztop parameters, consult its man pages. Besides, you can find information on some fields in the **Viewing Active Processes** subsection (p. 200).

In Parallels Management Console, you can view those processes that are currently running on the Hardware Node and in Container. To display the processes, click the Hardware Node name where you want to monitor processes, and choose **Monitor** > **Processes**. The list of the Host OS and Container OS processes appears in the right pane.



pri	rss	stat	time	user	veid	command
23	676	S<s	00:00:00	root	101	/sbin/udevd -d
139	0	S<	00:00:00	root	0	[migration/101/0]
24	740	Ss	00:00:00	root	101	init [3]
24	0	S	00:00:15	root	0	[pdflush]
24	0	S	00:00:00	root	0	[pdflush]
24	7776	Tl	00:00:03	root	0	./vzinstall-linux-x86_64.bin
24	1520	Ss+	00:00:00	root	0	-bash
24	3316	Ss	00:00:00	root	0	sshd: root@pts/1
24	588	T	00:00:00	root	0	vzctl monitor 102

Statistics

Up time:	3d 23h 39min	Users:	0	Load average:	0.00 0.00 0.00				
Total number of processes:	215	Running:	1	Sleeping:	199	Stopped:	15	Zombie:	0
CPU usage:	8.0%	User:	6.0%	System:	2.0%	Idle:	92.0%		
Used memory:	1135 MB	Used swap:	124 KB						

Action Name	Progress	Details

The column names and their description is presented in the table below:

Column name	Description
`pid`	The identifier of the process.
`%cpu`	The CPU time, in percent, used by the process.
`%mem`	The memory used by the process.
`ni`	The 'nice' parameter; weights the overall scheduling priority for the process.
`pri`	The kernel scheduling priority for the process.
`rss`	Number of resident pages for the swap-out guarantee (the resident set size).
`stat`	The process current status. Can be '`R`' (running), '`S`' (sleeping, waiting for 'wake-up call)', '`D`' (uninterruptable sleep), '`Z`' (zombie, waiting for parent process), '`T`' (stopped or traced). Sometimes the second symbol may appear: '`W`' (process swapping), '`N`' ('niced' process), '`L`' (process has pages locked into memory). If the < sign is displayed after the status, it means that this information was returned by the Parallels Agent software which, in turn, got this information from the `ps` tool.
`time`	The total CPU time the process has used.
`user`	The user who has launched the process.
`veid`	The ID of the Container where the process is running.
`command`	The command that invoked the process.

To view the processes inside a particular Container, double-click on its name, and select **Monitor** > **Processes**.

Note: Starting with Virtuozzo 3.0, the IDs of the processes running inside Containers displayed by selecting **Monitor** > **Processes** on the Hardware Node does not coincide with the IDs of the same processes shown when opening the Container Manager window and selecting **Monitor** > **Processes**.

You can send different signals to process by right-clicking a process and selecting the corresponding signal on the pop-up menu.

Changing Services Mode

xinetd is a service used to start and stop a variety of data communication services. xinetd starts on the Hardware Node startup and waits for a connection request from a remote client that wants to connect to the server. There can be a number of remote clients in the network, and each of them can use different network protocols to establish connection to the server. In order not to run all network services responsible for a specific protocol, which will negatively influence the system performance, the system starts only the xinetd service. This service controls all other network services and, at the connection time, it starts the corresponding service to process this connection. In such a way, xinetd saves system resources allowing you to run only those network services in the system that are really needed at the moment.

The vzsetxinetd utility allows you to switch Container services between the standalone and xinetd mode. The services that can be either standalone or dependent on xinetd are sendmail, sshd, proftpd, and courier-imap. Whereas they are xinetd-dependent by default, in order to consume less resources, you may want to make them standalone due to the following reasons:

- The CPanel application does not recognize sshd if it is dependent on xinetd.
- sendmail does not process some rules correctly if it is dependent on xinetd.
- A number of control panel applications and some others are not able to manage xinetd-based services at all.

The courier-imapd, courier-imapds, courier-pop3d, and courier-pop3ds services are provided by the courier-imap service, thus vzsetxinetd can manage these services via the courier-imap service.

Let us assume that you wish to check the mode of the sendmail service and set it to standalone if it is in the xinetd mode. First, you should check the current status of the sendmail service. To this effect, type the following command in the command line:

```
# vzsetxinetd -s 222 sendmail
```

where 222 is the Container ID, sendmail denotes the name of the corresponding service, and the -s option gets the status of the sendmail service of the Container with ID 222. The output will tell you if this service has the standalone or xinetd mode:

```
sendmail is xinetd service
```

In our case it is in the xinetd mode. Now you can change the mode of the sendmail service to standalone. To make it standalone, type the following line:

```
# vzsetxinetd 222 sendmail off
sendmail is standalone service
```

where off specifies that the sendmail service should be set to the standalone mode. The output confirms that the sendmail service is now standalone.

For more information on the vzsetxinetd utility, please consult the corresponding man pages or turn to the *Parallels Virtuozzo Containers 4.7 Reference Guide*.

Note: You cannot use the vzsetxinetd utility to change the mode of the xinetd-dependent services in Containers where the Debian 3.0 OS template is installed.

Determining Container Identifiers by Process IDs

Each process is identified by a unique PID (process identifier), which is the entry of that process in the kernel's process table. For example, when you start Apache, it is assigned a process ID. This PID is then used to monitor and control this program. The PID is always a positive integer. In Parallels Virtuozzo Containers, you can use the vzpid (retrieve process ID) utility to print the Container ID the process with the given id belongs to. Multiple process IDs can be specified as arguments. In this case the utility will print the Container number for each of the processes.

The typical output of the vzpid utility is shown below:

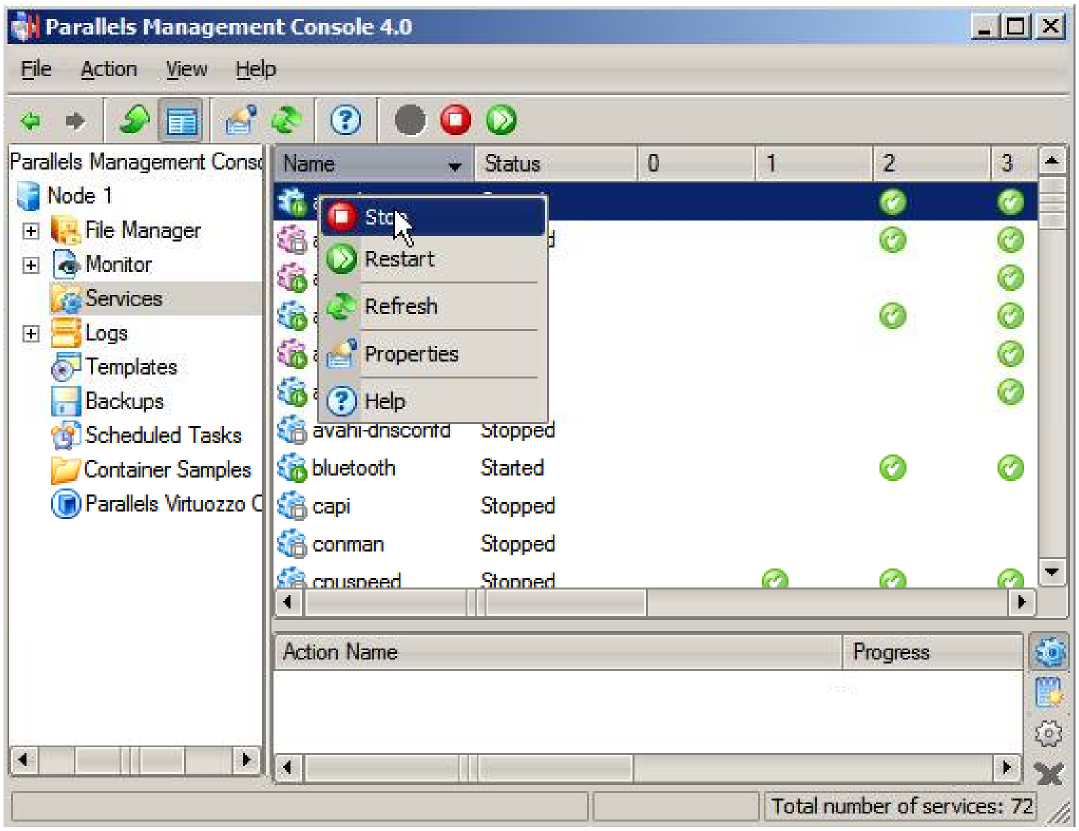
```
# vzpid 12
Pid     VEID     Name
12       101     init
```

In our example the process with the identifier 12 has the name 'init' and is running in the Container with ID 101.

Note: You can also display the Container ID where the corresponding process is running by using the vzps utility.

Starting, Stopping, and Restarting Services

Parallels Management Console allows you to manage the services present in the Host operating system of the Hardware Node or in a Container. Click the **Services** item in the tree below the Hardware Node nameor the Container name. The list of the Host OS or Container OS services appears in the right pane.



Name	Status	0	1	2	3
avahi-dnsconfd	Stopped				
bluetooth	Started			✓	✓
capi	Stopped				
conman	Stopped				
cpuspeed	Stopped		✓	✓	✓

To start, stop, or restart a service, select its line in the table and either use the pop-up menu or the buttons on the toolbar. For `xinetd`-dependent services (the services having `xinetd` in parentheses beside their name), you do not start and stop but enable and disable services. The services enabled in this way are started and stopped on the basis of the corresponding state of the `xinetd` daemon. Disabled services are not started whatever the `xinetd` state.

To edit the default run levels for the service, use the **Properties** item on the context menu or just double–click on the service name within the list. When the **Properties** dialog is open, select the check boxes of the run levels on which the service will start automatically. Click the **OK** button to apply your settings. If the service is dependent on `xinetd`, you cannot choose its run levels, as the latter are determined by the `xinetd` daemon. Besides, you cannot change run levels for certain services, which means that they are critical and you are not allowed to change their run levels.

You can also manage (i.e. start, stop, and restart) services by using the command line. For example, you wish to start the `httpd` service. To do this, execute the following command:

```
[root@ct222 /]# service httpd start
```

where `service` is the standard Linux command, `httpd` denotes the name of the corresponding service, and `start` is the command that will launch this service. In order to check that the `httpd` service was successfully launched, you can either type the following Linux command:

```
[root@ct222 /]# service httpd status
```

or use the `vzps` utility when working on your Node or the `ps` utility when working inside your Containers and passing them the `x` argument. The output will tell you if the `httpd` service is running in your system or not.

CHAPTER 7

Managing Parallels Virtuozzo Containers Network

The given chapter familiarizes you with the Parallels Virtuozzo Containers network structure, enumerates Parallels Virtuozzo Containers networking components, and explains how to manage these components in Parallels Virtuozzo Containers-based systems. In particular, it provides the following information:

- How you can manage physical and VLAN adapters on the Hardware Node.
- What Virtual Networks are and how you can manage them on the Hardware Node.
- What the `venet0` networking mode is and how to make your Containers operate in this mode.
- What the `veth` networking mode is and how to make your Containers operate in this mode.
- How to create `veth` virtual network adapters inside your Containers and configure their parameters.
- How to connect Containers to LANs (Local Area Networks) and VLANs (Virtual Local Area Networks).

In This Chapter

Managing Network Adapters on the Hardware Node

Physical and VLAN (Virtual Local Area Network) adapters installed on the Hardware Node are used to provide Containers with access to each other and to external networks. In Parallels Virtuozzo Containers, you can perform the following operations on adapters:

- List the adapters currently installed on the Hardware Node
- Create new VLAN adapters on the Hardware Node
- Connect adapters to Virtual Networks on the Hardware Node

These operations are described in the following subsections in detail.

Listing Adapters

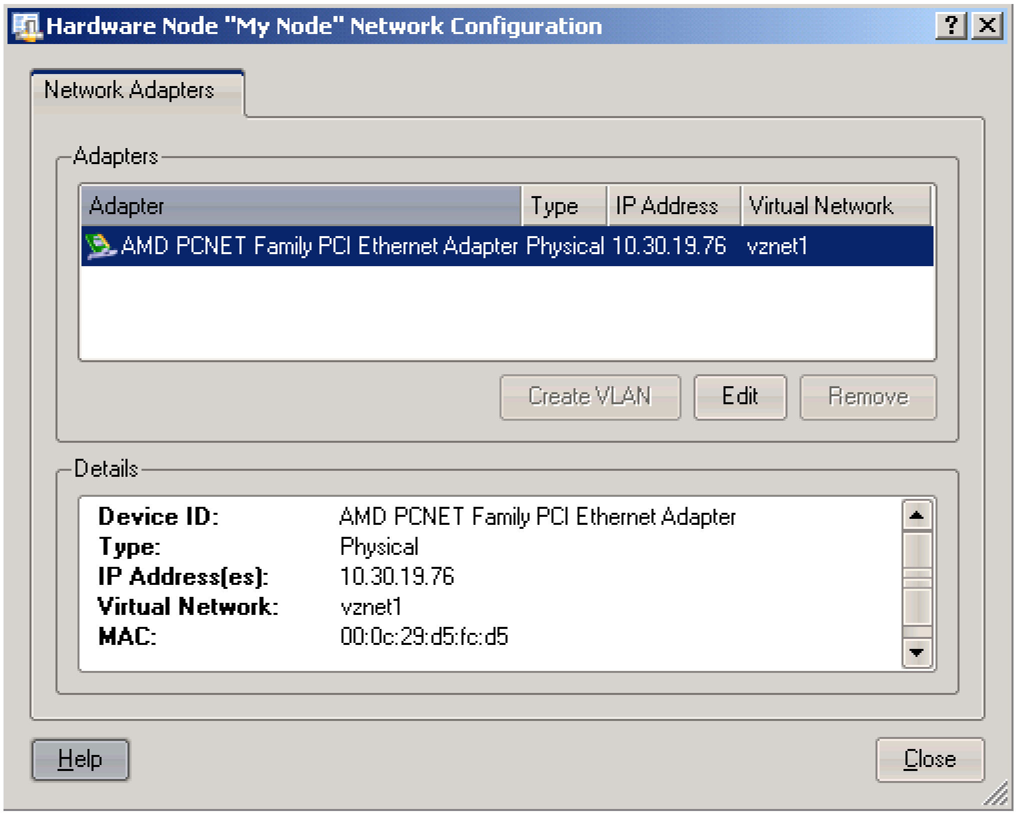
You can view the physical and VLAN network adapters currently installed on your Hardware Node using the `vznetcfg` utility. For example, you can execute the following command to find out what network adapters are available on your Node:

```
# vznetcfg if list
Name      Type    Network ID    Addresses
eth0      nic                   192.168.0.170/22,dhcp
```

As can be seen from the command output, only one physical adapter - `eth0` - is currently installed on the Hardware Node. The information on adapters produced by `vznetcfg` is presented in the table having the following columns:

Column Name	Description
Name	The adapter name.
Type	The type of the network adapter which can be one of the following: • `nic` denotes a physical adapter; • `vlan` stands for a VLAN adapter.
Network ID	The ID of the Virtual Network where the network adapter is connected. Detailed information on Virtual Networks is provided in the **Managing Parallels Virtuozzo Containers Networks** section (p. 217).
Addresses	The IP address(es) and subnet mask(s) assigned to the network adapter.

In Parallels Management Console, you can list all available adapters on the Node by right-clicking the needed Hardware Node and choosing **Network Configuration** > **Configure Network Adapters**.



Adapter	Type	IP Address	Virtual Network
AMD PCNET Family PCI Ethernet Adapter	Physical	10.30.19.76	vznet1

Device ID:	AMD PCNET Family PCI Ethernet Adapter
Type:	Physical
IP Address(es):	10.30.19.76
Virtual Network:	vznet1
MAC:	00:0c:29:d5:fc:d5

The **Adapters** table in the displayed window lists all the network adapters currently available on the Node. To view detailed information on the corresponding adapter, select its name in the **Adapters** table. All adapter-related data (its name, type, the MAC and IP address assigned to the adapter, etc.) will be shown in the **Details** table at the bottom of the **Hardware Node Network Configuration** window.

Creating a VLAN Adapter

Parallels Virtuozzo Containers allows you to create new VLAN adapters on the Hardware Node. You can use these adapters later on to connect your Containers to any of the available Virtual Networks (for more information on Virtual Networks, please turn to the **Managing Virtual Networks** section (p. 217)). 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 Hardware Node - and the physical network adapter on the Node to which the VLAN is to be bound. For example, you can execute the following command to make a new VLAN adapter on the Node, 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 Hardware Node:

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

To check that the VLAN adapter has been successfully created on the Hardware Node, you can 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, there is only one VLAN adapter currently existing on the Hardware Node. It is assigned the name of `eth0.5` which is automatically generated on the basis of the specified VLAN ID and the name of the physical adapter to which the VLAN adapter is tied.

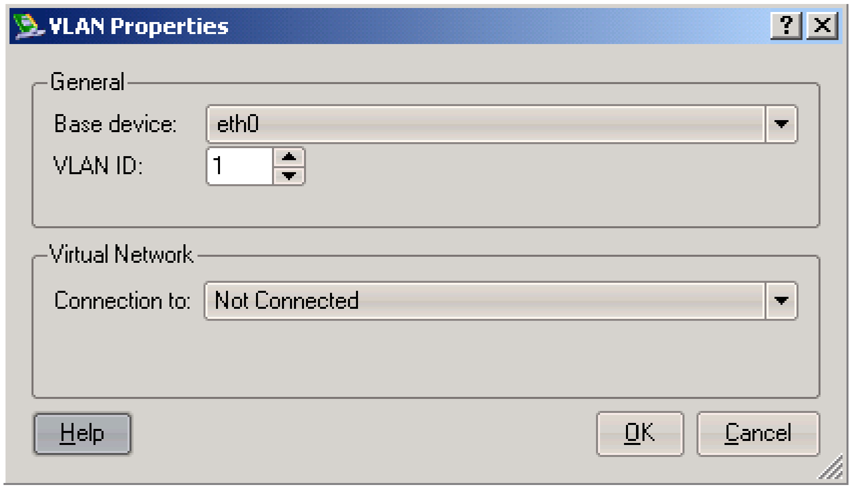
At any time you can delete the `eth0.5` VLAN adapter and thus destroy VLAN 5 by issuing the following command on the Node:

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

To create a new VLAN adapter in Parallels Management Console, do the following:

1 Right-click the needed Hardware Node, and choose **Network Configuration** > **Configure Network Adapters**.

2 In the **Hardware Node Network Configuration** window, click the **Create VLAN** button.


VLAN Properties

General

Base device: eth0

VLAN ID: 1

Virtual Network

Connection to: Not Connected

3 The **VLAN Properties** window allows you to set the following parameters for the VLAN adapter:

- **Base device**: choose the physical network adapter on the Hardware Node where the VLAN adapter is to be bound.
- **VLAN ID**: specify the VLAN ID—an arbitrary integer number which will uniquely identify the virtual LAN among other VLANs on the Hardware Node.

4 Click **OK**.

At any time, you can remove any of the VLAN adapters existing on the Hardware Node by selecting its name in the **Adapters** table and clicking the **Remove** button at the bottom of the table.

Note: By default, all VLANs created on the Hardware Node by means of Parallels Virtual Automation, Parallels Management Console, or the `vznetcfg` utility are in the 'down' state. To enable a newly created VLAN, assign a valid IP address to it and then bring the VLAN to the running state using the Linux `ip` utility.

Connecting Adapters to Virtual Networks

Connecting a physical or VLAN adapter to a Virtual Network allows you to join all Containers included in the Virtual Network to the network (either LAN or VLAN) where the corresponding adapter is connected.

Let us assume the following:

- The `eth0` physical adapter and the `vznetwork1` Virtual Network exist on the Hardware Node. For information on how to create Virtual Networks, please turn to the **Creating Virtual Networks** subsection (p. 218).
- The `eth0` physical adapter is connected to the local network.
- Container 101 and Container 102 are connected to the `vznetwork1` Virtual Network. Detailed information on how to join Containers to Virtual Networks is given in the **Connecting Containers to Virtual Networks** subsection.

To connect the `eth0` adapter to the `vznetwork1` Virtual Network and thus to join Container 101 and 102 to the local network, you should issue the following command on the Node:

```
# vznetcfg net addif vznetwork1 eth0
```

To check that the `eth0` physical adapter has been successfully added to the `vznetwork1` Virtual Network, you can execute the following command:

```
# vznetcfg if list
Name     Type   Network ID   Addresses
eth0     nic    vznetwork1   192.168.0.170/22,dhcp
...
```

As you can see, the `eth0` adapter is now joined to the `vznetwork1` Virtual Network, which means that Container 101 and 102 whose virtual network adapters are connected to `vznetwork1` can access the local network behind `eth0`.

At any time you can disconnect the `eth0` physical adapter from the `vznetwork1` Virtual Network (and thus detach Container 101 and 102 from the local network) by running the following command:

```
# vznetcfg net delif eth0
```

To check that the physical adapter has been successfully disconnected from `vznetwork1`, issue the following command:

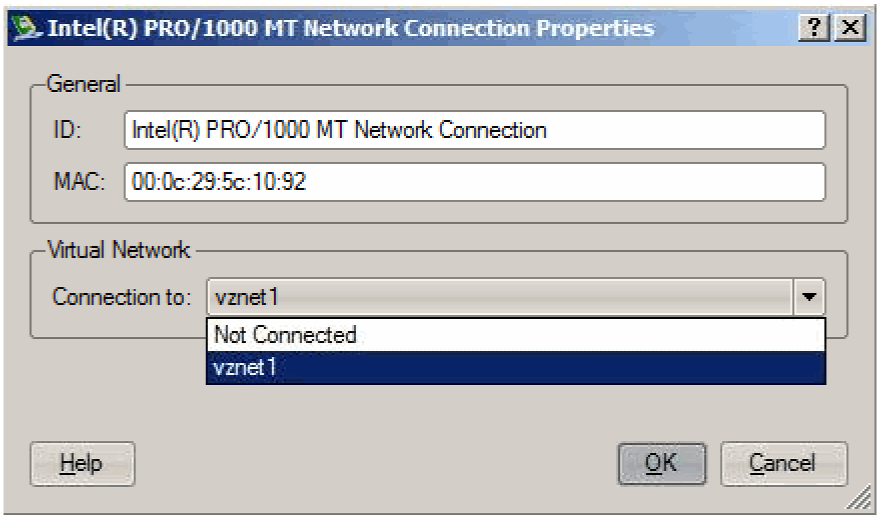
```
# vznetcfg if list
Name     Type   Network ID   Addresses
eth0     nic                 192.168.0.170/22,dhcp
...
```

To join a physical or VLAN adapter to a Virtual Network in Parallels Management Console, do the following:

1 Right-click the needed Hardware Node, and choose **Network Configuration** > **Configure Network Adapters**.

2 In the **Hardware Node Network Configuration** window, select the name of the network adapter (either physical or VLAN) to be joined to a Virtual Network, and click **Edit** button.

3 Under **Virtual Network**, choose on the drop-down menu the Virtual Network where you want to join the network adapter.



Intel(R) PRO/1000 MT Network Connection Properties

General

ID: Intel(R) PRO/1000 MT Network Connection

MAC: 00:0c:29:5c:10:92

Virtual Network

Connection to: vznet1

4 Click **OK**.

To disconnect an adapter from the corresponding Virtual Network, perform **Steps 1** and **2** above and, in the **Properties** window, choose `Not connected` on the drop-down menu.

Managing Virtual Networks

A Virtual Network acts as a binding interface between a Container virtual network adapter and the corresponding physical or VLAN adapter on the Hardware Node allowing you to include your Containers in different networks (local or VLAN). Parallels Virtuozzo Containers enables you to manage Virtual Networks as follows:

- Create a new Virtual Network on the Hardware Node and remove an existing one.
- List the Virtual Networks currently existing on the Hardware Node and configure their properties.
- Delete a Virtual Network that you do need any more from the Hardware Node.

These operations are described in the following subsections in detail.

Creating a Virtual Network

Virtual Networks serve as binding interfaces between the `veth` virtual network adapters inside Containers and the physical/VLAN adapters on the Hardware Node allowing you to connect the corresponding Containers to different LANs and VLANs. New Virtual Networks can be created using the `vznetcfg` utility. For example, to make a new Virtual Network with the name of `vznetwork1`, you can issue the following command:

```
# vznetcfg net new vznetwork1
```

To check that `vznetwork1` has been successfully created on the Hardware Node, you can execute the following command:

```
# vznetcfg net list
Network ID   Status  Master Interface Slave Interfaces
vznetwork1   active
```

You can see that the `vznetwork1` Virtual Network is now available on the Node.

Each Virtual Network is associated with some bridge which is automatically made on the Hardware Node 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:

- Issue the following command:

```
# vznetcfg if list
Name                 Type        Network ID        Addresses
eth0                 nic         vznetwork1        192.168.0.150/22,dhcp
br0                  bridge      vznetwork1
...
```

  The command output that the `vznetwork1` Virtual Network is bound to the `br0` bridge on the Node.

- Check the `/etc/vz/vznet.conf` file on the Node:

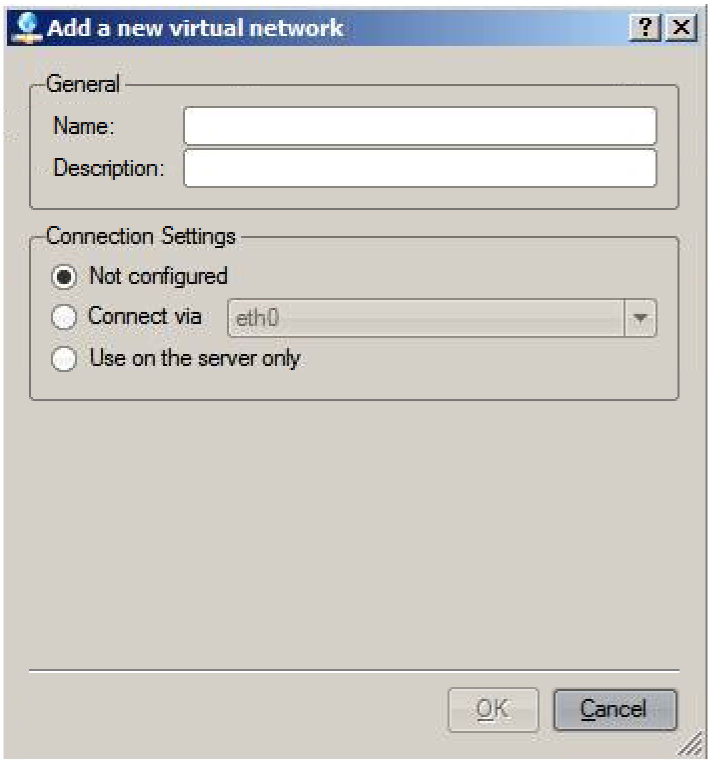
```
# cat /etc/vz/vznet.conf
VNID_br0="vznetwork1"
...
```

  In the output above, the name of the bridge - `br0` - is a component of the `VNID_br0` parameter defining the Virtual Network name.

Note: Detailed information on the `vznetcfg` utility and the `/etc/vz/vznet.conf` file is provided in the *Parallels Virtuozzo Containers 4.7 Reference Guide*.

To create a new Virtual Network in Parallels Management Console, do the following:

1. Right-click the needed Hardware Node, and choose **Network Configuration** > **Configure Virtual Networks**.
2. In the **Virtual Networks** window, click the **Add** button.



Name:

Description:

● Not configured

○ Connect via eth0

○ Use on the server only

3 Specify an arbitrary name for the Virtual Network in the **Name** field, and provide its description, if necessary, in the **Description** field.

4 Click **OK**.

Listing Virtual Networks

Sometimes, you may wish to list all Virtual Networks currently existing on the Hardware Node. To do this, you should execute the following command on the Hardware Node:

```
# vznetcfg net list
Network ID   Status   Master Interface  Slave Interfaces
vznetwork1   active   eth0
vznetwork2   active
```

In the example above, two Virtual Networks - `vznetwork1` and `vznetwork2` - exist on the Hardware Node. The information on these Virtual Networks is presented in the table having the following columns:

`Network ID` The name assigned to the Virtual Network.

`Status` Indicates the status of the Virtual Network. It can be one of the following:

- `active`: the Virtual Network is up and running.
- `configured`: the information on the Virtual Network is present in the `/etc/vz/vznet.conf` file on the Hardware Node; however, the bridge to which the Virtual Network is bound is down or absent from the Node.

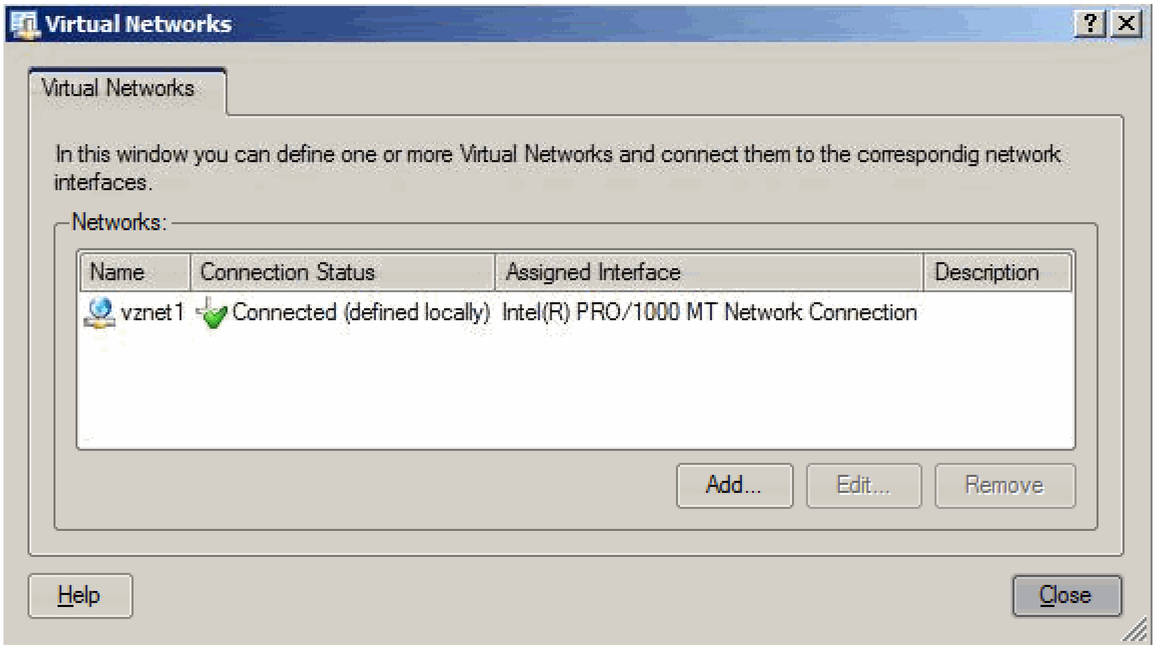
> **Note:** Detailed information on the `vznet.conf` file is given in the *Parallels Virtuozzo Containers 4.7 Reference Guide*.

`Master Interface` The name of the physical/VLAN adapter on the Hardware Node connected to the Virtual Network, if any.

`Slave Interfaces` The name of the `veth` virtual network adapters joined to the Virtual Network, if any.

To list the Virtual Networks in Parallels Management Console, do the following:

1 Right-click the needed Hardware Node, and choose **Network Configuration** > **Configure Virtual Networks**.



In this window you can define one or more Virtual Networks and connect them to the correspondig network interfaces.

Networks:

Name	Connection Status	Assigned Interface	Description
vznet1	Connected (defined locally)	Intel(R) PRO/1000 MT Network Connection	

2 The **Virtual Networks** window lists all the Virtual Networks currently existing on the Hardware Node.

Deleting a Virtual Network

At any time, you can remove a Virtual Network that you do not need any more from the Hardware Node. For example, you can delete the `vznetwork1` Virtual Network by running the following command:

```
# vznetcfg net del vznetwork1
```

To check that `vznetwork1` has been successfully remove from the Node, issue the following command:

```
# vznetcfg net list
Network ID   Status  Master Interface Slave Interfaces
vznetwork2   active
```

Note: Detailed information on the `vznetcfg` utility and all its options is provided in the *Parallels Virtuozzo Containers 4.7 Reference Guide* and the `vznetcfg` manual pages.

To remove an existing Virtual Network from the Hardware Node in Parallels Management Console, do the following:

1 Right-click the needed Hardware Node, and choose **Network Configuration** > **Configure Virtual Networks**.

2 In the **Virtual Networks** window, select the name of the Virtual Network you wish to delete, and click the **Remove** button.

Managing Virtual Network Adapters

Parallels Virtuozzo Containers provides you with ample opportunities of configuring virtual network adapters inside Containers and including them in different network environments. The given section starts with the explanation of the two network modes—`venet0` and `veth`–in which any Container can operate and then shows you the way to:

- Create new virtual network adapters inside your Containers and delete existing ones
- Configure the parameters of an existing virtual network adapter (e.g. assign an IP address to it)
- Join Container virtual network adapters to Virtual Networks on the Hardware Node, thus, connecting them to external networks (either LANs or VLANs)

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

Container Networking Modes

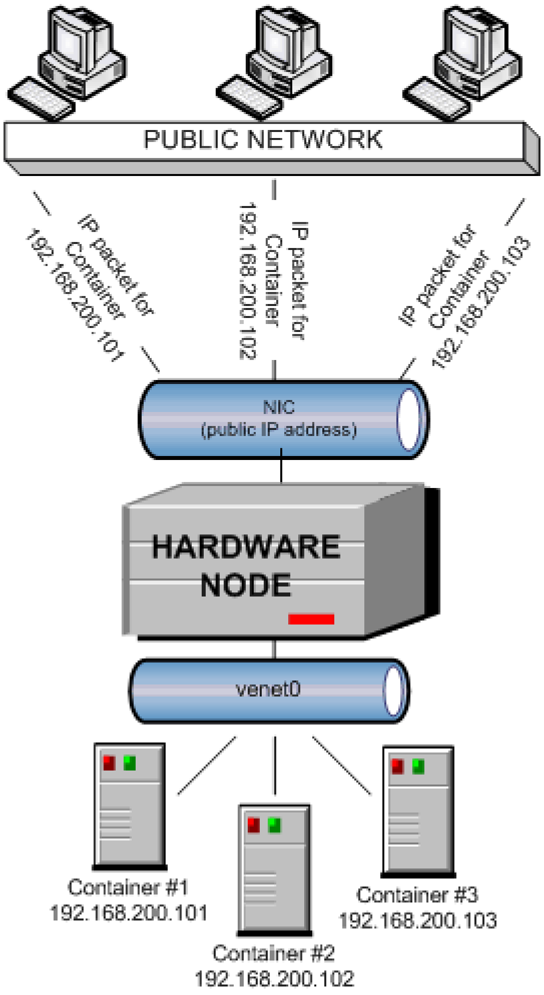
In Parallels Virtuozzo Containers, any Container can operate in one of the two operating modes:

- *venet0*
- *veth*

Detailed information on these operating modes is provided in the following subsections.

venet0 Mode

By default, all newly created Containers on the Node start operating in the `venet0` mode, which means that they are connected among themselves and with the Node using a virtual network adapter called `venet0`. The picture below provides an example of the network structure when all Containers (`Container #1`, `Container #2`, `Container #3`) are functioning in the `venet0` mode.



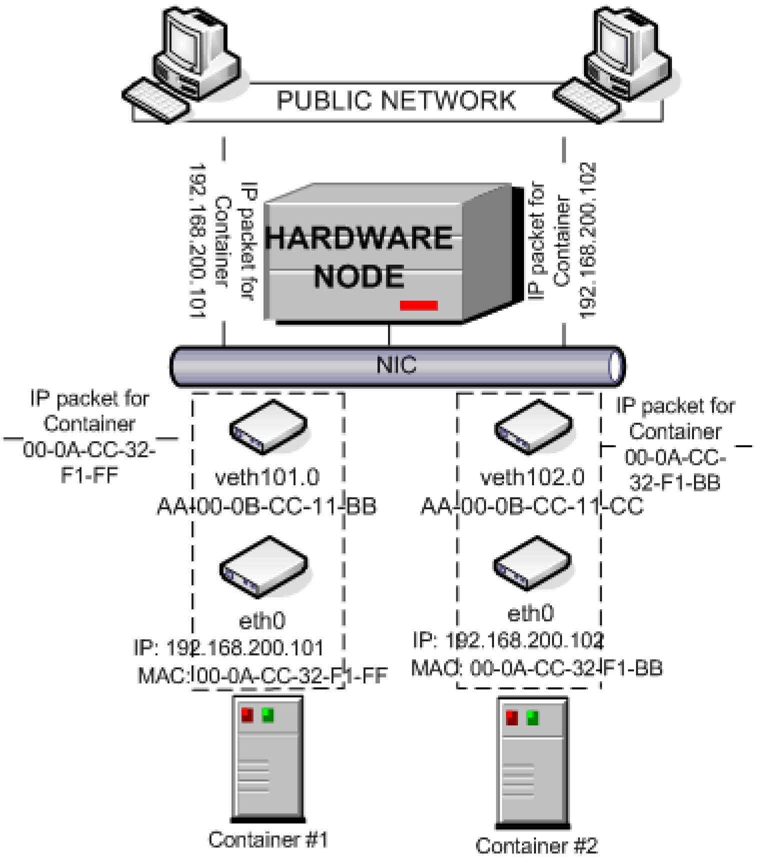
All Containers on the Node use the venet0 virtual adapter as the default gateway to send and receive data to/from other networks (shown as the *PUBLIC NETWORK* in the picture above). The procedure of handling incoming and outgoing IP packets can be described as follows:

- All IP packets from Containers operating in the venet0 mode come to this adapter and are redirected through a public IP address of the Node to the corresponding server on the public network.
- All IP packets coming from external networks and destined for Container IP addresses reach the public IP address of the Node first and, afterwards, are sent through venet0 to the IP addresses of the corresponding Containers.

The venet0 adapter is also used to exchange the traffic among all the Containers hosted on the given Node. All the network traffic of a Container is isolated from that of the other Containers, i.e. all Containers are protected from each other in the way that makes traffic snooping impossible.

veth Mode

You can also create veth virtual adapters inside your Containers and make the Containers operate in the veth mode. The following figure represents an example of the network structure where all Containers (Container#1 and Container#2) are operating in the veth mode:



In the `veth` mode, a separate `veth` virtual adapter is created for each Container on the Node. You are allowed to create several `veth` adapters for a Container. Any `veth` virtual adapter consists of two interfaces:

- An Ethernet interface inside the Container. This interface represents a counterpart of a physical network adapter installed on a standalone server. As any other physical adapter, it has a MAC address (e.g., `00-0A-CC-32-F1-FF` and `00-0A-CC-32-F1-BB`), can be assigned one or more IP addresses (e.g., `192.168.200.101` and `192.168.200.102`) and included in different network environments, etc. Refer to the **Configuring veth Adapter Parameters** section for detailed information on configuring Ethernet interfaces inside Containers.
- An Ethernet interface on the Node. This interface is responsible for the adapter operation in the Node context and mostly used to maintain the interaction and communication between the Node and the Ethernet interface inside the Container. Each Ethernet interface on the Node should be assigned a MAC address (e.g., `AA-00-0B-CC-11-BB` and `AA-00-0B-CC-11-CC`). Detailed information on how to manage Ethernet interfaces on the Node is provided in the **Configuring veth Adapter Parameters** section.

Both interfaces are closely linked to each other, which means that an IP packet entering one interface will always come out from the other one.

Differences Between venet0 and veth Modes

The `veth` mode demonstrates the following differences as compared to the `venet0` mode:

- Each of the Ethernet interfaces constituting a `veth` virtual adapter has a MAC address assigned to it while `venet0` does not have any. Thanks to this fact:
  - Any Container can see all broadcast and multicast packets received from or sent to the selected network adapter on the Node.
  - Using a `veth` virtual adapter inside a Container allows you to host a DHCP or Samba server inside this Container, etc.
- There is no more need to assign all network settings (IP addresses, subnet mask, gateway, etc.) to a Container from the Host OS. All network parameters can be set from inside the Container.
- `veth` adapters can be bridged among themselves and with other devices. If several `veth` adapters are united into a bridge, this bridge can be used to handle network traffic for the Containers whose `veth` adapters are included in the bridge.
- Due to the fact that `veth` adapters act as full members on the network (rather than 'hidden' beyond `venet0`), they are more prone to security vulnerabilities: traffic sniffing, IP address collisions, etc. Therefore, `veth` adapters are recommended to be used in trusted network environments only.

Creating and Deleting veth Network Adapters

By default, any Container on the Hardware Node starts functioning in the venet0 mode right after its creation. However, at any time you can create additional virtual adapters for your Container and set them to work in the veth mode. This can be done by using the --netif_add option of the vzctl set command.

Let us assume that you wish to create a new virtual adapter with the name of eth1 inside Container 101 and make it function in the veth mode. To this effect, you can execute the following command on the Hardware Node:

```
# vzctl set 101 --netif_add eth1 --save
Saved parameters for Container 101
```

The settings of the newly created virtual adapter are saved as the value of the NETIF parameter in the configuration file of Container 101 (/etc/vz/conf/101.conf). So, you can use the following command to display the parameters assigned to the veth network adapter inside Container 101:

```
# grep NETIF /etc/vz/conf/101.conf
NETIF="ifname=eth1,mac=00:10:41:F0:AA:B6,host_mac=00:18:51:A0:8A:D7"
```

As you can see, the parameters set for the veth virtual network adapter during its creation are the following:

- ifname: the name set for the veth Ethernet interface inside Container 101. You specified this name when creating the Container virtual network adapter. Usually, names of Ethernet interfaces inside Containers are set in the form of eth*Ad_N* where *Ad_N* denotes the index number of the created adapter (e.g. eth0 or eth1); however, you can choose any other name you like and specify it during the virtual adapter creation.
- mac: the MAC address assigned to the veth Ethernet interface inside Container 101.
- host_mac: the MAC address assigned to the veth Ethernet interface on the Hardware Node.

ifname is the only mandatory parameter that should be indicated when creating a Container virtual network adapter. All the other parameters are optional and generated by Parallels Virtuozzo Containers automatically, if not specified.

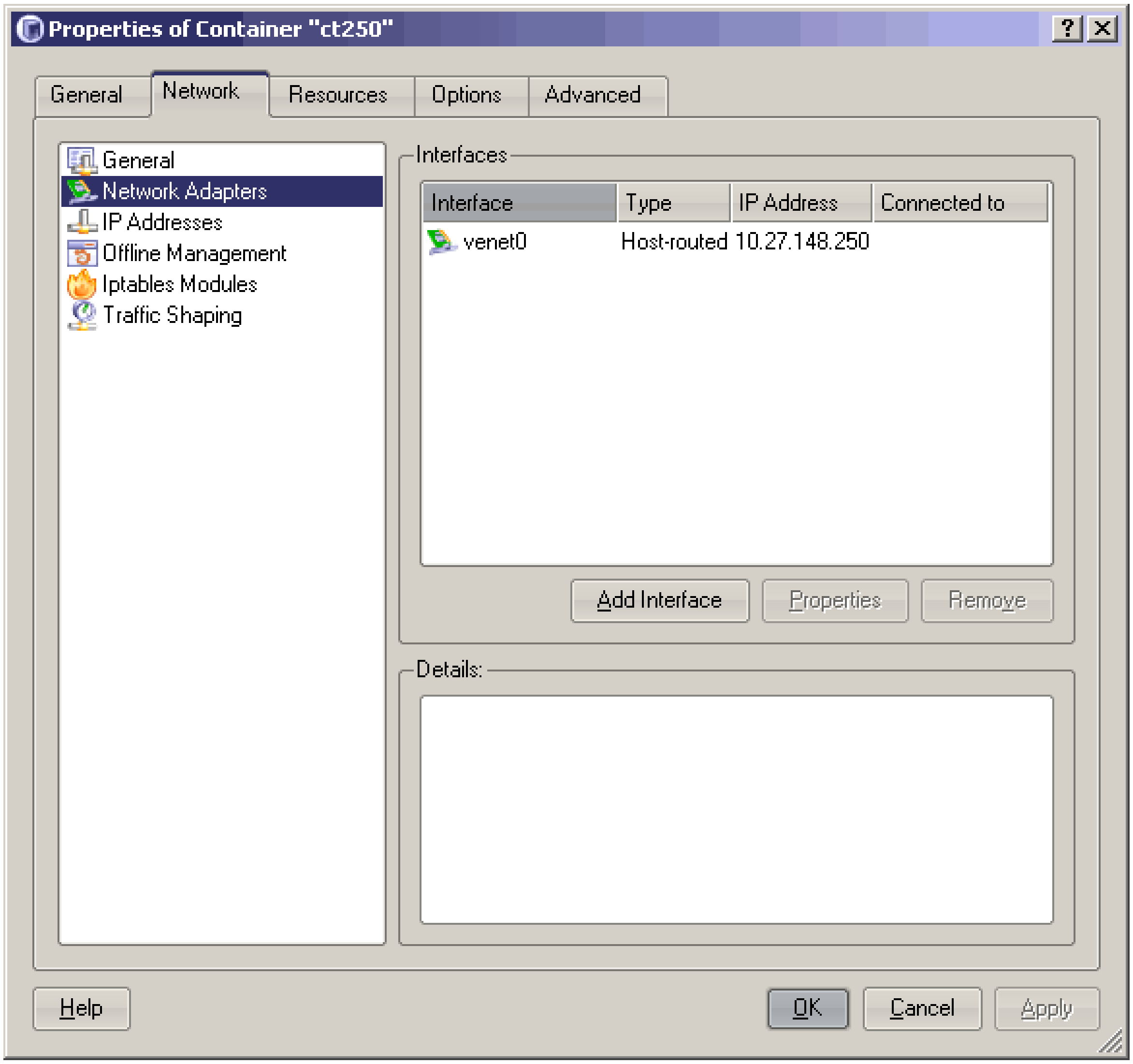
At any time, you can remove the veth virtual network adapter inside Container 101 by executing the following command:

```
# vzctl set 101 --netif_del eth1 --save
Saved parameters for Container 101
# grep NETIF /etc/vz/conf/101.conf
NETIF=""
```

In Parallels Management Console, you can create a new virtual network adapter or delete an existing one by doing the following:

1. Select the **Parallels Virtuozzo Containers** item under the corresponding Hardware Node name.
2. Right-click the Container, and choose **Properties**.

3 Go to the **Network** tab of the displayed window, and select the **Network Adapters** item in the left part of the window.



Interface	Type	IP Address	Connected to
venet0	Host-routed	10.27.148.250	

4 In the right part of the window, use either the **Add Interface** or **Remove** button to create or delete virtual network adapters.

5 Click **OK**.

Configuring veth Adapter Parameters

While functioning in the `veth` mode, each Container virtual network adapter appears as a full participant on the network to which it is connected and needs to have its own identity on this network.

Fist of all, to start functioning on a TCP/IP network, a `veth` virtual adapter should be assigned one or several IP addresses. This can be done as follows:

```
# vzctl set 101 --ifname eth1 --ipadd 192.168.144.123 --save
Saved parameters for Container 101
```

This command will set an IP address of `192.168.144.123` for the `eth1` adapter inside Container 101. You can also assign a network mask to the `eth1` adapter. For example, to set IP address `192.168.144.123` and network mask `255.255.255.0` for the adapter, you can run this command:

```
# vzctl set 101 --ifname eth1 --ipadd 192.168.144.123/24 --save
Saved parameters for Container 101
```

If you want to use the Dynamic Host Configuration Protocol (DHCP) to make the `eth1` adapter of Container 101 automatically receive TCP/IP configuration settings, you can issue the following command instead:

```
# vzctl set 101 --ifname eth1 --dhcp yes --save
Saved parameters for Container 101
```

Any static IP address assigned to the Container virtual network adapter can be removed by executing the following command:

```
# vzctl set 101 --ifname eth1 --ipdel 192.168.144.123 --save
Saved parameters for Container 101
```

You can also delete all IP addresses set for Container 101 at once:

```
# vzctl set 101 --ifname eth1 --ipdel all --save
Saved parameters for Container 101
```

You may also wish to set the following parameters for a Container network adapter:

- The DNS server that the Container virtual adapter will to use:

```
# vzctl set 101 --ifname eth1 --nameserver 192.168.100.111 --save
Saved parameters for Container 101
```

- The gateway to be used for routing the traffic of the Container virtual adapter:

```
# vzctl set 101 --ifname eth1 --gateway 192.168.111.1 --save
Saved parameters for Container 101
```

Detailed information on all options which can be used with the `vzctl set` command to manage Container adapter parameters is given in the *Parallels Virtuozzo Containers 4.7 Reference Guide* and the `vzctl` manual pages.

> **Note:** For detailed information on all parameters that can be configured for each default Container network adapter (i.e. for the adapter operating in the `venet0` mode), see **Configuring Containers** (p. 35).

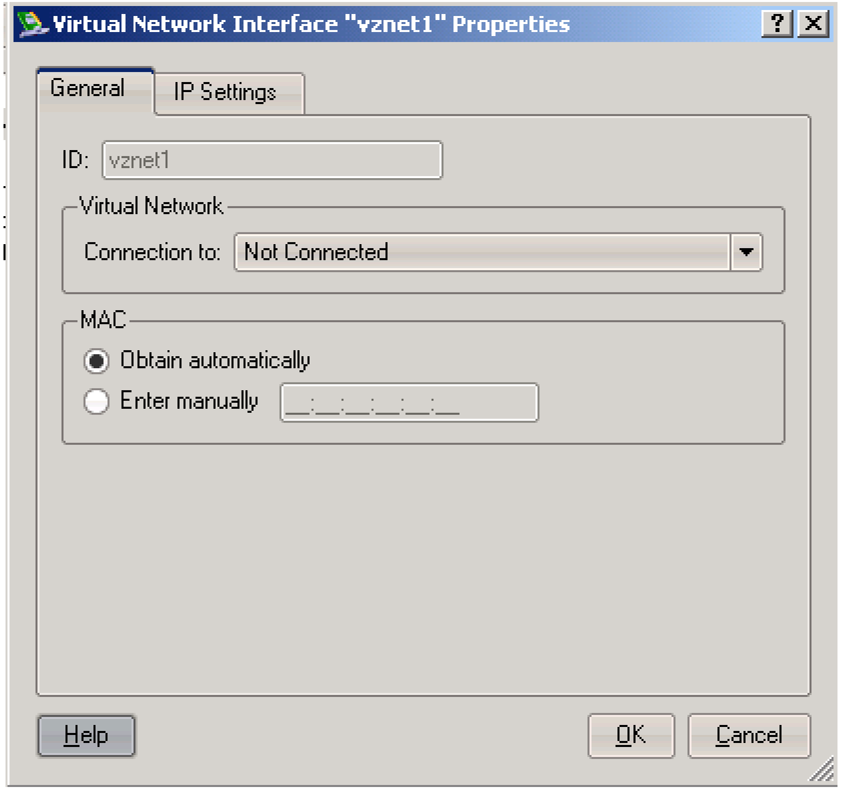
To configure the adapter settings in Parallels Management Console, do the following:

1 Select the **Parallels Virtuozzo Containers** item under the corresponding Hardware Node name.

2 Right-click the Container whose network adapter settings you want to configure, and choose **Properties**.

3 In the displayed window, go to the **Network** tab, and select the **Network Adapters** item in the left part of the window. The list of network adapters existing inside the Container will be shown in the **Interfaces** table in the right part of the window.

4 Select the network adapter the network settings of which you want to configure, and click the **Properties** button at the bottom of the **Interfaces** table.



ID: vznet1

Virtual Network

Connection to: Not Connected

MAC

● Obtain automatically

○ Enter manually __:__:__:__:__:__

5 In this window, you can configure the following adapter parameters:

On the **General** tab of the **Virtual Network Interface Properties** window:

- Change the MAC address assigned to the `veth` Ethernet interface inside the Container by entering the needed MAC address in the **Enter manually** field.
- Connect the Container virtual network adapter to a Virtual Network by clicking the down arrow in the **Connection to** field and selecting the desired Virtual Network on the context menu. Detailed information on how to connect Containers to Virtual Networks is provided in the **Connecting Containers to Virtual Networks** subsection.

On the **IP Settings** tab of the **Virtual Network Interface Properties** window:

- configure the network adapter IP addresses:

a Select the **Obtain IP address via DHCP** radio button to make the adapter automatically receive its IP address and the information on the default gateway through the Dynamic Host Configuration Protocol (DHCP).

b Select the **Get IP address from pool** radio button to make the adapter automatically receive its IP address from the IP addresses pool configured on the Hardware Node.

c Select the **Enter IP addresses manually** radio button and use the **Add** button to manually set one or more IP addresses for the adapter.

- specify the IP address of the default gateway to be used by the network adapter in the **Default gateway address** field (this option is inaccessible if you select the **Obtain IP address via DHCP** radio button).

6 Click **OK** twice.

Connecting Containers to Virtual Networks

With the implementation of `veth` virtual adapters allowing Containers to function as full participants on the network, it has become possible to include Containers in a wide range of network configurations the most common of which are Ethernet networks and VLANs (virtual local area networks). The process of connecting `veth` virtual network adapters to an Ethernet network or to a VLAN is carried out using certain physical and VLAN adapters, respectively, available on the Node and involves completing the following tasks:

1. Creating a Virtual Network that will act as an intermediary between the `veth` adapters and the physical/VLAN adapter.
2. Connecting the `veth` virtual adapters you want to include in an Ethernet network/VLAN to the Virtual Network.
3. Joining the Virtual Network where the `veth` virtual adapters are included to the corresponding physical/VLAN adapter.

After completing these tasks, the Container virtual network adapters will be able to communicate with any computer on the network (either Ethernet or VLAN) where they are included and have no direct access to the computers joined to other networks.

The process of creating new Virtual Networks and joining physical and VLAN adapters to these Virtual Network is described in the **Creating a Virtual Network** (p. 218) and **Connecting an Adapter to a Virtual Network** subsections, respectively. So, in the example below we assume the following:

- The `eth0` physical adapter and the `vznetwork1` Virtual Network exist on the Node.
- The `eth0` physical adapter is connected to the local Ethernet network and to the `vznetwork1` Virtual Network.
- You want to connect Container 101 and Container 102 to the local Ethernet network.

To join Container 101 and 102 to the local Ethernet network behind the `eth0` adapter, you should connect these Containers to the `vznetwork1` Virtual Network. This can be done as follows:

1. Find out the name of the `veth` Ethernet interfaces inside Container 101 and 102:

```
# vzlist -a -o ctid,ifname
  CTID  IFNAME
   101  eth1
   102  eth0
   103  -
```

   The command output shows that the `veth` Ethernet interfaces inside Container 101 and 102 have the names of `eth1` and `eth0`, respectively.

   Note: To add a `veth` adapter to a Virtual Network, you must use the name of its Ethernet interface inside the Container.

2. Join the `veth` adapters to the `vznetwork1` Virtual Network:

- Add the `veth` adapter of Container 101 to the Virtual Network:

```
# vzctl set 101 --ifname eth1 --network vznetwork1 --save
Saved parameters for Container 101
```

- Add the `veth` adapter of Container 102 to the Virtual Network:

```
# vzctl set 102 --ifname eth0 --network vznetwork1 --save
Saved parameters for Container 102
```

After completing these tasks, Container 101 and Container 102 will be able to access any of the servers in the network where the `eth0` physical adapter is connected.

At any time, you can disconnect the `veth` virtual network adapters of Container 101 and 102 from the `vznetwork1` Virtual Network by executing the following commands:

- To disconnect the `veth` adapter of Container 101 from the Virtual Network:

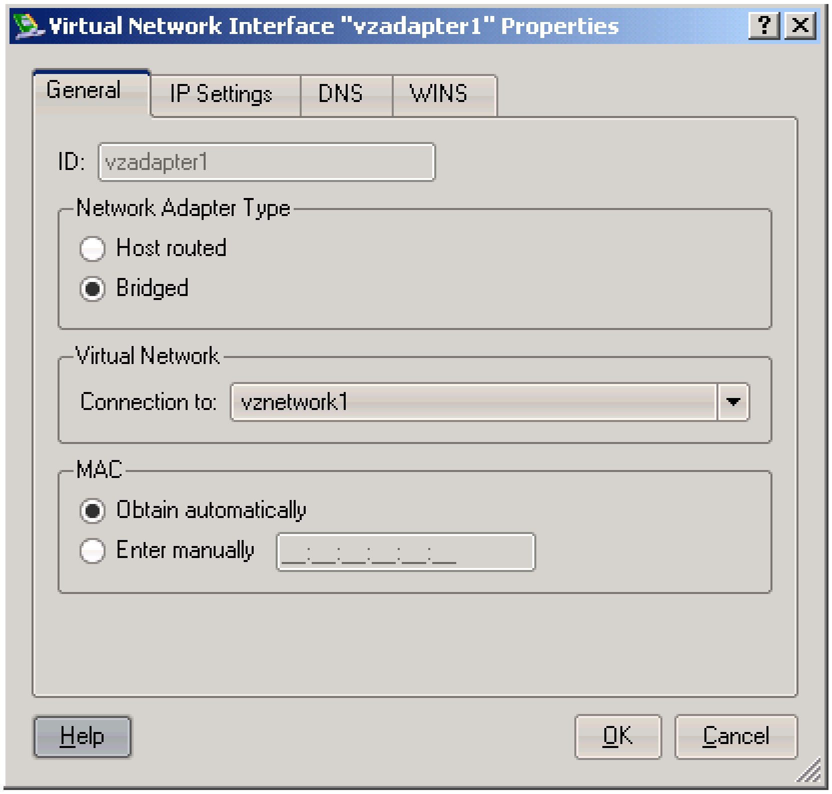
```
# vzctl set 101 --ifname eth1 --network "" --save
Saved parameters for Container 101
```

- To disconnect the `veth` adapter of Container 102 from the Virtual Network:

```
# vzctl set 102 --ifname eth1 --network "" --save
Saved parameters for Container 102
```

In Parallels Management Console, you can join a Container to any Virtual Network on the Hardware Node by doing the following:

1. Click the **Parallels Virtuozzo Containers** item under the corresponding Hardware Node name, right-click the Container you want to join to the Virtual Network, and choose **Properties**.
2. On the **Network** tab of the displayed window, select the **Network Adapters** item.
3. Double-click the Container virtual network adapter to be connected to the Virtual Network.
4. In the **Virtual Network Interface Properties** window, under **Virtual Network**, select the **Connect to** radio button and, on the drop-down menu, choose the needed Virtual Network.



ID: vzadapter1

Network Adapter Type

○ Host routed

● Bridged

Virtual Network

Connection to: vznetwork1

MAC

● Obtain automatically

○ Enter manually __:__:__:__:__:__

5 Click **OK** twice.

To remove a Container virtual network adapter from the Virtual Network where it is currently included, perform **Steps 1-3** described above and, in the **Virtual Network Interface Properties** window, select `Not Connected` on the drop-down menu.

Note: If you are deploying Parallels Virtuozzo Containers in a VMware ESX Server environment, you should perform the following operations to make your Containers operating in the `veth` mode accessible from external servers:
- Make sure that the value of the **Promiscuous Mode** field on the **Security** tab of the **vSwitch Properties** window is set to `Accept`.
- Ensure that the ESX Server adapter always has one and the same MAC address assigned.

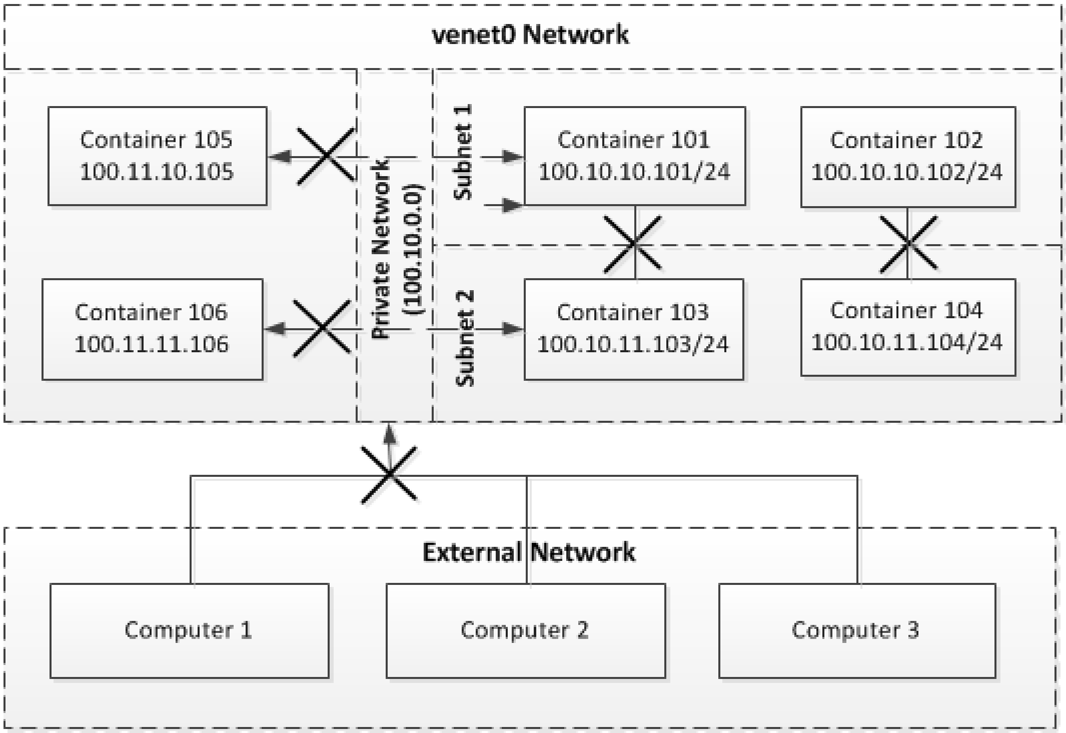
Managing Private Networks and Subnetworks

This section describes how to manage private networks and subnetworks in Parallels Virtuozzo Containers 4.7.

Learning Private Networks

By default, all Containers on the Hardware Node that are working in the `venet0` mode can access each other, irrespective of whether you set a subnet mask for the Containers trying to connect them to different subnets. For example, if Container 101 has the IP address of 100.10.10.101 and Container 102 has the IP address of 100.10.11.102 and you set the subnet mask for these Containers to 255.255.255.0, both Containers will be able to communicate with each other, though they technically belong to different subnets: 100.10.10.0 and 100.10.11.0.

In Parallels Virtuozzo Containers 4.7, you can create the so-called *private networks* for Containers. Within these private networks, you can make subnets and connect Containers to these subnets so that the Containers from one subnet will not be able to access Containers from other subnets, Containers outside the private network, and computers on external networks. The following figure demonstrates a system containing a private network:



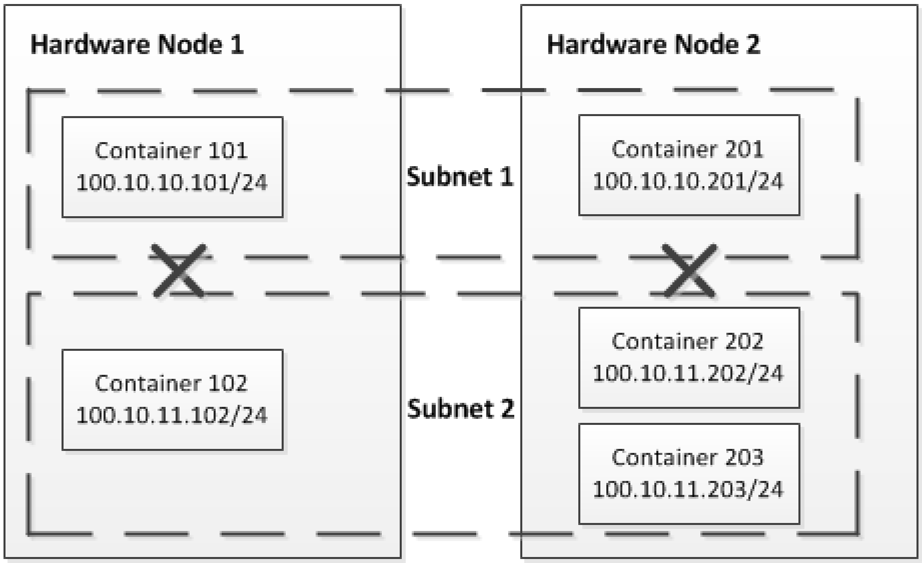
In this example, the network is configured as follows:

- A private network (*Private Network*) is created within the Hardware Node network (*venet0 Network*).
- The private network contains two private subnets: *Subnet 1* and *Subnet 2*.
- Containers 101 and 102 are connected to Subnet 1, and Containers 103 and 104 are joined to Subnet 2.
- Containers 105 and 106 do not belong to the private network.
- The Hardware Node network is connected to an external network (*External Network*) that contains computers *Computer 1*, *Computer 2*, and *Computer 3*.

In this network, Containers 101 and 102 can access each other, but cannot connect to Containers 103, 104, 105, and 106. Containers 103 and 104, in turn, can also access each other, but cannot connect to Containers 101, 102, 105, and 106. None of the Containers in private networks can access computers on the external network.

Network Across Several Nodes

The example above deals with a private network created within one Hardware Node. However, private networks can span Containers on two or more Hardware Nodes. The following figure demonstrates such a network:



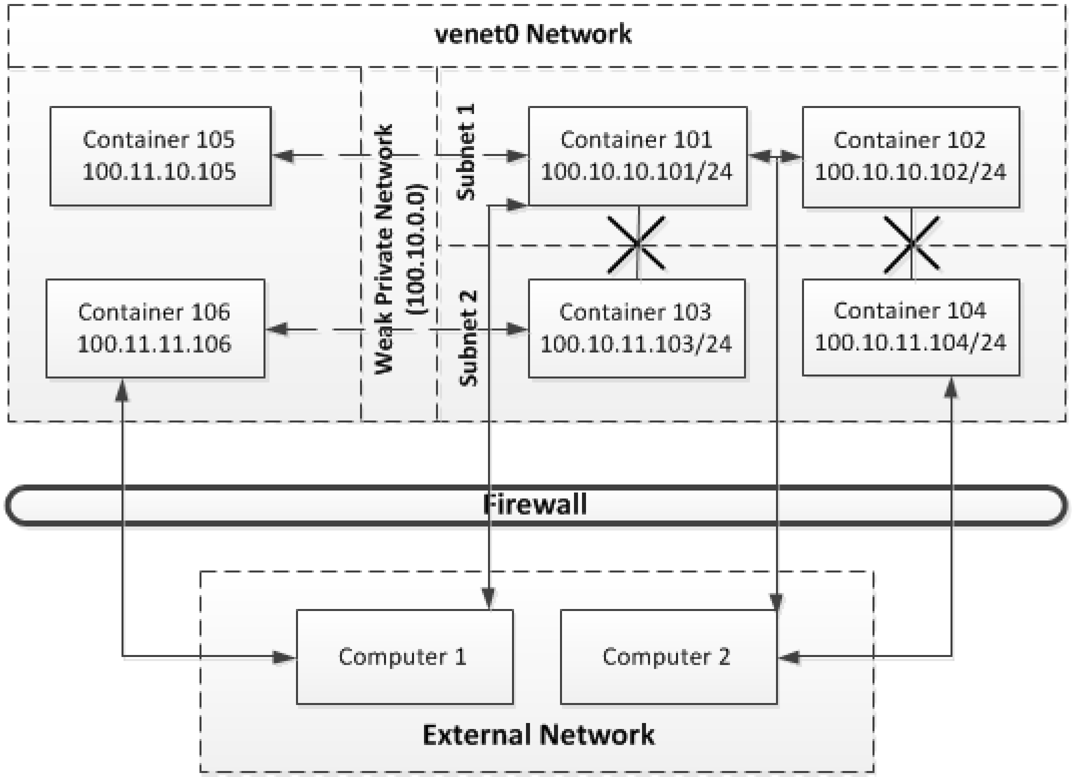
In this figure, the private network also includes two private subnets—Subnet 1 and Subnet 2, but the Containers included in these subnets reside on two Hardware Nodes. Containers 101 and 201 are joined to Subnet 1, and Containers 102, 202, and 203 are joined to Subnet 2. The Containers on Subnet 1 can connect to each other but cannot access the Containers on Subnet 2, and vice versa.

Weak Private Networks

By default, when you create a private network, no Container on this network can access

- Containers that are joined to other subnets
- Containers that are not part of the private network
- computers that are located on external networks

However, you can configure a private network so that its Containers cannot communicate with Containers on other subnets in the private network, but can connect to Containers outside the private network and to computers on external networks. Such private networks are called *weak private networks*. "Weak" in this context means that these networks can be accessed by computers on external networks and are, therefore, more prone to security vulnerabilities and threats. The following picture demonstrates a system with a weak private network:



In this example, the private network on the Hardware Node is divided into two subnets: *Subnet 1* and *Subnet 2*. Containers 101 and 102 are connected to Subnet 1, and Containers 103 and 104 are joined to Subnet 2. Containers 105 and 106 do not belong to the private network. Containers 101 and 102 can access each other, but cannot connect to Container 103 and Container 104. Container 103 and Container 104, in turn, can also access each other, but cannot connect to Container 101 and Container 102.

All four Containers can communicate with Containers 105 and 106 and, as they have public IP addresses assigned, can also access computers on other networks (for example, the computers *Computer 1* and *Computer 2* on the external network *External Network*). To protect the Containers from possible security vulnerabilities and threats (for example, from Internet malware), the firewall is configured on the Hardware Node, blocking unauthorized access to the Containers.

Setting Up Private Networks

The process of setting up a private network on the Hardware Node includes the following steps:

1. Loading the *vzprivnet* module on the Node.
2. Configuring the *vzprivnet* module to meet your demands. When configuring the module, you specify the following parameters:
   a. The range of IP addresses to allocate to the private network.
   b. The number of subnets and hosts in the private network.

Let us assume that you want to create a private network with the following parameters:

- The network includes the IP addresses from 10.10.0.0 trough 10.10.255.255.
- You can create 256 private subnets in the network.
- Each private subnet can have 256 hosts.

To create a network with the specified parameters, do the following:

1. Load the *vzprivnet* module on the Node:

```
# modprobe ip_vzprivnet
```

2. Configure the *vzprivnet* module by specifying the range of IP addresses to allocate to the private network and the maximal number of subnets and hosts allowed in this network. To do this, add the string `10.10.0.0/16/24` to the `ip_vzprivnet` file:

```
# echo "+10.10.0.0/16/24" > /proc/net/ip_vzprivnet
```

   In this command, 10.10.0.0/16 denotes the range of IP addresses to allocate to the private network, and 10.10.0.24 defines the number of subnets and hosts the network can have.

You can create more than one private network on the Hardware Node. To do this, add a line with parameters for each new private network to the `ip_vzprivnet` file, for example:

```
# echo "+10.11.0.0/16/24" > /proc/net/ip_vzprivnet
```

This command creates a new private network that includes the IP addresses from 10.11.0.0 to 10.11.255.255. You can create up to 256 private subnets in this network, and each subnet can have up to 256 hosts.

Creating Weak Private Networks

In a weak private network, any Container on this network can communicate with the other Containers in the same subnet, Containers outside the private network, and computers on external networks. To create a weak private network, you additionally need to append the `*` sign to the line defining the private network settings in the `ip_vzprivnet` file, for example:

```
# echo "+10.20.0.0/16/24*" > /proc/net/ip_vzprivnet
```

This command creates a new weak private network that includes the IP addresses from 10.20.0.0 through 10.20.255.255. The private network can have 256 private subnets, and each subnet can have 256 hosts.

Connecting Containers to Private Subnets

Once you set up a private network, you can connect Containers to different subnets within this network. Assuming that you followed the instructions above, you now have the private network with the IP addresses range from 10.10.0.0 through 10.10.255.255. Let us join Container 101 to subnet 10.10.10.0 and Container 102 to subnet 10.10.11.0. To do this:

1 Open the Parallels Virtuozzo Containers global file for editing and enable the functionality of setting network masks for Containers operating in the `venet0` mode. To do this, set the value of the `USE_VENET_MASK` parameter to `yes`:

```
USE_VENET_MASK="yes"
```

2 Assign IP address 10.10.10.101 and subnet mask 255.255.255.0 to Container 101:

```
# vzctl set 101 --ipadd 10.10.10.101/24 --save
```

3 Assign the IP address of 10.10.11.102 and subnet mask 255.255.255.0 to Container 102:

```
# vzctl set 102 --ipadd 10.10.11.102/24 --save
```

Now Containers 101 and 102 belong to different subnets and cannot access each other. You can check this by logging in to Container 101 and pinging Container 102:

```
# vzctl enter 101
entered into Container 101
-bash-3.2# ping 10.10.11.102
PING 10.10.11.102 (10.10.11.102) 56(84) bytes of data.

--- 10.10.11.102 ping statistics ---
7 packets transmitted, 0 received, 100% packet loss, time 6009ms
```

As you can see, no packets could be transmitted from Container 101 to Container 102.

At any time, you can remove the private network by removing the information about it from the `ip_vzprivnet` file:

```
# echo "-10.10.0.0/16/24" > /proc/net/ip_vzprivnet
```

Once you do this, Container 101 will be able to connect to Container 102 again.

CHAPTER 8

Managing Hardware Nodes

This chapter centers on all those operations you can perform on your Hardware Nodes. You will learn how to manage your Parallels Virtuozzo Containers licenses, to unite your Nodes into a group, to view and configure a number of Parallels Virtuozzo Containers-related parameters.

In This Chapter

Managing Parallels Virtuozzo Containers Licenses

The given section provides information on managing Parallels Virtuozzo Containers licenses. In particular, you will know how to view the current license status, to install a new license on your Hardware Node or to update an existing one, to transfer the license from one Node to another, and so on.

Understanding Licenses

Parallels Virtuozzo Containers license is needed to start using the Parallels Virtuozzo Containers software and management tools (Parallels Management Console, Parallels Virtual Automation, and Parallels Power Panel) on your server. You can install the Parallels Virtuozzo Containers license after or when installing Parallels Virtuozzo Containers on your server. Every Hardware Node must have its own license installed. Licenses are issued by Parallels and define a number of parameters in respect of your Node. The main licensed parameters are listed below:

- The number of CPUs which can be installed on the Hardware Node. Keep in mind that each of the Dual Core and Hyperthreading processors is regarded as one CPU.
- The number of users who can simultaneously use Parallels Management Console and Parallels Virtual Automation to manage the Hardware Node and its Containers.
- The license expiration date. Any license can be time-limited or permanent.

  Parallels Virtuozzo Containers licenses have a start date and, if they are time-limited, can also have an expiration date specified in them. Make sure you set up your system clock correctly; otherwise, the license validation may fail.
- The number of Containers the Hardware Node will be able to host.
- The platform and architecture with which Parallels Virtuozzo Containers is compatible.
- The possibility of managing the Hardware Node by means of Parallels Virtual Automation.

Licenses can be shipped in one of the following forms:

- As an activation code. In this case, you are provided with a special alphanumeric code which you must activate before starting to use Parallels Virtuozzo Containers on the Hardware Node. During the activation, the code is sent to the Parallels Key Authentication (KA) server which verifies the code, generates a special license file, sends it back to the Node, and installs it there.
- As a product key. In this case, you are provided with an alphanumeric key which is installed on the Hardware Node directly, without connecting to the Parallels KA server and exchanging any information with it.

Installing Licenses

Depending on the way you have obtained your Parallels Virtuozzo Containers license, it can be installed on the Hardware Node as follows:

- If you have obtained the license in the form of a product key, you can install it on the Node using the `-p` option of the `vzlicload` command. For example, you can execute the following command to install the `5BVMF2-560MM0-D28DQA-B59NTE-10H4HG` product key on your Hardware Node:

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

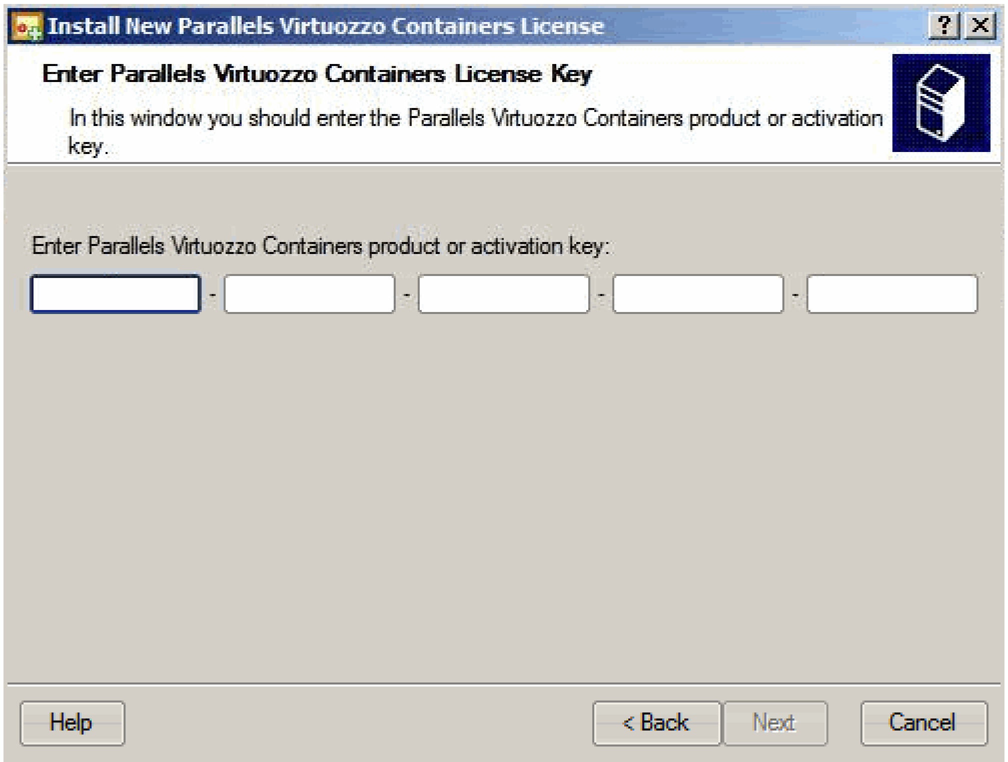
- If you have obtained the license in the form of an activation code, you can install it on the Node using the `-a` option of the `vzlicupdate` command. For example:

```
# vzlicupdate -a 5K4N96-05WRT4-P28A4R-M65W3T-VB4A7C
```

  where `5K4N96-05WRT4-P28A4R-M65W3T-VB4A7C` is the Parallels Virtuozzo Containers 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 Hardware Node from where the activation code has been dispatched, and installs it on this Node. So, before executing the aforementioned command, it is necessary to make sure that the Hardware Node is connected to the Internet.

In Parallels Management Console, you can install a Parallels Virtuozzo Containers license (using both a product key and an activation code) by doing the following:

1. Follow the **Manage License** link at the Hardware Node dashboard.
2. In the **Manage Licenses** window, click the **Install License** button.
3. In the **Choose License Installation Method** window, select the **Enter a new Parallels Virtuozzo Containers license key** radio button, and click **Next**:



Install New Parallels Virtuozzo Containers License

Enter Parallels Virtuozzo Containers License Key

In this window you should enter the Parallels Virtuozzo Containers product or activation key.

Enter Parallels Virtuozzo Containers product or activation key:

4 Enter the product key number or the activation code in the field provided and click **Next**.

5 In the **Review License Details** window, you can view detailed information on the license that will be installed on your Node. Click the **Install** button to initiate the installation process.

If you are activating your Parallels Virtuozzo Containers installation by means of an activation key, you should have an active Internet connection to successfully complete the Parallels Virtuozzo Containers 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 Hardware Node. You can find this ID in the Parallels Management Console warning message displayed after clicking the **Install** button in the **Review License Details** window.

3 Clicking the **Activate License** button.

If you have entered the correct information on the **Parallels Virtuozzo Containers License Activation** page, you will be provided with a link to a license file that you should download to and install on the Hardware Node to start using Parallels Virtuozzo Containers. To install the obtained license file on the Node, do the following:

- Run the `vzlicload` utility with the `-f` option on the Hardware Node where the license file is to be loaded. For example:

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

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

- Using Parallels Management Console:

1 Follow the **Manage License** link at the Hardware Node dashboard.

2 In the **Manage Licenses** window, click the **Install License** button.

3 Select the **Upload the Parallels Virtuozzo Containers license file** radio button in the **Choose License Installation Method** window and click **Next**:

4 In the **Specify Parallels Virtuozzo Containers License File** window, you can do one of the following:

  - Enter the path to the license file in the field provided or use the **Browse** button to specify the location of the license file.
  - Select the **Paste the license text in the area below** radio button and copy the contents of the license file in the field at the bottom of the window.

  When you are ready, click **Next**.

5 In the **Review License Details** window, you can view detailed information on the license that will be installed on your Node. Click the **Install** button to upload the license to the Hardware Node and install it there.

Updating Licenses

The vzlicupdate utility allows you to update the Parallels Virtuozzo Containers license currently installed on the Hardware Node. When executed, the utility tries to connect to the Parallels Key Authentication (KA) server and to retrieve a new license in order to install it on the Node. So, before starting to use this utility, you should make sure that the Hardware Node where you wish to update the license is connected to the Internet. After that, you can issue the following command to update your license:

```
# 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 is to be used by passing the --server option to the utility:

```
# 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 Hardware Node where vzlicupdate has been executed.

Note: In the current version of Parallels Virtuozzo Containers, you can update licenses installed on the Hardware Node with the help of activation code only. If you wish to update a Parallels Virtuozzo Containers product key, contact a Parallels sales representative to learn how you can do it.

To update a license in Parallels Management Console, do the following:

1. Make sure that the workstation where Parallels Management Console is installed and the Hardware Node where you are planning to update the license are connected to the Internet.
2. Follow the **Manage License** link at the Hardware Node dashboard.
3. In the **Manage Licenses** window, click the **Update License** button. Parallels Management Console will try to connect to the Parallels Key Authentication (KA) server, retrieve a new license, and install it on the Node.

Transferring Licenses to Another Node

Sometimes, you may wish to transfer Parallels Virtuozzo Containers licenses from one Hardware Node (*Source Node*) to another (*Destination Node*). For example, this may be the case if the Node where the license is installed starts experiencing problems for some reason or other or requires the hardware upgrade.

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

- If you have activated your Parallels Virtuozzo Containers installation by means of a product key, you can transfer the installed license from the Source Node to the Destination Node as follows:

  a Remove the installed license from the Source Node (e.g. using the `vzlicload -r` *`product_key`* command).

  b Log in to the Destination Node.

  c Install the product key on the Destination Node. Detailed information on how to install Parallels Virtuozzo Containers licenses is provided in **Installing Licenses** (p. 242).

- If you have activated your Parallels Virtuozzo Containers installation by means of an activation code, you should use the `vzlicupdate` utility to move licenses between Hardware Nodes. For example, to transfer the Parallels Virtuozzo Containers license that has been installed on Node 1 using the `9BVMF2-560MN0-F28DQA-059NTE-12H6HG` activation code to Node 2, you should do the following:

  1. Ascertain that Node 1 is shut down or the license is removed from this Node.
  2. Make sure that Node 2 is up and connected to the Internet.
  3. Log in to Node 2 (e.g. via `ssh`).
  4. Execute the following command on Node 2:

```
# vzlicupdate -t -a 9BVMF2-560MN0-F28DQA-O59NTE-12H6HG
```

When executed, `vzlicupdate` sends the `9BVMF2-560MN0-F28DQA-O59NTE-12H6HG` license key to the Parallels KA server, thus informing the server of its intention to transfer the license to a new Hardware Node. The KA server verifies the received license key, generates a new license file, sends it back to Node 2, and installs it there.

To transfer a license from the Source Node to the Destination Node in Management Console, do the following:

a Ascertain that the Source Node is shut down or the license is removed from this Node.

b Make sure that the Destination Node and the computer where Parallels Management Console is installed are connected to the Internet.

c In Parallels Management Console, click the Destination Node name, and follow the **Manage License** link at the Hardware Node dashboard.

d In the **Manage Licenses** window, click the **Install License** button.

e Select the **Transfer a license from another Hardware Node** radio button in the **Choose License Installation Method** window, and click **Next**.

f In the **Enter Product Activation Code** window, enter the activation code and click the **Install** button. Parallels Management Console will connect to the Parallels KA server, inform the server of its intention to transfer the license to a new Hardware Node, get a new license file from the KA server, and install it on the Destination Node.

You can check that the license transferal has completed successfully by means of the `vzlicview` utility. For example, to check that the `U8IK3F-P6QJ8A-O59NTE-42H6HL-D5R07H` product key is now installed on Node 2 (see the example above), issue the following command:

```
# vzlicview
Show installed licenses...
VZSRV
        status="ACTIVE"
        version=4.0
        serial="9BVMF2-560MN0-F28DQA-O59NTE-12H6HG"
        expiration="05/01/2011 23:59:59"
         ...
```

The command output shows that the `9BVMF2-560MN0-F28DQA-O59NTE-12H6HG` license key has been successfully installed on Node 2 and you can start using the Parallels Virtuozzo Containers software on this Node. Detailed information on the `vzlicview` utility and its output is provided in the **Viewing Current License** subsection (p. 247).

Viewing the Current License

The given subsection familiarizes you with the way to view the information on the Parallels Virtuozzo Containers license currently installed on your Hardware Node.

Viewing the License

To view the information on the license and find out its current status, Parallels ships a special `vzlicview` utility. When executed, this utility checks the license currently installed on the Hardware Node and prints the license contents along with its status obtained from the kernel. A sample output of `vzlicview` is given below:

```
# vzlicview
Show installed licenses
VZSRV
        status="ACTIVE"
        version=4.0
        serial="6BWMF2-560MM0-D28DQA-C59NTE-10H6HG"
        expiration="01/01/2011 23:59:59"
        graceperiod=86400 (86400)
        key_number="VZ.00000001.0000"
        cpu_total=64 (1)
        ct_total=8200 (1)
        max_vzmcpmc_users=128
        max_pim_users=260
        platform="Any"
        product="Parallels Virtuozzo Containers"
        vzpp_allowed=1
        backup_mgmt_allowed=1
        workflow_mgmt_allowed=1
        vzagent_allowed=1
        architecture="Any"
```

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

Column Name	Description
`status`	The status of the license currently installed on the Hardware Node. The information on all license statuses is provided in **License Statuses** (p. 249).
`version`	The Parallels Virtuozzo Containers version with which the license is compatible.
`serial`	The Parallels Virtuozzo Containers license serial number.
`expiration date`	The license expiration date, if it is time-limited.
`grace_period`	The period during which Parallels Virtuozzo Containers continues functioning after your license has expired, in minutes.
`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 Hardware Node.
`ct_total`	The total number of Containers which can simultaneously run on the Hardware Node.
`max_vzmc_users`	The number of users able to simultaneously connect to the Node.
`max_vzcc_users`	The number of users able to simultaneously connect to the Node.
`platform`	The operating system with which the license is compatible.
`product`	The product name for which the license has been issued.

`vzpp_allowed`	Indicates whether you can manage Containers residing on the given Hardware Node by means of Parallels Power Panel: • `1`: the 'Parallels Power Panel' functionality is enabled; • `0`: the 'Parallels Power Panel' functionality is disabled.
`backup_mgmt_allowed`	Indicates whether the 'backup' functionality is enabled for the given Hardware Node: • `1`: the 'backup' functionality is enabled; • `0`: the 'backup' functionality is disabled.
`workflow_mgmt_allowed`	Indicates whether the 'Container requesting' functionality is enabled for the given Hardware Node: • `1`: the 'Container requesting' functionality is enabled; • `0`: the 'Container requesting' functionality is disabled.
`vzagent_allowed`	Indicates whether you are allowed to use the Parallels Agent functionality on the given Hardware Node: • `1`: the Parallels Agent functionality is enabled; • `0`: the Parallels Agent functionality is disabled.
`architecture`	The system architecture with which the license is compatible.
`concerto`	If this field is present, the license supports the ability to use the Plesk application in Containers.

In Parallels Management Console, you can check the current status of the license installed on the Hardware Node by doing the following:

1. Follow the **Manage License** link at the Hardware Node dashboard.
2. Choose **Parallels Virtuozzo Containers license** in the top part of the **Manage Licenses** window. The full information about the installed license is displayed in the **License details** table at the bottom of the window.

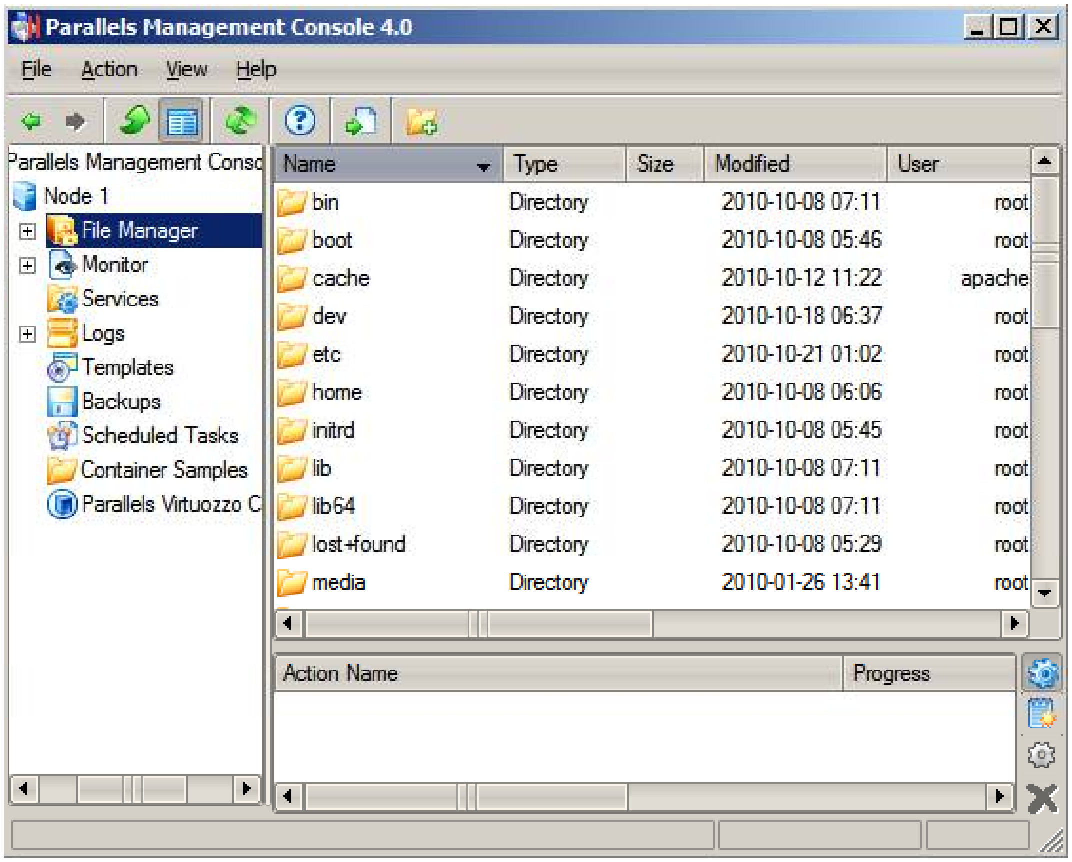
License Statuses

When viewing information on licenses, pay special attention to their status. It can be one of the following:

ACTIVE	The license installed on the Hardware Node is valid and active.
VALID	The license the utility parses is valid and can be installed on the Hardware Node.
EXPIRED	The license has expired.
GRACED	The license has been successfully installed on the Hardware Node; 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 Hardware Node architecture mismatch) or corrupted.

Managing Files

Parallels Management Console provides you with a special file manager allowing you to perform various operations on files and folders located on the Hardware Node. You can access the file manager by clicking the **File Manager** item under the corresponding Hardware Node name. After expanding the **File Manager** item, you will see the list of directories available on the Hardware Node.



Name	Type	Size	Modified	User
bin	Directory		2010-10-08 07:11	root
boot	Directory		2010-10-08 05:46	root
cache	Directory		2010-10-12 11:22	apache
dev	Directory		2010-10-18 06:37	root
etc	Directory		2010-10-21 01:02	root
home	Directory		2010-10-08 06:06	root
initrd	Directory		2010-10-08 05:45	root
lib	Directory		2010-10-08 07:11	root
lib64	Directory		2010-10-08 07:11	root
lost+found	Directory		2010-10-08 05:29	root
media	Directory		2010-01-26 13:41	root

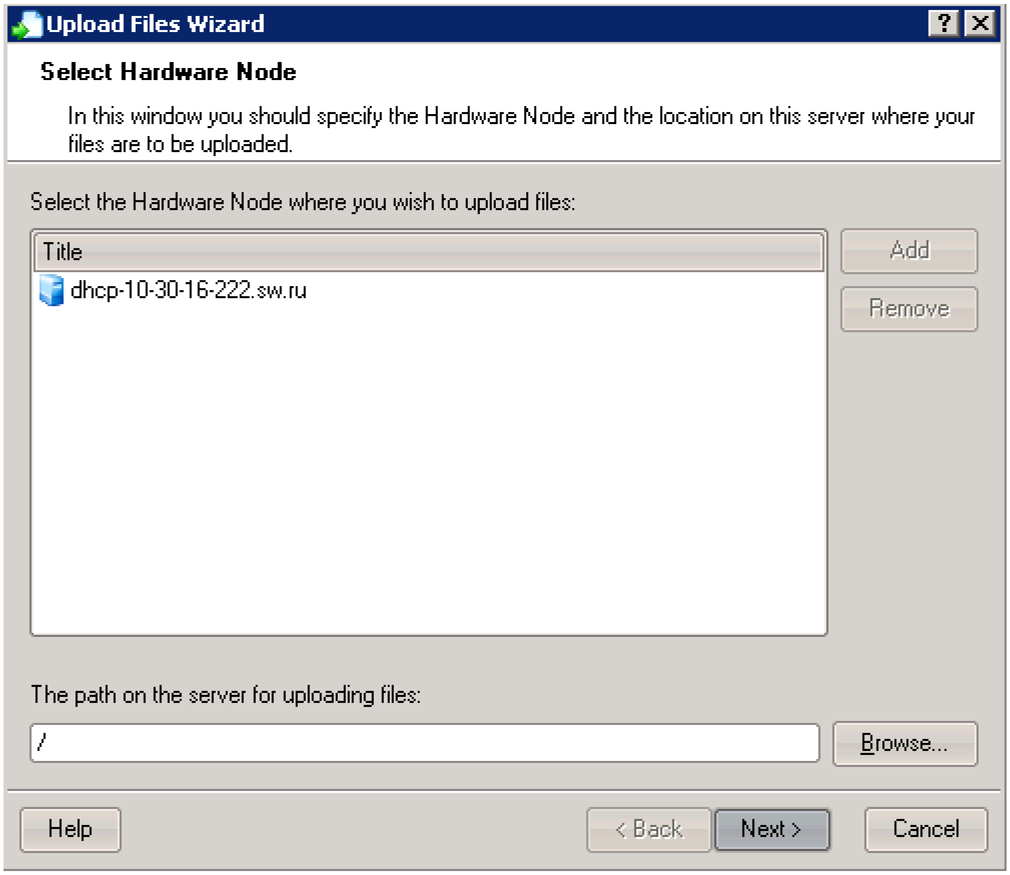
The principles of working with the Hardware Node file manager are standard. You can move through the hierarchy of directories by double-clicking their names or selecting the necessary directories in the left pane. Use the menu items, toolbar buttons, table view, and context menus to perform the following tasks:

- View the contents of simple text files.
- View the principal information about a file/directory available on the Hardware Node.
- Upload any number of files or whole directories from your local computer (the computer where Management Console is installed) to any directory on the Hardware Node.
- Download any number of files from the Hardware Node to your local computer.
- Create new directories on the Hardware Node.
- Copy files to another directory on the Hardware Node.
- Move files to another directory on the Hardware Node.
- Delete files/directories from the Hardware Node.
- Rename files/directories on the Hardware Node.
- Set permissions for Container files.

Parallels Management Console provides a user-intuitive interface for performing all these tasks.

Uploading Files to the Hardware Node

In Parallels Management Console, you can upload any number of files or whole directories from the local computer (the computer where Parallels Management Console is installed) to any directory on the Hardware Node. Under the corresponding Hardware Node name, right-click the **File Manager** item, and choose **Tasks** > **Upload Local File(s)**. The **Upload Files** wizard opens.



Upload Files Wizard

Select Hardware Node

In this window you should specify the Hardware Node and the location on this server where your files are to be uploaded.

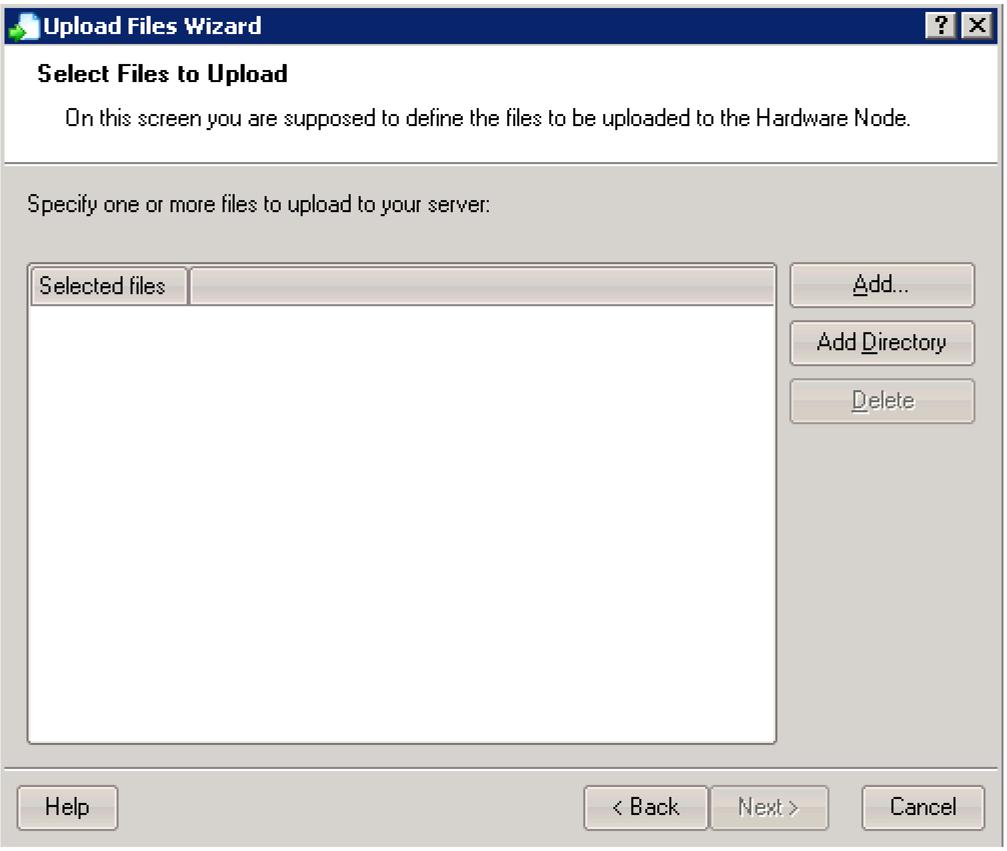
Select the Hardware Node where you wish to upload files:

Title
dhcp-10-30-16-222.sw.ru

The path on the server for uploading files: /

It is a four-step wizard. In the first step of the wizard, define the Hardware Node and the path on this Node where the files will be uploaded. Click the **Add** button to open the **Select Hardware Node(s)** window, and select the Hardware Node you want to add to the upload list. Repeat this sequence for every Hardware Node where you want to upload files. and then click **OK**. After that, enter the path where the files are to be uploaded or browse for this path on the remote Node. Click **Next** when you are finished.

In the second step of the wizard, specify the local files to upload to the Hardware Nodes that you specified in the previous step.



Upload Files Wizard

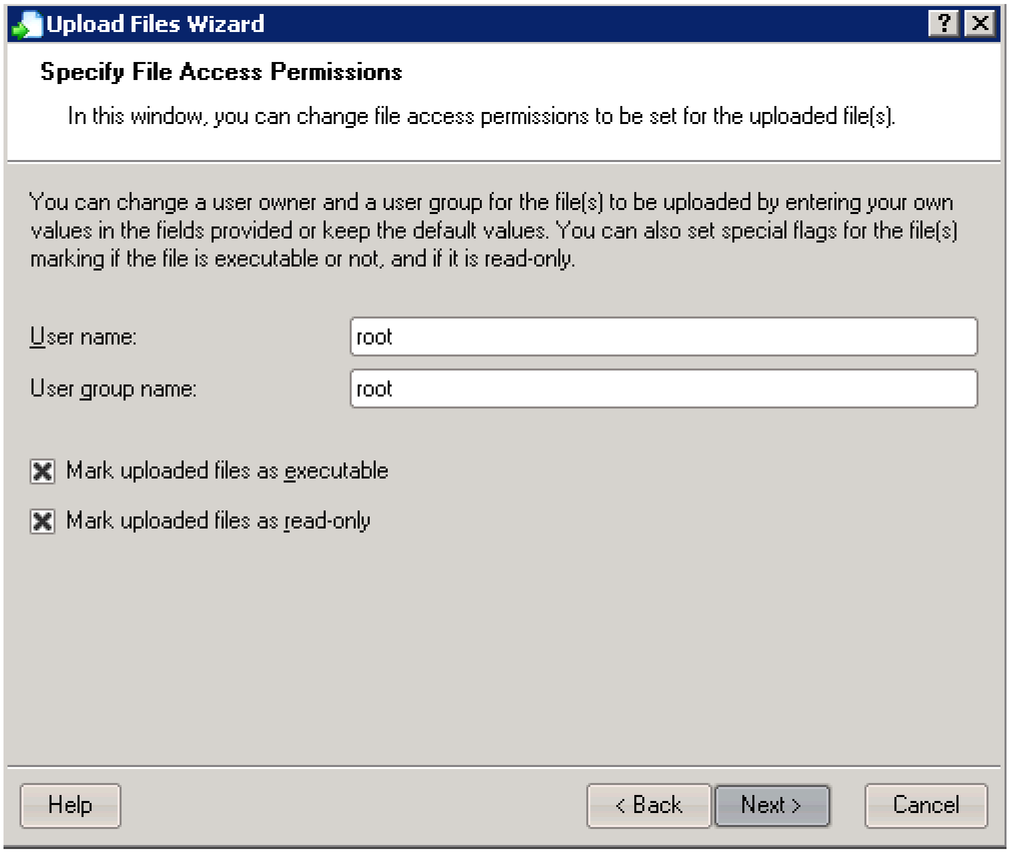
Select Files to Upload

On this screen you are supposed to define the files to be uploaded to the Hardware Node.

Specify one or more files to upload to your server:

Click the **Add** button, and select a file or a group of files from a single directory for uploading. You can also upload the whole directory by clicking the **Add Directory** button. If you need to upload files from various local directories, click the **Add** button the required number of times. After you have added all the files and directories to be uploaded, click **Next**.

Next, specify file access permissions.



Specify File Access Permissions

In this window, you can change file access permissions to be set for the uploaded file(s).

You can change a user owner and a user group for the file(s) to be uploaded by entering your own values in the fields provided or keep the default values. You can also set special flags for the file(s) marking if the file is executable or not, and if it is read-only.

User name: root

User group name: root

☒ Mark uploaded files as executable

☒ Mark uploaded files as read-only

Each file in any Unix system must have a user owner and a user group. The default values are `root` in both cases. You may specify your own values in the fields provided. A file has also special flags marking if the file is executable or not, and if it is read-only. Depending on your choice, the files may be uploaded with any values of these attributes. Review the settings, make the necessary corrections, and click **Next**.

The next window lets you review all the information provided by you in the previous steps of the wizard. Make sure the settings are correct. To change the settings, click the **Back** button and make the necessary corrections. After you click **Next**, the uploading process begins. The operation progress is graphically displayed in the window of the **Upload Files Wizard**. You can see how each of the selected files is being consecutively uploaded to the Hardware Node. Please wait for the operation to finish.

After the uploading process has finished, you will get informed of the results of the operation. The table in the displayed window lets you view the results regarding every file uploaded to the Node. Click **Finish** to exit the wizard.

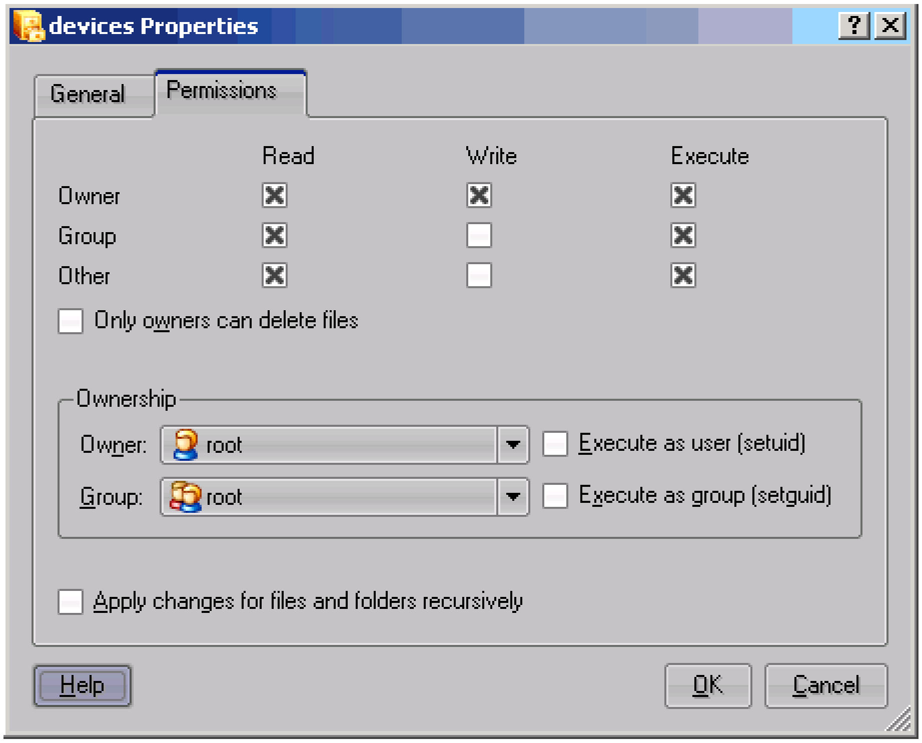
Downloading Files to the Local Computer

Parallels Management Console allows you to download any file or directory located on the Hardware Node to the computer where Parallels Management Console is installed. To do this:

1. Expand the **File Manager** item under the corresponding Hardware Node name.
2. Select the file or directory to download to your local computer (you can use CTRL+Click to select or deselect the file/directory, SHIFT+Click to select a range of files/directories, CTRL+A to select all files/directories).
3. Right-click it, and choose **Tasks** > **Copy To Local Computer**.
4. In the displayed window, specify the directory where to download the selected file/directory.
5. Click **OK**.

Setting Permissions for Files on the Node

Parallels Management Console allows you to view and change the properties files and directories on the Hardware Node. Under the corresponding Hardware Node name, expand the **File Manager** item, right-click the file/directory whose properties you want to display or configure, and choose **Properties**. The **Properties** window opens.



	Read	Write	Execute
Owner	☒	☒	☒
Group	☒	☐	☒
Other	☒	☐	☒

☐ Only owners can delete files

Ownership

Owner: root ☐ Execute as user (setuid)

Group: root ☐ Execute as group (setguid)

☐ Apply changes for files and folders recursively

The information is presented on two tabs:

- **General**: This tab contains only one editable field (**Name**) where you can rename the current file or directory. You can also view the type, location, size, and the last modification date of the file or directory.
- **Permissions**: This tab allows you to set the owner and the group for the corresponding file/directory and its standard Unix properties.

  If you are working with a directory, there are two other options on the tab. They are described in the table below:

Option	Description
Only owners can delete files	This option is used to override the **Write** permission when it is given to **Group** or **Other**. If this is the case, selecting this check box will allow the **Group** and **Other** members only to write the files in the corresponding directory, but not to delete them.
Apply changes for files and folders recursively	If you select this check box, the changes in ownership and permissions that you have made for the current directory, will be recursively applied to all its subdirectories and files.

Updating the Parallels Virtuozzo Containers Software

Parallels Virtuozzo Containers is constantly developing: there appear new versions of the Parallels Virtuozzo Containers core and of existing utilities, OS and application templates are perfected, new templates and utilities are also added from time to time. Thus, Parallels Virtuozzo Containers as a single product may often be repackaged to include the latest changes in any of its parts. As these changes grow in number, new Parallels Virtuozzo Containers versions with incremented major and/or minor numbers are released.

You can use the `vzup2date` utility to update your Parallels Virtuozzo Containers software.

Using these tools, you can connect to the Parallels Virtuozzo Containers update server and update the following components on the Hardware Node:

- Kernel.
- Linux packages copyrighted by third parties (by the OS vendor, for example) but built by Parallels for compatibility with Parallels Virtuozzo Containers. Such packages are usually rebuilt by Parallels and put on the update server after a security or other important hotfix is issued by the third party.
- Parallels Virtuozzo Containers packages copyrighted and built by Parallels, Inc.
- Parallels templates installed on the Hardware Node.

Updating Parallels Virtuozzo Containers With vzup2date

The `vzup2date` utility is intended to relieve Parallels Virtuozzo Containers administrators of the necessity to manually update existing Parallels Virtuozzo Containers installations. It provides a single information channel for learning if updated Parallels Virtuozzo Containers versions are available. In other words, a regular launching of this utility helps ensure that you always have the latest Parallels Virtuozzo Containers software available.

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

- Graphical mode representing the **Parallels Virtuozzo Containers Update** wizard allowing you to update either the Parallels Virtuozzo Containers system files or the Parallels 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 inclusive possibilities for the Parallels Virtuozzo Containers updates management (e.g. the ability to use special filters while selecting updates for your system).

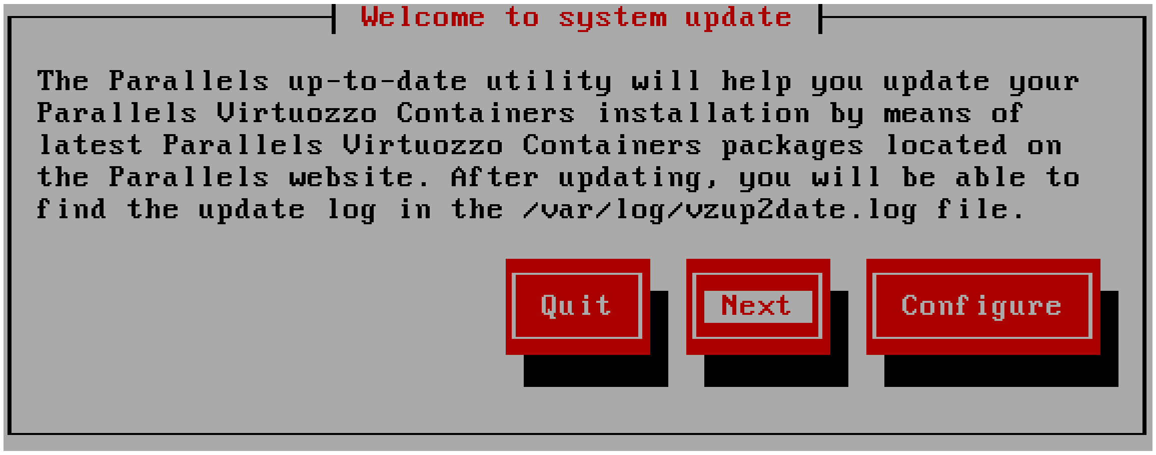
Both modes are described in the following subsections in detail.

Updating in Graphical Mode

In the graphical mode, the `vzup2date` utility can be launched in three submodes. If invoked without any parameters or with the `-s` switch, it checks, downloads, and installs Parallels Virtuozzo Containers system files—that is, the newest versions of the Parallels Virtuozzo Containers core and utilities. On the other hand, the `-t` and `-z` switches provided when running the utility tells it to perform the same operations for OS and application standard and EZ templates, respectively. There is no single interface for checking Parallels Virtuozzo Containers system files and templates at once. You should consecutively call the `vzup2date` utility with and without the `-t` and `-z` switches, if you want to check for both 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.

The `vzup2date` utility is implemented as a wizard, the first few steps of which are common for all three modes. After you launch the utility from the command line, you will be presented with a greeting screen like this:

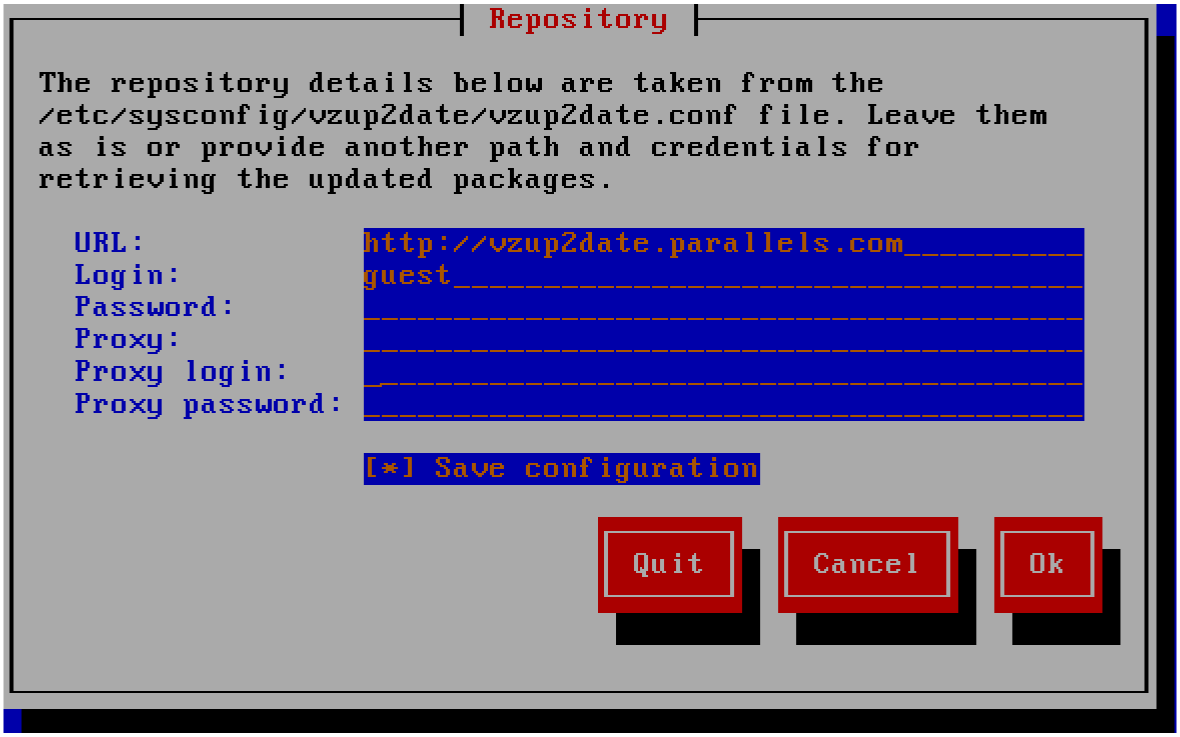


Welcome to system update

The Parallels up-to-date utility will help you update your Parallels Virtuozzo Containers installation by means of latest Parallels Virtuozzo Containers packages located on the Parallels website. After updating, you will be able to find the update log in the /var/log/vzup2date.log file.

In this window, do one of the following:

- Click the **Next** button to connect to the Parallels repository storing updated Parallels Virtuozzo Containers packages and templates and check for available updates.
- Click the **Configure** button to display the current settings used to connect to the Parallels repository and to configure it, if necessary.



Repository

The repository details below are taken from the /etc/sysconfig/vzup2date/vzup2date.conf file. Leave them as is or provide another path and credentials for retrieving the updated packages.

```
URL:            http://vzup2date.parallels.com_________
Login:          guest__________________________________
Password:       _______________________________________
Proxy:          _______________________________________
Proxy login:    _______________________________________
Proxy password: _______________________________________
```

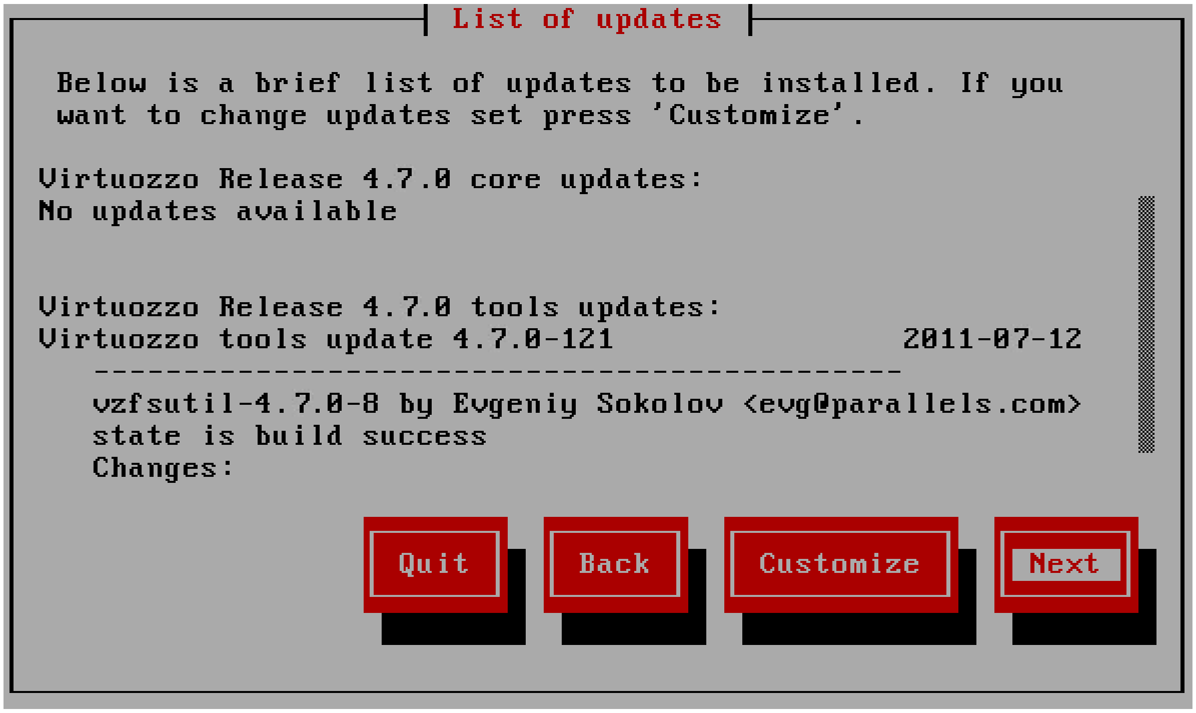
[*] Save configuration

The information in the **Repository** window is taken from the `/etc/sysconfig/vzup2date/vzup2date.conf` file on the Hardware Node. If you want to change this information and save the changes to the configuration file, enter the desired settings into the fields provided, and press **OK**. For example, you may need to configure the settings if you have your own local mirror of the Parallels Virtuozzo Containers official repository. For detailed information on creating local mirrors, see **Creating Local Repositories for vzup2date** (p. 310).

When you press **Next** in the **Welcome** window, the utility will try to connect to the specified repository—either the Parallels official repository or your own one—and if the connection is successful, display the next screen, which will vary depending on whether you are updating Parallels Virtuozzo Containers system files or templates. First, we will describe the mode of updating Parallels Virtuozzo Containers system files and then proceed with updating Parallels Virtuozzo Containers standard and EZ templates.

Updating Parallels Virtuozzo Containers System Files

Once you press **Next** in the **Welcome** window, the utility connects to the repository and checks it for updated system files. If it finds any updates, you will see the list of updates to install on the Hardware Node. This list includes the latest Parallels Virtuozzo Containers updates for the current release.

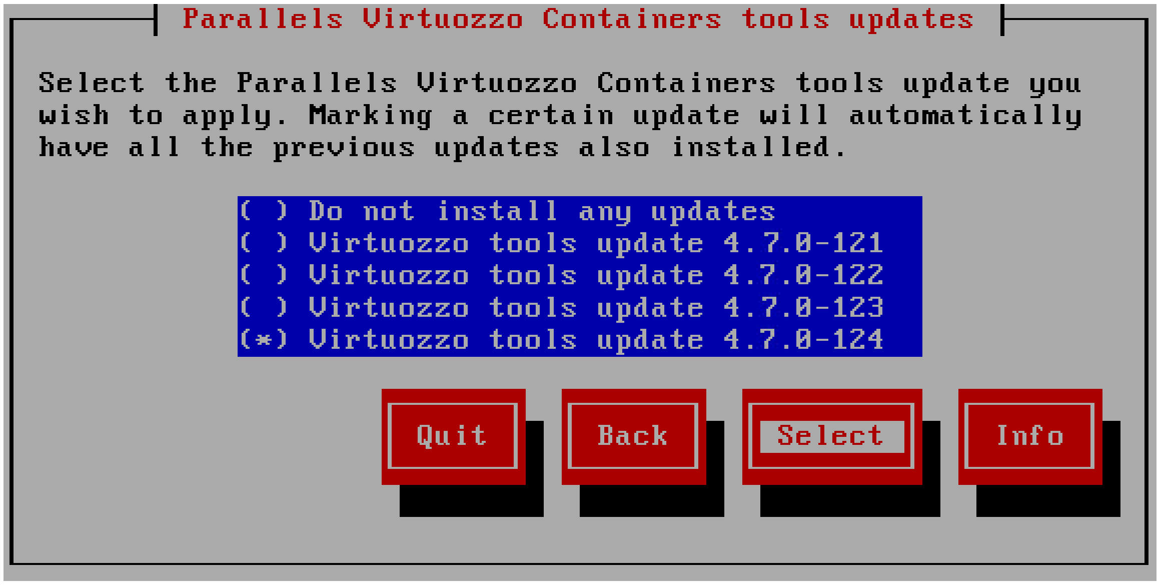


```
List of updates

 Below is a brief list of updates to be installed. If you
 want to change updates set press 'Customize'.

Virtuozzo Release 4.7.0 core updates:
No updates available


Virtuozzo Release 4.7.0 tools updates:
Virtuozzo tools update 4.7.0-121                2011-07-12
   ---------------------------------------------
   vzfsutil-4.7.0-8 by Evgeniy Sokolov <evg@parallels.com>
   state is build success
   Changes:
```

If you want to update to the latest Parallels Virtuozzo Containers core and utilities versions, just press **Next**, and the `vzup2date` utility will download and install them asking your confirmation before each action.

> **Note:** The `vzup2date` utility can see that the selected update includes an updated version of the `vzup2date` utility itself. In this case, you first have to update the utility and then to re-launch it and choose the desired Parallels Virtuozzo Containers system update once again.

If you do not want to install the latest updates for both the Parallels Virtuozzo Containers core and utilities, press **Customize**. You will be able to choose whether to perform the customization on the Parallels Virtuozzo Containers core or on the Parallels Virtuozzo Containers utilities. This step is skipped if updates are currently available either only for the Parallels Virtuozzo Containers core or only for the Parallels Virtuozzo Containers utilities. In the next step, you will be asked to choose the Parallels Virtuozzo Containers core or utilities updates to install. For example:



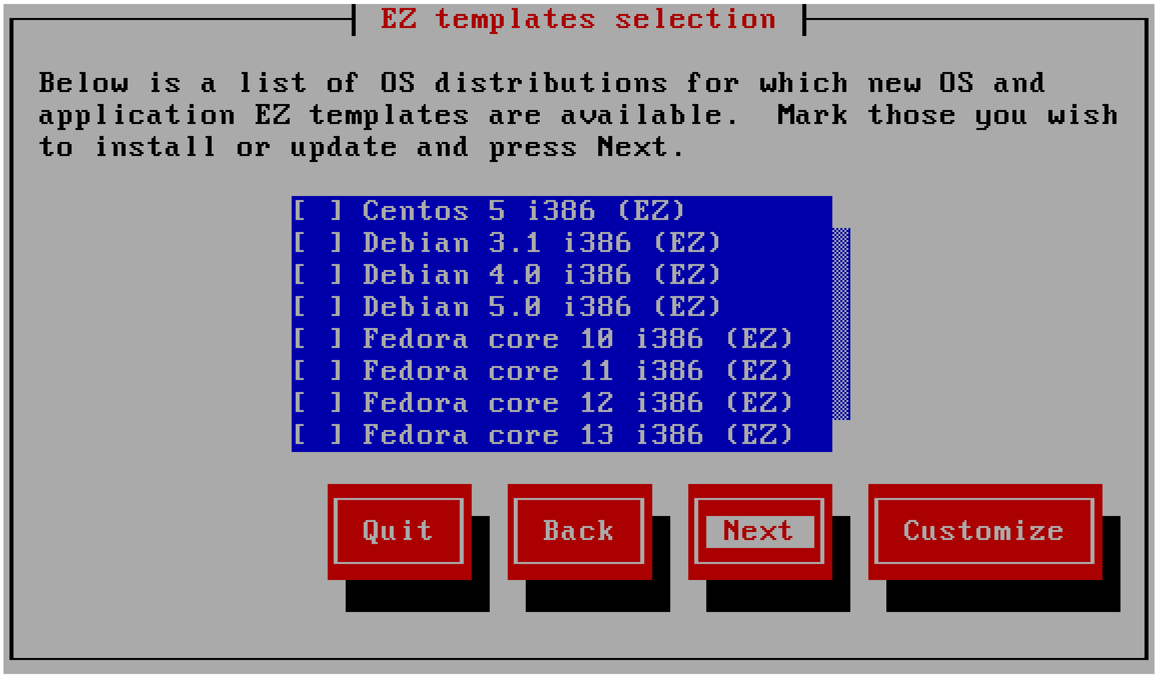
```
Parallels Virtuozzo Containers tools updates

Select the Parallels Virtuozzo Containers tools update you
wish to apply. Marking a certain update will automatically
have all the previous updates also installed.

( ) Do not install any updates
( ) Virtuozzo tools update 4.7.0-121
( ) Virtuozzo tools update 4.7.0-122
( ) Virtuozzo tools update 4.7.0-123
(*) Virtuozzo tools update 4.7.0-124
```

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.

Downloading and installing the necessary updates is straightforward.

Updating EZ Templates

Updating EZ templates consists in updating one or more EZ templates configuration files located in the `/vz/template/<os_name>/<os_version>/<arch>/config` directory on the Node and takes place if you have launched the `vzup2date` utility with the `-z` option. The first few steps of the wizard were described in the **Updating in Graphical Mode** subsection (p. 259). As soon as you press **Next** in the **Welcome** window, the utility will try to connect to the EZ templates repository (either the Parallels default repository or your own one) and, if the connection is successful, display the **EZ templates selection** window listing all EZ templates that have one or more updates available or that are not installed on your Node at all.



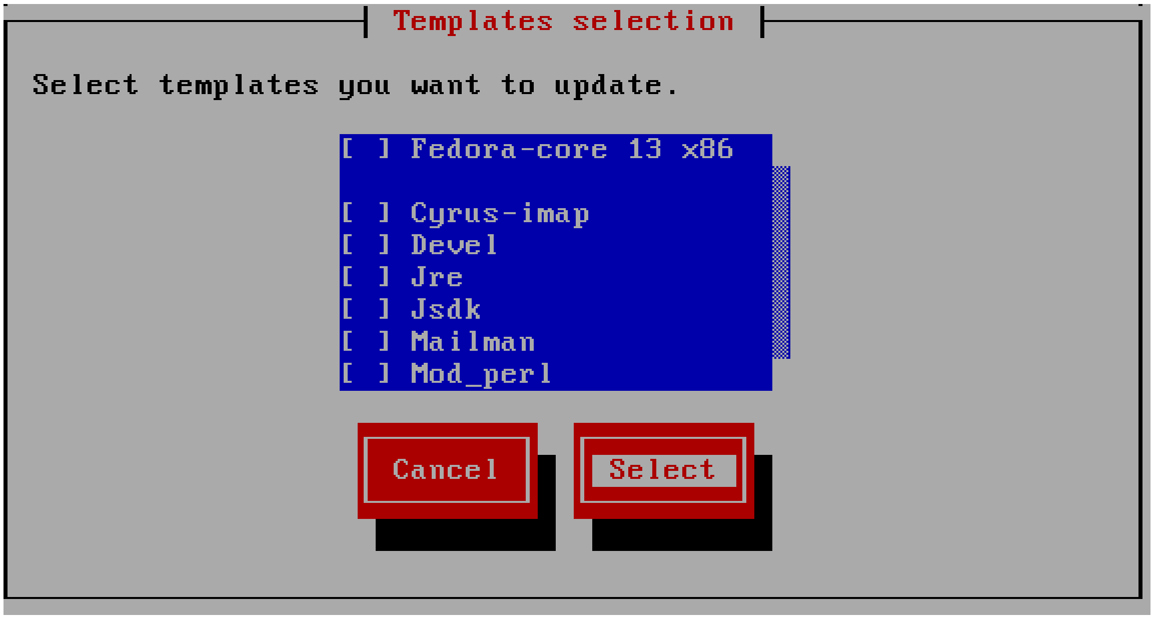
EZ templates selection

Below is a list of OS distributions for which new OS and application EZ templates are available. Mark those you wish to install or update and press Next.

```
[ ] Centos 5 i386 (EZ)
[ ] Debian 3.1 i386 (EZ)
[ ] Debian 4.0 i386 (EZ)
[ ] Debian 5.0 i386 (EZ)
[ ] Fedora core 10 i386 (EZ)
[ ] Fedora core 11 i386 (EZ)
[ ] Fedora core 12 i386 (EZ)
[ ] Fedora core 13 i386 (EZ)
```

In this window, do one of the following:

- If you wish to download and install all available EZ templates/template updates for a certain Linux distribution, select this distribution by placing the cursor beside it and pressing the space bar on your keyboard; then click **Next**.
- If you wish only certain EZ templates of the corresponding Linux distribution to be installed/updated on the Hardware Node, place the cursor beside this distribution and press F2 on your keyboard. You will be presented with the **Templates selection** window where you can select the corresponding EZ templates.



Templates selection

Select templates you want to update.

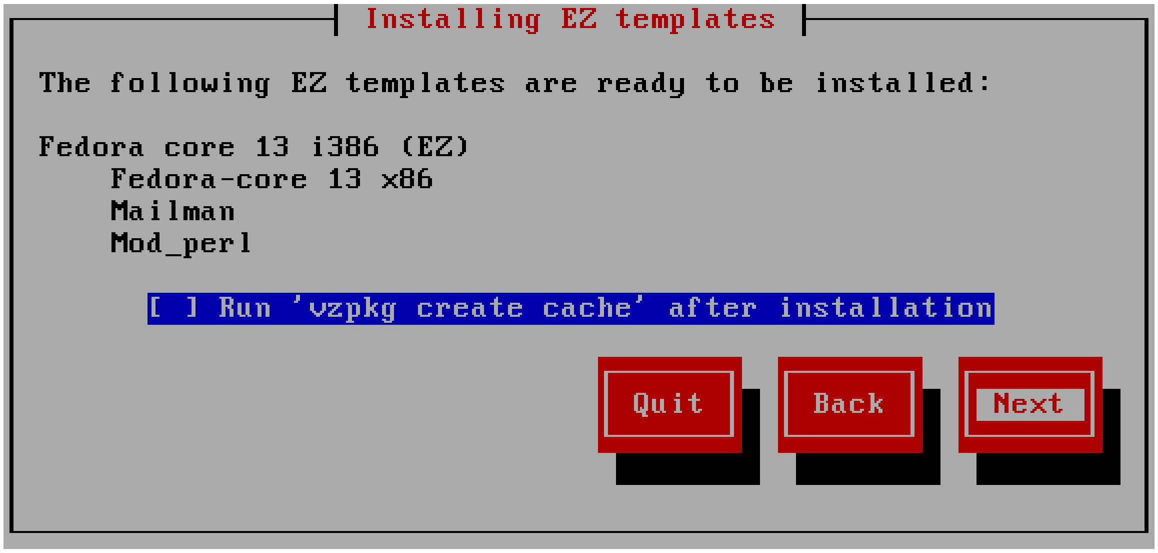
```
[ ] Fedora-core 13 x86

[ ] Cyrus-imap
[ ] Devel
[ ] Jre
[ ] Jsdk
[ ] Mailman
[ ] Mod_perl
```

After choosing the right EZ templates, click the **Select** button to close the displayed window and then click **Next** to proceed with the wizard.

Note: New application EZ templates for a Linux distribution can be installed on the Hardware Node only if the corresponding OS EZ template is already installed on this Node.

In the next step, you can review the EZ templates/template updates you selected in the previous step and scheduled for downloading and installing on your Hardware Node. If you are not satisfied with the chosen templates/template updates, click the **Back** button to return to the previous step and modify the set of templates; otherwise, click **Next** to start downloading the templates/template updates on the Node.

After the EZ templates/templates have been successfully downloaded to the Hardware Node, the **Installing EZ templates** window is displayed.



```
Installing EZ templates

The following EZ templates are ready to be installed:

Fedora core 13 i386 (EZ)
     Fedora-core 13 x86
     Mailman
     Mod_perl

     [ ] Run 'vzpkg create cache' after installation
```

In this window, you can view the templates/template updates ready to be installed on your Node. If you are installing a new OS EZ template/OS EZ template update, you can make use of the **Run 'vzpkg cache' after installation** check box to specify whether to cache the corresponding OS EZ template/template update right after its installation on the Node or to do it at a later time. By default, all OS EZ templates are just installed on the Hardware Node without being cached; however, you can select the provided check box and schedule your OS EZ template/template update for caching. Clicking **Next** starts installing the EZ templates on the Hardware Node. By the time the wizard finishes you should have updated OS and application templates on your system.

Updating in Command-Line Mode

Another way of updating your Parallels Virtuozzo Containers system files and templates is to run the `vzup2date` utility in the command line mode, which can be done by passing the corresponding commands, switches, and options to `vzup2date`. 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 Virtuozzo Containers system files or the Parallels Virtuozzo Containers templates and have the identical syntax. However, the output produced by these commands is different. The messages submode output is less user friendly than the batch submode one 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 (for the batch submode) or the `-m messages` switch (for the messaged submode). Let us assume that you want to update your Parallels Virtuozzo Containers system files by installing the latest Parallels Virtuozzo Containers core in the batch submode. To do this, issue the following command on the Hardware Node:

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

This command will check the Parallels Virtuozzo Containers repository for the latest Parallels Virtuozzo Containers core updates and, in the case of finding any, download and install them on the Hardware Node. However, to be able to update your Parallels Virtuozzo Containers installation, you may need to edit the `/etc/sysconfig/vzup2date/vzup2date.conf` file to specify the repository from where the Parallels Virtuozzo Containers updates are to be downloaded or configure a number of other parameters. Detailed information on the `vzup2date.conf` file is provided in the *Parallels Virtuozzo Containers 4.7 Reference Guide*.

You can also execute the `vzup2date` utility in the batch mode to update Parallels Virtuozzo Containers templates installed on the Hardware Node. For example, you can issue the following command

```
# vzup2date -z -m batch install --all-os
```

to update all EZ OS templates installed on the Node. Detailed information on all options that can be passed to the `vzup2date` utility is given in the *Parallels Virtuozzo Containers 4.7 Reference Guide*.

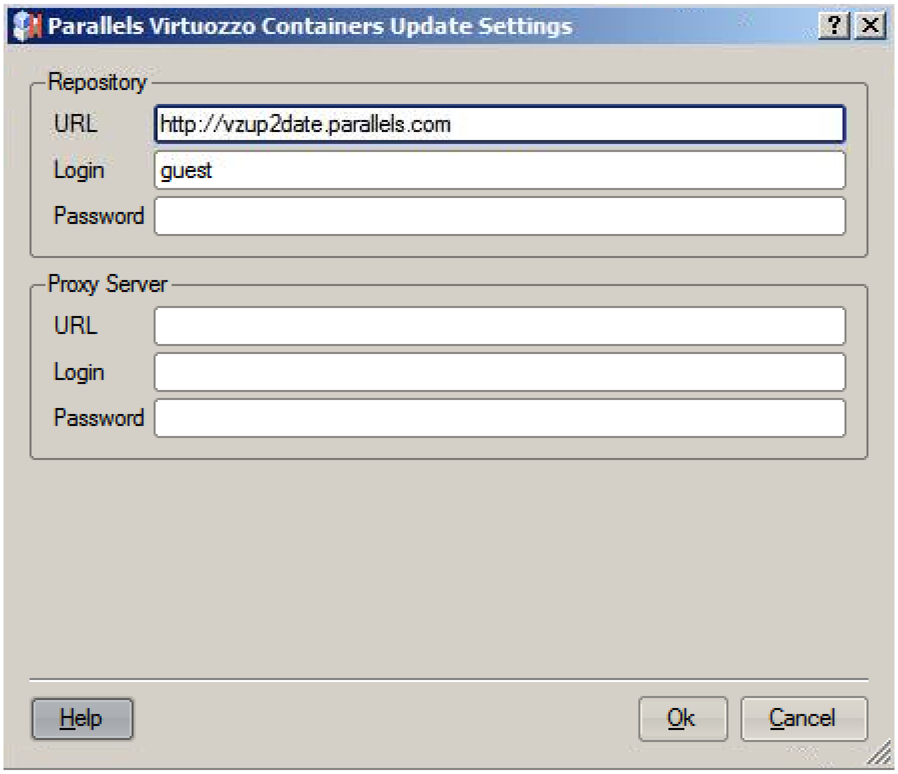
Note: To perform the aforementioned operations in the messages submode, specify the `-m messages` option when running the `vzup2date` utility instead of `-m batch`.

Using Parallels Management Console to Update Parallels Virtuozzo Containers Software

You can also use Parallels Management Console to keep your Parallels Virtuozzo Containers software up to date.

Configuring Parallels Virtuozzo Containers Update Server Settings

Before starting the update procedure in Parallels Management Console, you may wish to check and configure the parameters to be used by the **Parallels Virtuozzo Containers Update** wizard to connect to the Parallels Virtuozzo Containers update server. To view the current settings of the update server, right-click the name of the Hardware Node, and choose **Parallels Virtuozzo Containers Update** > **Configure Parallels Virtuozzo Containers Settings**. The **Parallels Virtuozzo Containers Update Settings** window appears.



Repository URL: http://vzup2date.parallels.com

Login: guest

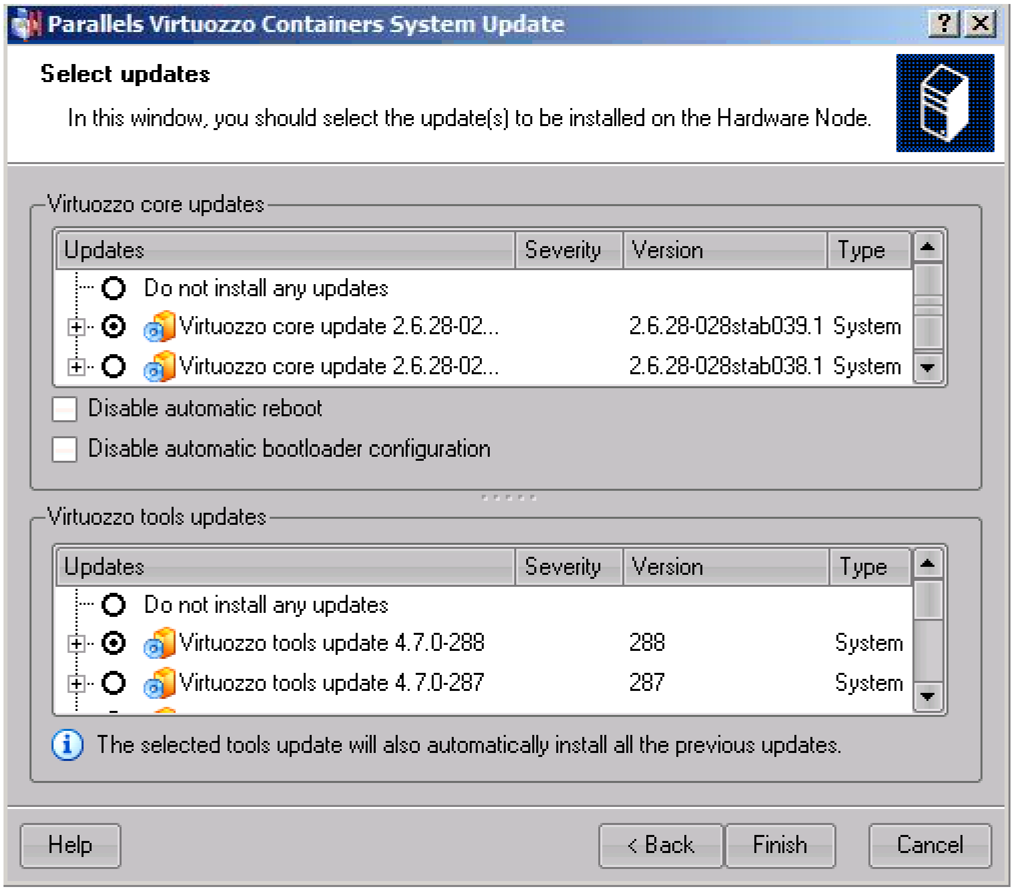
In this window you can view and, if necessary, modify the following settings:

- Under the **Repository** group, you can change a number of parameters related to the update server:
  - the URL (Uniform Resource Locator) to be used to connect to the update server (e.g. `http://vzup2date.parallels.com`)
  - the user name for accessing the update server
  - the password of the user specified in the **Login** group and used for accessing the update server.
- If you use a proxy server to connect to the Internet, you may also need to specify/configure the following settings for your proxy server:
  - the proxy server address in the **URL** field (e.g. `http://192.168.1.20`)
  - the user name used by the proxy server for your authentication in the **Login** field
  - the password of the user specified in the **Login** field and used for your authentication by the proxy server.

Updating Parallels Virtuozzo Containers System Files

In Parallels Management Console, you can use the **Parallels Virtuozzo Containers System Update** wizard to check, download, and install newest versions of Parallels Virtuozzo Containers system files. To invoke the wizard, right-click the name of the Hardware Node to update, and choose **Parallels Virtuozzo Containers Update** > **Check for System Updates** (alternatively, you can follow the **Check for System Updates** link on the Hardware Node dashboard). The wizard will try to connect to the repository storing the updated packages for Parallels Virtuozzo Containers and, if the connection is successful, display the list of available updates in the **Select updates** window.

Note: If the connection to the update server has failed, the **Update Repository Settings** window is displayed allowing you to check and configure the settings to be used for connecting to the repository. Detailed information on how to change the parameters in this window is given in the **Checking Parallels Virtuozzo Containers Update Center Settings** subsection.



Select updates

In this window, you should select the update(s) to be installed on the Hardware Node.

Virtuozzo core updates

Updates	Severity	Version	Type
○ Do not install any updates			
● Virtuozzo core update 2.6.28-02...		2.6.28-028stab039.1	System
○ Virtuozzo core update 2.6.28-02...		2.6.28-028stab038.1	System

□ Disable automatic reboot

□ Disable automatic bootloader configuration

Virtuozzo tools updates

Updates	Severity	Version	Type
○ Do not install any updates			
● Virtuozzo tools update 4.7.0-288		288	System
○ Virtuozzo tools update 4.7.0-287		287	System

The selected tools update will also automatically install all the previous updates.

All updates that can be currently applied to your system are listed in the **Parallels Virtuozzo Containers Core Updates** (storing the latest patches to the Parallels Virtuozzo Containers kernel) and **Parallels Virtuozzo Containers Tools Updates** (storing the latest versions of Parallels Virtuozzo Containers command-line utilities) tables on the **Select Updates** screen. In this window you can do the following:

- If you wish to update to the latest Parallels Virtuozzo Containers core and utilities versions, just click **Finish** on this screen.
- If you wish to install updates of certain Parallels Virtuozzo Containers core or utilities only, select the radio buttons next to these updates and click **Finish**. Please keep in mind that the uppermost update includes the functionality of all the other updates (e.g., update `4.7.0-271` includes all the functionality of update `4.7.0-270`).
- If you wish to view detailed information on an update, expand the plus sign next to this update in the corresponding table.
- If you do not wish to install any updates, select the **Do not install any updates** button.

If you are going to install a Parallels Virtuozzo Containers core update, you can additionally specify what operations are to be performed after the update installation on the Hardware Node:

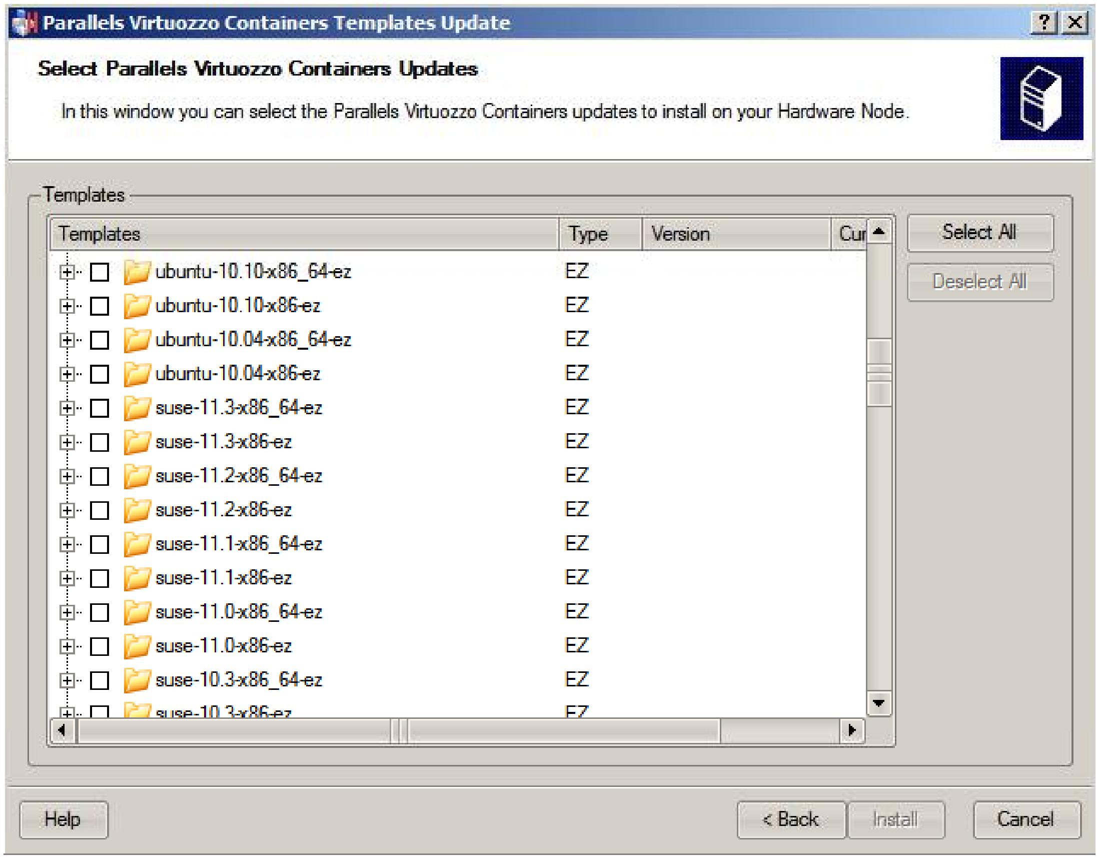
- If you wish your system to be automatically rebooted upon the update installation completion, leave the **Disable automatic reboot** check box cleared. Rebooting the Node is usually required for the changes made to the Parallels Virtuozzo Containers kernel to take effect.
- If you wish the **Parallels Virtuozzo Containers System Update** wizard to automatically reconfigure your system boot loader (either Lilo or Grub) on applying the update, leave the **Disable automatic bootloader configuration** check box cleared; otherwise, select this check box.

When you are ready, click **Finish** to start downloading the selected updates and installing them on the Node.

Updating Templates in Parallels Management Console

Parallels Management Console provides you with the **Templates Update** wizard allowing you to update any of EZ and standard templates installed on your Hardware Node. You can also use this wizard to download new templates to the Hardware Node and install them there. To invoke the **Templates Update** wizard, right-click the **Templates** item under the corresponding Hardware Node name and select **Check for Template Updates** on the context menu. When launched, the wizard tries to connect to the templates repository (either the Parallels default repository or your own one) and, if the connection is successful, display the **Select Updates** window listing those templates that have one or more updates available or that are not installed on your Node at all. For example:

Note: If the connection to the Parallels Virtuozzo Containers update server cannot be established, you will be presented with the **Repository Update Settings** window where you will be asked to provide the correct information to connect to the update server. Detailed information on how to change the parameters in this window is given in the **Checking Parallels Virtuozzo Containers Update Server Settings** subsection.



Parallels Virtuozzo Containers Templates Update

Select Parallels Virtuozzo Containers Updates

In this window you can select the Parallels Virtuozzo Containers updates to install on your Hardware Node.

Templates

Templates	Type	Version	Cur
☐ ubuntu-10.10-x86_64-ez	EZ		
☐ ubuntu-10.10-x86-ez	EZ		
☐ ubuntu-10.04-x86_64-ez	EZ		
☐ ubuntu-10.04-x86-ez	EZ		
☐ suse-11.3-x86_64-ez	EZ		
☐ suse-11.3-x86-ez	EZ		
☐ suse-11.2-x86_64-ez	EZ		
☐ suse-11.2-x86-ez	EZ		
☐ suse-11.1-x86_64-ez	EZ		
☐ suse-11.1-x86-ez	EZ		
☐ suse-11.0-x86_64-ez	EZ		
☐ suse-11.0-x86-ez	EZ		
☐ suse-10.3-x86_64-ez	EZ		
☐ suse-10.3-x86-ez	EZ		

In this window, do one of the following:

- If you wish to download and install all available templates/template updates for a certain Linux distribution, click the **Next** button to go to the next step of the wizard.
- If you wish only certain templates of a Linux distribution to be installed/updated on the Hardware Node, click on the plus sign beside the corresponding Linux distribution to display a list of application templates available for this distribution. You can then get detailed information about a particular template by selecting the corresponding template and viewing its data in the right part of the displayed window. By default, all new templates/template updates are set for downloading to and installing on the Hardware Node. To prevent this or that template from being downloaded/installed, just clear its check box. When you are ready, click **Next**.

Click **Finish** to start installing the selected templates/template updates on the Hardware Node.

CHAPTER 9

Advanced Tasks

This chapter describes those tasks that are intended for advanced system administrators who would like to obtain deeper knowledge about Parallels Virtuozzo Containers capabilities.

In This Chapter

Configuring Capabilities

Capabilities are sets of bits that permit of splitting the privileges typically held by the root user into a larger set of more specific privileges. The POSIX capabilities are defined by a draft IEEE standard (IEEE Std 1003.1e); they are not unique to Linux or Parallels Virtuozzo Containers. When the Linux or Parallels Virtuozzo Containers documentation says “requires root privileges”, in nearly all cases it really means “requires a specific capability”.

This section documents the tasks that can be achieved using per-Container capabilities in Parallels Virtuozzo Containers and all configurable capabilities.

Creating VZFS Symlinks Inside a Container

Normally it is impossible to create a VZFS symlink from a Container. The ability to create VZFS symlinks presents a serious security concern explained further in this subsection. However, there may be a situation when you need such an ability, for example, for testing created templates or creating VZFS mounts.

A VZFS symlink is a symbolic link starting with four slashes. You can see VZFS symlinks in the private area of any Container, as is illustrated below:

```
# ls -l /vz/private/101/root/bin/bash
lrwxr-xr-x   1 root  root  37 Jul  9  2009 \
/vz/private/101/root/bin/bash -> \
////redhat-as4/bash-3.0-19.2/bin/bash
```

VZFS symlinks have no special meaning if the private area is not mounted over VZFS (to the Container root directory). If it is, then instead of a VZFS symlink the users inside the Container will see the file located in the template directory (in this particular case, `/vz/template/redhat-as4/bash-3.0-19.2/bin/bash`) instead of the VZFS symlink.

If you try to create a VZFS symlink inside the Container, you will get an error:

```
[root@ct101 root]# ln -s ////redhat-as4/bash-3.0-19-2/bin/bash .
ln: creating symbolic link `./bash' to \
`////redhat-as4/bash-3.0-19.2/bin/bash': Invalid argument
```

The reason for this restriction is security considerations. If an intruder can correctly guess where the template area (defined by the `TEMPLATE` variable in the global configuration file `/etc/sysconfig/vz`) is located, he/she can access any file on the Node provided the path to the file is guessed correctly. However, in case it is necessary to allow the VZFS symlinks creation inside a Container, it is possible to make use of the `sys_rawio` capability:

```
# vzctl set 101 --capability sys_rawio:on
Unable to set capability on running Container
Saved parameters for Container 101
```

After restarting the Container, you can unpack VZRPMs inside the Container or simply create VZFS symlinks:

```
# ssh root@ct101
root@ct101's password:
Last login: Mon Oct 28 23:25:58 2008 from 10.100.40.18
[root@ct101 root]# rpm2cpio bash-3.0-19.2.i386.vz.rpm | cpio -id
94 blocks
[root@ct101 root]# ls -l bin/bash
-rwxr-xr-x    1 root      root       519964 Oct 29 23:35 bin/bash
[root@ct101 root]# ln -s ////redhat-as4/bash-3.0-19.2/bin/bash .
[root@ct101 root]# ls -l bash
-rwxrwxrwx    1 root      root       519964 Oct 29 23:35 bash
```

As you can see both VZFS symlinks look like regular files for Container users. If you need to unpack and work on symlinks themselves, you have to create a Container that has a directory bind-mounted over a regular file system such as EXT2FS, EXT3FS or ReiserFS.

Remember that assigning this capability to non-trusted Containers can lead to compromising the Node. The session below shows how a malicious Container administrator can get a copy of the Node password database files:

```
[root@ct101 root]# ln -s ////../../etc/passwd .
[root@ct101 root]# ln -s ////../../etc/shadow .
[root@ct101 root]# ls -l
total 3
-rwxrwxrwx    1 root     root         1252 Oct 29 23:56 passwd
-rwxrwxrwx    1 root     root          823 Oct 29 23:56 shadow
```

While there is no easy way to substitute the password files on the Node, a malicious Container administrator could run a dictionary attack against the obtained files.

Available Capabilities for Containers

This section lists all the capabilities that can be set with the `prlctl` command. The capabilities are divided into two tables: the capabilities defined by the POSIX draft standard and Linux-specific capabilities. For each capability, its description is given together with the default value for a Container.

Please note that it is easy to create a non-working Container or compromise your Node security by setting capabilities incorrectly. Do not change any capability for a Container without a full understanding of what this capability can lead to.

Capabilities Defined by POSIX Draft

Name	Description	Default
`chown`	If a process has this capability set on, it can change ownership on the files not belonging to it or belonging to another user. You have to set this capability on to allow the Container root user to change ownership on files and directories inside the Container.	on
`dac_override`	This capability allows to access files even if the permission is set to disable access. Normally leave this on to let the Container root access files even if the permission does not allow it.	on
`dac_read_search`	Overrides restrictions on reading and searching for files and directories. The explanation is almost the same as above with the sole exclusion that this capability does not override executable restrictions.	on
`fowner`	Overrides restrictions on setting the `S_ISUID` and `S_ISGID` bits on a file requiring that the effective user ID and effective group ID of the process shall match the file owner ID.	on
`fsetid`	Used to decide between falling back on the old `suser()` or `fsuser().`	on
`kill`	Allows sending signals to processes owned by other users.	on
`setgid`	Allows group ID manipulation and forged group IDs on socket credentials passing.	on
`setuid`	Allows user ID manipulation and forged user IDs on socket credentials passing.	on

Linux-Specific Capabilities

Name	Description	Default
`setpcap`	Transfer any capability in your permitted set to any process ID; remove any capability in your permitted set from any process ID.	off
`linux_immutable`	Allows the modification of the `S_IMMUTABLE` and `S_APPEND` file attributes. These attributes are implemented only for the EXT2FS and EXT3FS Linux file systems. However, if you bind mount a directory located on the EXT2FS or EXT3FS file system into a Container and revoke this capability, the root user inside the Container will not be able to delete or truncate files with these attributes on.	on
`net_bind_service`	Allows to bind to sockets with numbers below 1024.	on
`net_broadcast`	Allows network broadcasting and multicast access.	on
`net_admin`	Allows the administration of IP firewalls and accounting.	off
`net_raw`	Allows to use the RAW and PACKET sockets.	on
`ipc_lock`	Allows to lock shared memory segments and `mlock/mlockall` calls.	on
`ipc_owner`	Overrides IPC ownership checks.	on
`sys_module`	Insert and remove kernel modules. Be very careful with setting this capability on for a Container; if a user has the permission of inserting kernel modules, this user has essentially full control over the Node.	off
`sys_chroot`	Allows to use `chroot()`.	on
`sys_ptrace`	Allows to trace any process.	on
`sys_pacct`	Allows to configure process accounting.	on
`sys_admin`	In charge of many system administrator tasks such as swapping, administering APM BIOS, and so on. Shall be set to off for Containers.	off
`sys_boot`	This capability currently has no effect on the Container behaviour.	on
`sys_nice`	Allows to raise priority and to set priority for other processes.	on
`sys_resource`	Override resource limits (do not confuse with user beancounters).	on
`sys_time`	Allows to change the system time.	off
`sys_tty_config`	Allows to configure TTY devices.	on
`mknod`	Allows the privileged aspects of `mknod()`.	on
`lease`	Allows to take leases of files.	on

Migrating a Physical Server to a Container

This section provides information on how you can migrate an external physical server to a Container.

Migration Overview

Along with migrating Containers between Hardware Nodes, you may wish to move a stand-alone physical server running a Linux operating system to a Container on your Node. The migration process includes copying the whole contents of the physical server (all its files, directories, quota limits, configuration settings, and so on) to a Container on the Hardware Node. After the server migration, you will have its exact copy in a Container including the operating system running inside the Container, the IP address(es) assigned to the Container, the amount of available disk space and memory, and so on.

Migration Steps

Before you start migrating a physical server to a Container on the Node, you should have a clear idea of the steps to be performed during the migration. The main steps of the migration procedure can be described as follows:

1. Creating the configuration file containing information on the main resources consumption on the physical server. This file is meant to be used for creating a Container on its basis. The data in the configuration file should be provided in the format readable by Parallels Virtuozzo Containers (i.e. in the form of `"`*`PARAMETER`*`"="`*`value`*`"`). Among other things, the file should include information on the Linux distribution your physical server is running and the number of user/group IDs allowed for Container internal disk quota. Detailed information on quota limits and Linux distributions is provided in the **Managing Resources** chapter (p. 120) and in the *Parallels Virtuozzo Containers 4.7 Reference Guide*, respectively.
2. Copying the configuration file made on the previous step from the physical server to the Hardware Node. You may copy the configuration file to any directory on the Node; the full path to this file should be specified during the physical server migration.

   This step is automatically performed by migrating a physical server to a Container using Parallels Virtuozzo Containers Tools (Parallels Management Console and Parallels Virtual Automation).
3. Creating a Container on the basis of the configuration file copied to the Node. In this step, you can also specify an OS template to be used for creating the Container. Using an OS template for the Container creation enables you to save RAM and disk space used by this Container on the Hardware Node. In case an OS template is not specified, the `mkvzfs` command is executed during the Container creation which makes an empty private area with the name of `/vz/private/`*`CT_ID`* on the Node. In the next step, all the physical server files including its system and application files will be copied to the `/vz/private/`*`CT_ID`* directory. Detailed information on OS templates is given in the *Parallels Virtuozzo Containers 4.7 Templates Management Guide*.
4. Migrating the physical server to the created Container. During the server migration, the following operations are consecutively performed:

   - All the files, directories, etc. are copied from the server to the Container on the Node by means of `rsync` - a utility providing the fast incremental data transfer. For more information on `rsync`, please see the man pages for this utility.
   - All the services on the physical server except for the critical ones (e.g. the `sshd` service needed to provide communication between the physical server and the Node) are stopped. This prevents the running services from modifying any files being moved. However, it depends entirely on you what services to stop.
   - The files, directories, etc. transferred to the Container during the first `rsync` run are compared with those on the physical server and, if any changes to the files have been made during the files migration, they are copied to the Container once more by means of `rsync` allowing to transfer just the differences between the two sets of files. This step is performed only if you chose the OS template for the Container creation on Step 3.

> **Note:** If the migration process fails on this step, the /vz/private/*CT_ID* directory on the Hardware Node will contain all the copied files and directories and may occupy a great amount of disk space. You can keep the directory, which will greatly speed up the repeated migration procedure, or manually remove the directory by using the rm utility.

5 Migrating the disk quota limits imposed on the selected partition from the physical server to the created Container. You may specify only one partition on the physical server which will be migrated to the Container on the Node together with all quotas imposed on it. All the other partitions of the server will be copied without keeping their quota limits. Moreover, the quota limits of the migrated partition will be applied to the entire Container after the server migration. Detailed information on the quota limits is provided in the vzquota subsection of the *Parallels Virtuozzo Containers 4.7 Reference Guide* and in the **Managing Resources** chapter (p. 120).

6 Executing the post-migration scripts depending on the Linux distribution the physical server was running. The names of the scripts to be run are read from the corresponding distribution configuration file in the /etc/vz/conf/dists directory on the Hardware Node. The scripts themselves and located in the /etc/vz/conf/dists/scripts directory on the Node. They are needed to tune the Container to be able to start it. Any script can be launched by executing the vzctl runscript *CT_ID script_path* command on the Node where *CT_ID* denotes the ID of the Container where the physical server has been migrated and *script_path* is the full path to the script on the Node.

7 Stopping the physical server and starting the Container on the Node.

Parallels Virtuozzo Containers allows you to complete all these steps using the following tools:

- the vzp2v command-line utility
- Parallels Management Console
- Parallels Virtual Automation

The aforementioned steps can be automatically performed while running the Parallels Virtual Automation migration wizards. However, if you plan to use the vzp2v utility to migrate a physical server to a Container, you need to create the configuration file manually by means of the vzhwcalc utility and copy it to the Hardware Node before starting the migration process itself. You can also use this utility before migrating a physical server in Parallels Management Console and/or Parallels Virtual Automation to find out the resources consumption on the server during its maximal loading and set the right resources parameters when running the migration wizards in Parallels Virtual Automation and Parallels Management Console. Detailed information on the vzhwcalc utility and on how to create and modify configuration files for Containers is provided in **Preparing a Container Configuration File** (p. 283).

Besides, when using vzp2v, you have to manually stop the physical server and start the Container on the Node after the server migration whereas Parallels Management Console and Parallels Virtual Automation allow you to select the corresponding options in the last step of their wizards.

The migration procedure by means of the `vzp2v` utility is described in the following subsections. Detailed information on how to migrate a physical server to a Container in Parallels Virtual Automation and Parallels Management Console is provided in the *Parallels Virtual Automation Administrator*'s *Guide* and Parallels Management Console online help.

Migration Requirements

To avoid delays and problems while migrating physical servers to Containers, make sure that the following requirements are fulfilled in respect of the server and the Hardware Node:

- The physical server is running a Linux distribution (for example, CentOS 5 or Red Hat Enterprise Linux 5).

  Note: None of the BSD operating systems is supported.

- The Linux distribution installed on the physical server is supported by Parallels Virtuozzo Containers. To find out if your Linux distribution can be recognized by Parallels Virtuozzo Containers, you can check the `/etc/vz/conf/dists` directory on the Node and look for the configuration file of your Linux distribution. It should have the name of `*Linux_Distribution_Name*-*version*.conf` where `*Linux_Distribution_Name*` and `*version*` denote the name of the Linux distribution running on your physical server and its version, respectively (e.g., `redhat-5.conf`). In case there is no corresponding distribution in the directory, you can proceed in one of the following ways:
  - Create a new distribution configuration file and place it to the `/etc/vz/conf/dists` directory on the Node. For information on creating new configuration files, see **Creating Configuration Files for New Linux Distributions** (p. 321).
  - Start the migration process without having the right configuration file for your Linux distribution. In this case the `unknown.conf` distribution configuration file from the `/etc/vz/conf/dists` directory on the Node will be used for tuning the Container after the physical server migration. However, using the `unknown.conf` configuration file means that you will not be able to use standard Parallels Virtuozzo Containers utilities (e.g. `vzctl`) for performing the main operations on the created Container (such as setting the Container IP address or configuring the DNS parameters) and have to manually complete these tasks from inside the Container.
- A network connection can be established among the physical server to be migrated and the Hardware Node.
- `ssh` is installed on both the physical server and the Hardware Node. `ssh` is used to provide secure encrypted and authenticated communication between the server and the Hardware Node. You can check if the `ssh` package is already installed on the server by executing the `ssh -V` command.

Migration Restrictions

Although Parallels Virtuozzo Containers allows you to migrate virtually any physical server running a Linux distribution to a Container, there is a number of limitations which should be taken into account before deciding on the migration process:

- During the migration, all the filesystems available on the physical server are joined to one filesystem in the Container—VZFS (Virtuozzo File System). Detailed information on VZFS is given in **Virtuozzo File System** (p. 20).
- If the physical server has more than one IP address assignedto the physical server, all these IP addresses will be reassigned to one and the same device on the Node—`venet0`—a virtual network adapter used to connect all the Containers on the given Hardware Node among themselves and with the Node. After the migration, you can create additional virtual network adapters inside the Container and decide what IP address to be assigned to what network adapter. For detailed information on how to create and manage Container virtual network adapters, please turn to the **Managing Virtual Network Adapters** section (p. 222).
- During the migration process, you may specify only one partition on the physical server which will be migrated to the Container on the Node together with all quotas imposed on it. All the other partitions of the server will be copied without keeping their quota limits. Moreover, the quota limits imposed on the selected partition will be applied to the entire Container after the server migration.
- While migrating your physical server running a Linux operating system with the security-enhanced (SE) Linux kernel, please keep in mind that the SE Linux kernel is currently not supported by Parallels Virtuozzo Containers. Therefore, the Container where the server running the SE Linux distribution has been migrated will not support the SE security features.
- If any of your files and/or directories on the physical server have extended attributes associated with them, these attributes will be lost after the server migration.
- Raw devices on the physical server cannot and will not be migrated to the Container on the Hardware Node.
- If you are running an application which is bound to the physical server MAC address, you will not be able to run this application inside the Container after the server migration. In this case, you can do one of the following:
    - If you are running a licensed application, you should obtain a new license and install the application inside the Container anew.
    - If you are running a non-licensed application, you can try to reconfigure the application and to make it work without being bound to any MAC address.
- If the migration process fails on the step of transferring files and directories from the physical server to the Container by means of `rsync`, the `/vz/private/`*`CT_ID`* directory on the Hardware Node will contain all the copied files and directories and may occupy a great amount of disk space. You can keep the directory, which will greatly speed up the repeated migration procedure, or manually remove the directory by using the `rm` utility.

Migrating in Command Line

This section describes how to migrate physical servers to Containers using the `vzp2v` command-line utility.

Preparing a Container Configuration File

If you wish to migrate a physical server to a Container in the command line, i.e. by using the `vzp2v` utility, you need to create the server configuration file manually and place it to the Hardware Node before starting the migration process. The configuration file contains information on the main server settings: its resource management parameters (e.g. disk space and the number of inodes consumed by the server, the server CPU power), network-related parameters (e.g. the server IP address and hostname), etc. During the physical server migration, information on the resources parameters from the configuration file is used to create a Container on their basis.

To prepare a configuration file for the physical server migration, you should perform the following operations:

- Copy the `vzhwcalc` utility from the Hardware Node to the server; you will need `vzhwcalc` to create the server configuration file.
- Copy the `distdetect-common.sh` script from the Hardware Node to the server; this script is used to determine the Linux version your server is running.
- Create the configuration file by running the `vzhwcalc` utility on the server.
- Edit the configuration file, if needed, and copy it to the Hardware Node.

As a result of the aforementioned operations, a valid configuration file should be created in the format readable by Parallels Virtuozzo Containers and copied to the Hardware Node. This file will be used to create a Container on its basis and the path to the file should be specified as the value of the `-c` option while running the `vzp2v` utility.

Creating Container Configuration Files

To create a configuration file of your physical server, you should first copy the vzhwcalc utility and the distdetect-common.sh script from the Hardware Node to the physical server. By default, vzhwcalc and distdetect-common.sh are stored in the /usr/local/bin and /usr/local/share/vzlinmigrate directories on the Node, respectively. The vzhwcalc utility is used to create a configuration file containing information on the server main resource parameters and used to create a Container on its basis. In its turn, the distdetect-common.sh script is intended to determine what Linux distribution the server is running and to set the value of the DISTRIBUTION variable in the generated configuration file in accordance with the detected distribution. You may copy the vzhwcalc and distdetect-common.sh file to any directory on the physical server.

When launched, the vzhwcalc utility scans the main resources on your physical server, makes a snapshot of their consumption, and writes down this information to the server configuration file. Besides, the utility initiates the execution of the distdetect-common.sh script used to determine the Linux version installed on your server and to put this information to the generated configuration file.

So, after you have copied the vzhwcalc and distdetect-common.sh files to the physical server, you should run the vzhwcalc utility on it to create a configuration file for your server:

```
# vzhwcalc --scan-time time -p time -d script_path
```

where --scan-time is the time during which the vzhwcalc utility will be periodically making snapshots of the main server resources, -p denotes the interval with which the resources snapshots will be made by the vzhwcalc utility, and -d is the full path to the distdetect-common.sh script on the server. The time and interval should be given in the dhms format (e.g. --scan-time 1d2h30m40s means that the vzhwcalc utility will run on the server for 1 day, 2 hours, 30 minutes, and 40 seconds).

While running the vzhwcalc utility, please keep in mind the following:

- The consumption of the resources may significantly vary depending on the server loading. Therefore, we recommend that you set the scan time of the vzhwcalc utility to 1 day or more. During this time, the utility will periodically (i.e. with the interval specified) check the resources consumption on the server. As a result, the configuration file will be created on the basis of the peak values reached by the resources during the time specified. By default, all the resource parameters are calculated by vzhwcalc with a 150% allowance as compared to their maximal values (except for memory which is calculated with a 120% allowance compared to its maximal value). However, you can use the --mem-scale and --disk-scale options to set your own enlargement factor by which the calculated memory and disk space resources parameters will be increased in the configuration file.
- After executing vzhwcalc, you will be presented with a list of directories on the physical server which are highly recommended to be excluded from the migration process. The names of these directories should be given as the value of the --exclude option while running the vzp2v utility.
- During the vzhwcalc execution, the following warning messages may be displayed:

- A message informing you that the `distdetect-common.sh` script has failed to determine the Linux distribution your physical server is running. In this case you should manually specify your distribution name as the value of the `DISTRIBUTION` variable in the created configuration file. Detailed information on how to work with the `DISTRIBUTION` variable is provided in the next subsection.
- A message informing you that your physical server has two or more network interface cards installed. In this case all IP addresses assigned to several network interfaces on the server will be reassigned to one virtual network adapter on the Node - `venet0`. This virtual adapter will be used by the created Container to communicate with the other Containers on the Node and with the outer world.
- A message containing a list of peer-to-peer IP addresses that cannot and will not be migrated to the Container to be created.
- A message informing you that the Linux OS installed on your physical server supports Native POSIX Thread Library (NPTL). For more information on NPTL, see the **Migration Restrictions** subsection (p. 282).

The configuration file created by the `vzhwcalc` utility is placed to the same directory on the physical server from where you have run this utility and has the default name of `ve.conf`. However, you can pass the `-o` option to `vzhwcalc` and set a name of your choice for the resulting configuration file.

Editing Container Configuration Files

After you have created the Container configuration file with the default name of `ve.conf`, you should check this file for the resources values listed in it. As has been mentioned above, the resource parameters in the configuration file are calculated on the basis of the physical server maximum load. However, you may wish to increase the resources available (e.g. in case you wish to exploit the Container to be created more intensively than the physical server). You can do it by opening the `ve.conf` file for editing (for example, by means of `vi`) and entering new values for the corresponding parameters.

Along with editing the resource parameters, you should also look for the `DISTRIBUTION` variable in the configuration file used to define what post-migration scripts are to be executed depending on the Linux distribution set in this file:

- If the `DISTRIBUTION` variable is present in the file:
  - Make sure that the distribution configuration file whose name is indicated as the value of the `DISTRIBUTION` variable is present in the `/etc/vz/conf/dists` directory on the Node. All distribution configuration files have `.conf` as their extension added to the corresponding distribution name (e.g. `redhat.conf`).
  - In case there is no corresponding distribution configuration file in the `/etc/vz/conf/dists` directory, create a new distribution configuration file with the name specified as the value of the `DISTRIBUTION` value in the `ve.conf` file and place it to this directory. More information on the distribution file creation see below.
- If the `DISTRIBUTION` variable is absent in the file meaning that the Linux version running on the physical server could not be detected, you should do the following:
  - Create a new distribution configuration file for the Linux version running on the server and place it to the `/etc/vz/conf/dists` directory on the Node.
  - Specify the name of the newly created distribution configuration file as the value of the `DISTRIBUTION` variable in the `ve.conf` configuration file.

  Detailed information on how to create new configuration files and set the `DISTRIBUTION` variable is provided in the **Creating Configuration Files for New Linux Distributions** section (p. 321).

You can also start the migration process without having the right configuration file for your Linux distribution. In this case the `unknown.conf` distribution configuration file from the `/etc/vz/conf/dists` directory on the Node will be used for tuning the Container after the physical server migration. However, using the `unknown.conf` configuration file means that you will not be able to use standard Parallels Virtuozzo Containers utilities (e.g. `vzctl`) for performing the main operations on the created Container (such as setting the Container IP address or configuring the DNS parameters) and have to manually complete these tasks from inside the Container.

Finally, you should copy the resulting configuration file to the Hardware Node. You will have to specify the full path to the configuration file while running the `vzp2v` utility.

Linux distribution installed on the physical server is supported by Parallels Virtuozzo Containers. To find out if your Linux distribution can be recognized by Parallels Virtuozzo Containers, you can check the `/etc/vz/conf/dists` directory on the Node and look for the configuration file of your Linux distribution. It should have the name of *`Linux_Distribution_Name`*`-`*`version`*`.conf` where *`Linux_Distribution_Name`* and *`version`* denote the name of the Linux distribution running on your physical server and its version, respectively (e.g. `redhat-5.conf`).

Migrating the Physical Server

Now that you have created the configuration file and copied it to the Hardware Node, you can start the migration procedure itself. To migrate a physical server to a Container, the `vzp2v` utility is used.

Let us assume that you wish to migrate a physical server running the Red Hat Enterprise Linux Server 5 (RHEL 5) operating system and having the IP address of `199.199.109.109` to Container 101 on your Hardware Node; moreover, you are supposed to use the `root` user name and the `3e5rrt4` password to log in to the server. To this effect, you should issue the following command on the Node:

```
# vzp2v root@199.199.109.109 --ctid 101 -c /etc/ve.conf \
-q /private_data -t -d rhel-5 redhat-el5-x86 \
--exclude=/proc/* --exclude=/usr/games -S iptables,crond
```

The options passed to the `vzp2v` utility in the example above are explained in the following table:

Option Name	Description
`--ctid`	Mandatory. The ID of the Container that will be created on the Node and where the physical server will be migrated. You can specify any unoccupied ID on the Node.
`-c`	Mandatory. The full path to the configuration file on the Node that was created on the physical sever by means of the `vzhwcalc` utility. You may specify only the name of the configuration file if you run the `vzp2v` utility from the directory where this file is located.
`-q`, `--quota`	Optional. The partition on your physical server which has any user and/or user groups quotas imposed on it. This partition will be migrated to the Container together with all quotas imposed on it. Moreover, these quotas will be applied to the entire Container after the server migration.
`-d`, `--dist`	Optional. The Linux version your physical server is running. The name of the version specified should coincide with the name of the corresponding distribution configuration file located in the `/etc/vz/conf/dists` directory on the Node. For example, if you specify `rhel-5` as the value of this option, the `rhel-5.conf` file should be present in the `/etc/vz/conf/dists` directory on the Node. You should obligatorily set this option, if there is no `DISTRIBUTION` variable specified in the server configuration file. In case the `DISTRIBUTION` variable is set in the configuration file and you have specified the `-d` option, the latter takes precedence.
`-t, --ostmpl`	Optional. The OS template to be used to create the Container. You may list all OS templates installed on the Node together with their updates by executing the `vzpkgls` command. The names of OS templates usually correspond to those of Linux distributions (e.g. `redhat-el5-x86` as in the example above); so you can easily guess what OS template to use for your Linux distribution. In case an OS template is not specified, the `mkvzfs` command is executed during the Container creation which makes an empty private area with the name of `/vz/private/CT_ID` on the Node. This private area is then used to copy all the physical server files to it.
`--exclude`	Optional. The path to the directories and files which will be excluded from copying to the Container. This option allows you to avoid migrating the data you do not need. To gain more understanding on this option, please

consult the man pages for the `rsync` utility.

> **Note**: We strongly recommend that you exclude the files and directories you were informed of while running the `vzhwcalc` utility on the physical server.

`-S, --srvstop`

Optional. The services to be stopped for the time of the physical server migration. We recommend that you stop all the services on the physical server except for the critical ones (e.g. the `sshd` service that is needed to provide communication between the physical server and the Node) before the migration. This will prevent the running services from modifying any files being moved.

In the example above, the following operations are performed during the physical server migration:

1 The `vzp2v` utility connects to the physical server with the IP address of `199.199.109.109` by using the `root` user name. While establishing a network connection, you will be asked for the password of `root` to log in to the server and have to enter `3e5rrt4` (which is, in our case, the password of the `root` user).

2 The `/etc/ve.conf` file is read and the `101.conf` file is created on its basis in the `/etc/vz/conf` directory on the Node.

3 Container 101 is created on the basis of the `101.conf` file and the `redhat-el-x86` OS template.

4 All the data except for the `/usr/games` directory and the contents of the `/proc` directory is copied from the physical server to Container 101.

5 The `iptables` and `crond` services are stopped on the physical server.

6 The files copied to Container 101 are compared with those on the physical server and, if any changes to the files were made during the 4th migration step, these changes are copied to Container 101.

7 The quota limits that were imposed on the `/private_data` partition on the physical server are copied to the Container. These quota limits are applied to the entire Container.

8 The post-migration script specific for the RHEL 5 OS is executed. The name of the script to be run is read from the `rhel-5.conf` distribution configuration file located in the `/etc/vz/conf/dists` directory on the Node and is needed to tune the Container before its starting.

Creating Customized Containers

If you wish to use one or more custom applications in many identical Containers, you may want to create Containers with necessary applications already preinstalled and tuned to meet your demands.

Parallels Virtuozzo Containers offers several ways to create customized Containers with preinstalled applications:

- By creating an OS EZ template cache with preinstalled application templates.
- By making a customized base OS EZ template and using it as the basis for Containers.
- By making a non-base OS EZ template and using it as the basis for Containers.
- By making a customized application EZ template, adding it to a new configuration sample file, and using this sample file as the basis for Containers.

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

Using Customized OS EZ Templates

You can make a customized base OS EZ template which can then be used to create Containers with a set of application already tuned to meet your demands. To make such a template, do the following:

1. Create a metafile that will serve as the basis for your customized base OS EZ template.

   > **Notes:**
   >
   > 1. Detailed information on how to create metafiles is given in the *Parallels Virtuozzo Containers 4.7Templates Management Guide*.
   >
   > 2. While creating a metafile for a new OS EZ template, make sure that the value of either the `%osname` parameter or the `%version` parameter in the metafile differs from the names or versions of all base OS EZ templates installed on the Node.

2. Create one or more scripts that will be executed on different stages of the OS EZ template life cycle and customize your applications to meet your needs. For example, you can create a postinstall script with the name of `post_install.bash` and make it perform a number of customization operations on some application included in the OS EZ template after installing this application inside your Container.
3. Create a customized OS EZ template by running the `vzmktmpl` utility and passing the corresponding options to it. So, you can use the `--post-install` option and specify the path to the `post_install.bash` script from the example above to make an OS EZ template that will customize your application after installing it inside your Container.

   > **Note:** For the full list of options allowing you to specify what scripts are to be executed on what stage of the EZ template life cycle, see **vzmktmpl** in the *Parallels Virtuozzo Containers 4.7 Command Line Reference Guide*.

4. Install the customized OS EZ template on the Node using the `rpm -i` command.
5. Cache the created OS EZ template by running the `vzpkg create cache` command. Detailed information on how you can do it is provided in the *Parallels Virtuozzo Containers 4.7 Templates Management Guide*.
6. Create a Container based on the OS EZ template.

For example, to create a Container that will run CentOS 5 and have the customized `mysql` and `apache` applications installed right after its creation, you can do the following:

1. Create a metafile for the Cent OS EZ template, name it, for example, `centos_5_customized.metafile`, and save in the `/root/centos_5` directory on the Node.
2. Make a script that will perform a number of custom operations after applying the `mysql` and `apache` application EZ templates to the Container, and name it `post_install.bash`.
3. Copy the script to the `/root/centos_5` directory on the Node.

4. Execute the following command on the Node to create the CentOS 5 OS EZ template:

```
# vzmktmpl /root/centos_5/centos_5_customized.metafile \
            --post-install /root/centos5/post_install.bash
```

   This command will create an OS EZ template for CentOS and put it to the `/root` directory (for example, `/root/centos_customized-5-x86-ez-4.7.0-1.noarch.rpm`).

5. Install the resulting OS EZ template on the Node:

```
# rpm -i /root/centos_customized-5-x86-ez-4.7.0-1.noarch.rpm
```

6. Cache the installed OS EZ template:

```
# vzpkg create cache centos_customized-5-x86
...
Complete!
Packing cache file centos_customized-5-x86.tar.gz ...
Cache file centos_customized-5-x86.tar.gz [14M] created.
```

7. Create Container 101 on the basis of the new OS EZ template:

```
# vzctl create 101 --ostemplate centos_customized-5-x86
                     --config basic
Creating Container private area (centos_customized-5-x86)
Container is mounted
Postcreate action done
Container is unmounted
Container private area was created
Delete port redirection
Adding port redirection to Container(1): 4643 8443
```

So you have just created Container 101 having the customized `mysql` and `apache` applications installed inside it.

Using EZ OS Template Sets

Another way of creating customized Containers is to make a non-base OS EZ template (also known as an OS EZ template set) differing from the corresponding base OS EZ template in the number of packages included in this template. For example, if you wish a Container to run CentOS 5 and to function as a Linux-based server only, you can create the `centos-5-x86-server` OS EZ template set and include only those packages in it that are needed for performing main server tasks. So, you can specify packages to be used for setting up file and print sharing and exclude all the packages for graphical interfaces (GNOME and KDE).

To create a non-base OS EZ template, do the following:

1 Create a metafile that will serve as the basis for your non-base OS EZ template. Any metafile for this kind of EZ template must contain the following information:

- `%osname`: the name of the Linux distribution for which you are creating the OS EZ template set. This name must correspond to that specified in the base OS EZ template. For example, if you are creating an OS template set of the base OS EZ template for CentOS 5, set the value of this parameter to `centos`.
- `%osver`: the version of the Linux distribution specified as the value of the `%osname` parameter. This name must correspond to that specified in the base OS EZ template. For example, if you are creating an OS template set of the base OS EZ template for CentOS 5, set the value of this parameter to `5`.
- `%osarch`: the system architecture where the EZ template is to be run. This name must correspond to that specified in the base OS EZ template. For example, if you are creating an OS template set of the base OS EZ template for CentOS 5, set the value of this parameter to `x86`.
- `%setname`: the name to be assigned to your non-base OS EZ template. You can specify any name you like for your OS template set:

a This name will be added to the name of the base OS EZ template after the indication of the architecture where the OS EZ template is to be run. For example, if you are creating an OS template set of the base OS EZ template for CentOS 5 that is supposed to run on x86 platforms, the name of your non-base OS EZ template should look like `centos-5-x86-`*`Template_Name`*`-ez-1.0-1.noarch.rpm`, where *`Template_Name`* is the name you specify as the value of the `%setname` parameter.

b This name will also be assigned to the directory which will store the meta data of your non-base OS EZ template after the template installation on the Node. For example, it will have the name of `/vz/template/centos/5/x86/config/os/my_non_base_template` if you set the value of this parameter to `my_non_base_template`, create a non-base OS EZ template for CentOS 5, and installed it on the Node.

- `%packages`: a list of RPM packages to be included in the non-base OS EZ template. This parameter allows you to specify what applications will be present inside your Containers based on this OS EZ template set right after their installation. The names of the packages listed as the value of this parameter must correspond to the names of real RPM packages (without indicating the package version, release, architecture, and the `.rpm` extension) that are stored in the repository used for managing your EZ templates.

> **Note:** You can also specify a number of additional parameters in your metafile. For example, you may wish to add one or several extra packages to your OS EZ template set which are not available in the repository used to handle the packages for the corresponding base OS EZ template. For this purpose, you will have to specify the `%mirrorlist` parameter providing information on the repository where these extra packages are kept. Detailed information on all parameters you can set in metafiles is given in the *Parallels Virtuozzo Containers 4.7 Command Line Reference Guide*.

2 You can also (though you do not have to) create a number of scripts that will be executed on different stages of the non-base OS EZ template life cycle and customize your applications to meet your demands. The path to these scripts should then be specified after the corresponding options while creating your OS template set. For example, you can create a preinstall script with the name of `pre_install.bash` and make it perform a number of customization operations on some application included in the non-base OS EZ template before installing this application in your Container.

> **Note:** If there are no scripts for a non-base OS EZ template, the scripts available for the corresponding base OS EZ template will be executed.

3 Create the non-base OS EZ template by running the `vzmktmpl` utility and passing the corresponding options to it, if needed. So, if you created one or several scripts in the previous step, you can use special options and specify the path to these scripts during the command execution. For example, you can use the `--pre-install` option and specify the path to the `pre_install.bash` script to make an OS EZ template that will customize your application before installing it inside your Container.

> **Note:** For the full list of options allowing you to specify what scripts are to be executed on what stage of the EZ template life cycle, see **vzmktmpl** in the *Parallels Virtuozzo Containers 4.7 Command Line Reference Guide*.

4 Install the non-base OS EZ template on the Node using the `rpm -i` command.

5 Cache the created OS EZ template by running the `vzpkg create cache` command. Detailed information on how you can do it is provided in the *Parallels Virtuozzo Containers 4.7 Templates Management Guide*.

6 Create a Container based on the OS EZ template.

Using Customized Application Templates

If the number of customized applications inside your Containers is relatively small, you can also use the following way of creating customized Containers:

1 Create a metafile that will serve as the basis for your customized application EZ template.

> **Note:** Detailed information on how to create metafiles is given in the **Creating Metafiles for EZ Templates** section of the *Parallels Virtuozzo Containers 4.7 Templates Management Guide*.

2 Create one or more scripts that will be executed on different stages of the application EZ template lifecycle and customize your applications to meet your demands. For example, you can create a postinstall script with the name of `post_install.bash` and make it perform a number of customization operations on your application after installing this application in your Container.

3 Create a customized application EZ template by running the `vzmktmpl` utility and passing the corresponding options to it. So, you can use the `--post-install` option and specify the path to the `post_install.bash` script from the example above to customize your application in accordance with your needs after installing it in your Container.

> **Note:** The full list of options allowing you to specify what scripts are to be executed on what stage of the EZ template lifecycle is provided in the **vzmktmpl** section of the *Parallels Virtuozzo Containers 4.7 Reference Guide*.

4 Install the customized EZ template on the Node using the `rpm -i` command.

5 Create a new Container configuration sample file and include the customized EZ template in this file. Detailed information on Container configuration sample files is provided in the **Managing Container Resources Configuration** section (p. 171).

6 Create a customized Container on the basis of the configuration sample.

The following example demonstrates how to create Container 101 that will run CentOS 5 and have the customized `mysql` application installed right after its creation:

1 Create a metafile for the `mysql` application, name it `mysql.metafile`, and save in the `/usr/mysql` directory on the Node.

2 Make a script that will perform a number of custom operations after applying the `mysql` EZ template to the Container, and name it `post_install.bash`.

3 Copy the script to the `/usr/mysql` directory on the Node.

4 Execute the following command on the Node to create the `mysql` EZ template:

```
# vzmktmpl /usr/mysql/mysql.metafile \
           --post-install /usr/mysql/post_install.bash
```

This command will create an EZ template for the `mysql` application and put it to the `/root` directory (e.g., `/root/mysql-centos-5-x86-ez-4.7.0-3.noarch.rpm`).

5 Install the `mysql` EZ template on the Node. Using the example above, you can install the template as follows:

```
# rpm -ihv /root/mysql-centos-5-x86-ez-4.7.0-3.noarch.rpm
```

6 Create a new Container configuration sample file and add the `mysql` EZ template to a list of templates that will be installed in Containers created on the basis of this configuration sample file. For example, you can create a new configuration sample with the `mysql` name by running the **Create Configuration Sample** wizard in Parallels Management Console and add the `mysql` EZ template to the list of templates in the **Select Application Templates** step of this wizard.

7 Create Container 101 by using the `vzctl create` command and the `mysql` sample file:

```
# vzctl create 101 --ostemplate centos-5-x86 --config mysql
Creating Container private area (centos-5-x86)
Container is mounted
Postcreate action done
Container is unmounted
Container private area was created
Delete port redirection
Adding port redirection to Container(1): 4643 8443
```

So, you have just created Container 101 that already has the customized `mysql` application installed.

Changing System Time From Containers

Normally, it is impossible to change the system time from a Container. Otherwise, different Containers could interfere with each other and could even break applications depending on the system time accuracy.

Normally only the Node system administrator can change the system time. However, if you want to synchronize the time via Network Time Protocol (NTP), you have to run NTP software, which will connect to external NTP servers and update the system time. It is not advisable to run application software on the Node itself, since flaws in the software can lead to compromising all Containers on this Node. Thus, if you plan to use NTP, you should create a special Container for it and configure it to have the `sys_time` capability. The example below illustrates configuring such a Container:

```
# vzctl set 101 --capability sys_time:on
Unable to set capability on running Container
Saved parameters for Container 101
```

The output of the above command warns you that `vzctl` cannot apply changes in the capabilities to a running Container. The Container has to be restarted before changes take effect:

```
# vzctl stop 101; vzctl start 101
Stopping Container ...
Container was stopped
Container is unmounted
Starting Container ...
Container is mounted
Adding IP address(es): 192.168.1.101
Hostname for Container set: Container101
Container start in progress...
# ssh root@ct101
root@ct101's password:
Last login: Mon Feb 28 23:25:58 2007 from 10.100.40.18
[root@ct101 root]# date
Mon Feb 28 23:31:57 EST 2007
[root@ct101 root]# date 10291300
Tue Feb 29 13:00:00 EST 2007
[root@ct101 root]# date
Tue Feb 29 13:00:02 EST 2007
[root@ct101 root]# logout
Connection to Container101 closed.
# date
Tue Feb 29 13:01:31 EST 2010
```

The command session above shows the way to change the system time from Container 101. The changes will affect all the Containers and the Node itself. It is not advisable to have more than one Container with the `sys_time` capability set on.

NTP is described in Internet Standard RFC 1305; more information including client software can be obtained from the NTP web server (http://www.ntp.org).

Setting Up an iSCSI Environment in Parallels Virtuozzo Containers Systems

iSCSI (Internet Small Computer System Interface) is a TCP/IP-based protocol meant for transmitting data over local area networks (LANs), wide area networks (WANs), or the Internet and providing location-independent data storage and retrieval. The iSCSI protocol is mainly used to interconnect hosts (e.g. database servers) with shared storage systems on SANs (Storage Area Networks). In this connection, it aims at achieving the following goals:

- *Storage Consolidation*. Various storage resources from many servers around the network can be moved to one or more central locations (e.g. data centers) on the SAN, which allows you to allocate storage resources more efficiently. For example, any server on the SAN can be allocated a new disk volume without making changes to the server resources. Similarly, any server upgrades or expansions can be performed without impacting the storage resources on the SAN.
- *Disaster Recovery* and *Business Continuity*. The iSCSI protocol can be used to allow for remote data replication and near real time data backup across vast distances providing a cost effective solution to disaster recover and business continuity.

The implementation of an iSCSI storage system in a Parallels Virtuozzo Containers environment does not differ from that in standard environments and is based on the three main components: a TCP/IP network, an initiator, and a target. The interaction among the components in a Parallels Virtuozzo Containers-based system may roughly be described as follows:

- A Hardware Node acting as an initiator sends a SCSI command (request) over the TCP/IP network to the target represented by a SCSI data storage system (i.e. one or more SCSI storage devices).
- The target processes the received request and takes the appropriate action.

To configure a Hardware Node to communicate with a target (e.g. some SCSI storage device) via the iSCSI protocol, you should perform the following operations on the Node:

1 Install the `iscsi-initiator-utils` RPM package providing the server daemon for the iSCSI protocol and the necessary utilities for its managing:

```
# rpm -ihv iscsi-initiator-utils-6.2.0.742-0.5.el5.i386.rpm
```

2 Discover your iSCSI target using the `iscsiadm` utility:

```
# iscsiadm --mode discovery --type sendtargets --portal <target_IP_address>
```

where `<target_IP_address>` denotes the IP address used to access the target.

3 Log in to the target using the `iscsiadm` utility:

```
# iscsiadm --mode node --login automatic
```

This command saves the information about the target to the `/var/lib/iscsi/nodes` directory on the Hardware Node, which allows your Node to automatically detect the iSCSI target on its boot.

After completing the operations above, a new iSCSI device should appear under the /dev directory on your Node. You can find out the device name using the fdisk -l or tail -f /var/log/messages command.

Now you can mount the iSCSI device to your Hardware Node using the mount utility. Assuming that your iSCSI device has the name of /dev/sdb1 and you wish to mount it to the /vz directory on your Node, this can be done as follows:

```
# mount /dev/sdb1 /vz
```

Note: If you have not yet partitioned your target, you should partition it and create a filesystem on it (using the fdisk and mkfs utilities) prior to mounting the iSCSI device to your Node.

You can also automate the procedure of mounting your iSCSI partition on the Hardware Node boot by editing the /etc/fstab file. For example, if you wish to have the /dev/sdb1 partition automatically mounted on the Node boot and this partition is formatted to ext3, you can add the following string to the /etc/fstab file:

```
/dev/sdb1 /vz ext3 defaults 0 0
```

Important! If your iSCSI partition is formatted to ext3, make sure that you have this partition mounted to only one Hardware Node at a time; otherwise, the SCSI storage may become corrupted.

Obtaining the Hardware Node ID From Containers

The default Parallels Virtuozzo Containers installation does not allow users inside a Container to obtain any information specific to the Hardware Node the Container is running on. The reason is that no Container shall have knowledge about the corresponding Hardware Node. A Container can be transparently migrated to another Hardware Node, and if this Container runs any applications depending on the particular Node, these applications might fail after the migration.

There are however situations when you have to provide some unique Hardware Node ID to some applications. For example, you might want to license your application per Hardware Node. In this case, after the migration your customer will need to re-apply the license for your application.

Parallels Virtuozzo Containers provides access to the unique Hardware Node ID via the /proc/vz/hwid file. The default Parallels Virtuozzo Containers installation makes this file accessible to Containers from 1 to 100 (i.e. Containers with reserved IDs). It is possible to change this range in the global configuration file. For example, this is the way to make the file visible in Containers from 1 to 1000:

```
# vi /etc/vz/vz.conf
VZPRIVRANGE=”1 1000”
# vzctl exec 101 cat /proc/vz/hwid
0C3A.14CB.391B.6B69.02C9.4022.3E2F.CAF6
```

The above example illustrates accessing the Hardware Node ID from Container 101.

Mounting the /vz Partition via the Parallels Virtuozzo Containers Script

If you experience problems with mounting or accessing the /vz partition (for example, due to some data corruption) and this interferes with the Hardware Node boot-up procedure, you can prevent the /vz partition from being mounted at the Hardware Node startup and have it mounted by a special /etc/init.d/vz script only after the Node is up and running.

To start using the vz script for mounting the /vz partition after the Hardware Node boot, do the following:

1 Open the /etc/fstab file on the Hardware Node for editing, and set the noauto flag for the /vz partition. After editing, your fstab file may look as follows:

```
LABEL=/         /               ext3 defaults 1 1
LABEL=/vz       /vz             ext3 defaults,noauto 1 2
LABEL=SWAP-sda3 swap            swap defaults 0 0
...
```

2 Make sure that the value of the VZMOUNTS parameter in the /etc/sysconfig/vz file on the Hardware Node is set to vz, as shown below:

```
VZMOUNTS="vz"
```

From this point on, the vz script will be used to automatically mount the /vz partition after the Hardware Node boot. During its execution, the script will do the following:

- Search the /etc/fstab file on the Node for partitions having the noauto flag set.

  Note: As the /etc/init.d/vz script checks the /etc/fstab file for all partitions with the noauto flag set, you can also have any other partition automatically mounted by this script after the Hardware Node boot rather than at the boot time by setting noauto for the corresponding partition in the /etc/fstab file and indicating the partition name as the value of the VZMOUNTS parameter in the /etc/vz/vz.conf file.

- Check if these partitions are mounted. If they are not, it will:

  a Run the fsck utility to examine the partitions and repair them if there are any errors or data loss (please keep in mind that it may take a rather long run to check and fix a damaged file system).

  b Mount the partitions.

If the /vz partition has errors that cannot be corrected automatically by the script, you can remotely log in to the Hardware Node and troubleshoot the problem.

Managing Mount Points In Containers

The previous versions of Virtuozzo (3.0 and earlier) provide you with the ability to remount any part of the Hardware Node file hierarchy and to have it automatically mounted to/unmounted from a particular Container on its startup/shutdown using special system-wide or per-Container mount/umount action scripts. In Parallels Virtuozzo Containers 4.7, this can also be done with the help of the `vzctl` utility. Along with defining what part of the Hardware Node file hierarchy is to be automatically mounted inside a Container on its booting, you can also use `vzctl` to configure certain options (or flags) to be applied to the mounted directories. Currently, you can set the following options for mounted Container directories:

- `noexec`. This option does not allow the execution of any binaries in the mounted directory.
- `nodev`. This option does not allow to interpret character or block special devices in the mounted directory.
- `nosuid`. This option does not allow set-user-identifier or set-group-identifier bits to take effect.

You can manage the mounted directories inside Containers (and, as a consequence, the aforementioned directory options) using the `--bindmount_add` option of the `vzctl set` command. For example, you can execute the following command to set the `noexec` flag for the `/tmp` directory inside Container 101, thus forbidding the execution of any binaries in this directory:

> **Note:** You can set mount points for and remove them from stopped Containers only; the mount points will become active/inactive on the Container startup.

```
# vzlist -a
      CTID      NPROC STATUS    IP_ADDR         HOSTNAME
         1         32 running   127.0.1.2       localhost
       101          - stopped   10.12.12.101    -
# vzctl set 101 --bindmount_add /tmp,noexec --save
Saved parameters for Container 101
# vzctl start 101
Starting Container ...
Container is mounted
Set up bind mount(s): /tmp
...
```

To check that the directory has been successfully mounted with the specified option, you can run the following command:

```
# vzctl exec 101 mount
vzfs  on / type vzfs (rw)
simfs on /tmp type simfs (rw,noexec)
proc  on /proc type proc (rw,nodiratime)
```

The directories mounted inside Containers using the `--bindmount_add` option are displayed as the ones of the `simfs` type. So, the command output above shows that the `/tmp` mount point is currently available inside Container 101 and that this mount point has the following flags set for it: `rw` and `noexec`.

If a directory to be remounted does not exist inside a Container, this directory is created under /vz/private/*CT_ID*/mnt/*Dir_Name* on the Hardware Node (where *Dir_Name* is the name of the directory you wish to mount) and becomes visible from inside the Container under the / directory. For example, assuming that there is no /root/MyTempDir directory inside Container 101, you can issue the following command to create such a directory inside the Container and mount it with the noexec flag:

```
# vzctl set 101 --bindmount_add /root/MyTempDir,noexec --save
Saved parameters for Container 101
# ls -R /vz/private/101/mnt
/vz/private/101/mnt:
media root

/vz/private/101/mnt/root:
MyTempDir
...
# vzctl exec 101 ls /root
MyTempDir
```

While working with mounted directories, please keep in mind the following:

- There are no restrictions on migrating a Container with one or several mount points inside. Having been moved to the Destination Node, the Container will have the same mount points with the same flags (noexec, nodev, nosuid) as it had on the Source Node before the migration.
- The permissions set for the mounted directories are taken from the corresponding upper-level directories (e.g. the permissions for the MyTempDir directory inside Container 101 in the example above are derived from the /root directory inside the Container).
- If there is no upper-level directory, the directory permissions are set to 0777 meaning that owners, groups, and others have read, write, and search permissions in respect of this directory.
- For mount points quota accounting, standard per-Container quota calculation rules are used since all bind mounts are located in the /vz/private/*CT_ID*/mnt directory on the Hardware Node.

At any time you can remove a mount point from a Container. For example, you can delete the /tmp mount point from Container 101 by executing the following command:

```
# vzlist -a
      CTID      NPROC STATUS    IP_ADDR         HOSTNAME
         1         32 running   127.0.1.2       localhost
       101          - stopped   10.12.12.101    -
# vzctl set 101 --bindmount_del /tmp --save
Saved parameters for Container 101
```

Preserving Application Data During Container Reinstallation

A typical Container reinstallation creates a new Container instead of the broken one using the corresponding OS and application templates and mounting the filesystem of the broken Container to the `/tmp` directory inside the new one, which does not let the necessary data from the old Container get lost. However, a manual copying of the broken Container contents to the new Container may prove a tedious and time-consuming task. Beginning with version 3.0.0 SP1, The Parallels Virtuozzo Containers software allows to automate this process by performing special scripts that would copy the relevant data to the appropriate places of the new Container after the reinstallation. Naturally, these scripts deal with the data of particular applications only; in fact, this functionality should be supported by application templates that should carry the reinstall scripts specific to them and install them to the `/etc/vz/reinstall.d` directory inside the Container. Only then will Parallels Virtuozzo Containers be able to make use of them, should the Container be reinstalled one day.

Let us consider a typical scenario of such an automation by the example of the Plesk application:

1 The Plesk application template is repackaged to include the necessary reinstall scripts.

> **Note:** Usually it is up to the application vendor or the template maker to provide this kind of scripts. However, if you have a certain experience with making application templates yourself, you may do it on your own. The reinstall scripts should be first packaged into an RPM, which should in its turn be added to the template.

2 A new Container is created and the Plesk application template is added to it. Part of this addition consists in copying the reinstall scripts to the `/etc/vz/reinstall.d` directory inside the Container.

3 A Plesk license is manually copied to the appropriate place inside the Container and installed.

4 The Container administrator performs typical day-to-day tasks with the help of the Plesk control panel. The local Plesk database gets filled up with all kinds of objects (servers, domains, hostnames, IP addresses, logs, etc.).

5 Some day the Container gets broken and wouldn't start. The Container administrator clicks the **Reinstall** button in Parallels Power Panel. At this point Parallels Virtuozzo Containers:

   a Creates a brand-new Container with the necessary templates added to it. This means that Plesk is also added and the `/etc/vz/reinstall.d` directory with the Plesk scripts is created.

   b Mounts the filesystem of the broken Container to the `/mnt` directory inside the new Container.

   c **NB:** Launches scripts from the `/etc/vz/reinstall.d` directory. These scripts are executed one by one in the alphabetical order. They take care of copying both the Plesk license and the Plesk database to the new Container and installing the license.

   d Dismounts the old filesystem from the /mnt directory.

6 The Container administrator gets their working Container again with the Plesk application having retained both its license and database, so no manual copying is involved in the process.

When launching the vzctl reinstall command from the command line, you have the option to drop certain scripts from the reinstallation procedure. This can be done with the help of the --scripts option:

```
# vzctl reinstall 101 --scripts 'script1 script2'
```

In this example only the scripts named script1 and script2 will be launched at the end of the reinstallation, and all the other scripts from the Container /etc/vz/reinstall.d directory will be discarded.

Accessing Devices From Inside Containers

It is possible to grant a Container read, write, or read/write access to a character or block device. This might be necessary, for example, for Oracle database software if you want to employ its ability to work with raw disk partitions.

In most cases, providing access to the file system hierarchy for a Container is achieved by using bind mounts. However, bind mounts do not allow you to create new partitions, format them with a file system, or mount them inside a Container. If you intend to delegate disk management to a Container administrator, you shall use either the `--devices` or the `--devnodes` option of the `vzctl set` command.

The example session below illustrates the following situation: you want to allow the root user of Container 101 to take responsibility for administering the `/dev/sdb`, `/dev/sdb1` and `/dev/sdb2` devices. In other words, you allow the Container 101 system administrator to repartition the `/dev/sdb` device and create file systems on the first two partitions (or use them with any software capable of working with raw block devices, such as Oracle database software).

First, we are going to grant the Container the permissions to work with the needed block devices:

```
# vzctl set 101 --devices b:8:16:rw --devices b:8:17:rw --devices
 b:8:18:rw --save
Setting devperms
Saved parameters for Container 101
```

This command sets the read/write permissions for block devices with major number 8 and minor numbers 16, 17 and 18 (corresponding to `/dev/sdb`, `/dev/sdb1`, and `/dev/sdb2`). If you are not sure which major and minor numbers correspond to the necessary block devices, you may issue the following command:

```
# ls -l /dev/sdb{,1,2}
brw-rw----    1 root     disk       8,   16 Jan 30 13:24 /dev/sdb
brw-rw----    1 root     disk       8,   17 Jan 30 13:24 /dev/sdb1
brw-rw----    1 root     disk       8,   18 Jan 30 13:24 /dev/sdb2
```

Now let us create a 100-Mb Linux partition in addition to an already existing 2 GB partition on `/dev/sdb1` from Container 101.

```
[root@ct101 root]# fdisk /dev/sdb

Command (m for help): p

Disk /dev/sdb: 255 heads, 63 sectors, 2231 cylinders
Units = cylinders of 16065 * 512 bytes

   Device Boot    Start       End    Blocks    Id  System
/dev/sdb1   *         1       255   2048256    83  Linux

Command (m for help): n
Command action
   e   extended
   p   primary partition (1-4)
p
```

```
Partition number (1-4): 2
First cylinder (256-2231, default 256):
Using default value 256
Last cylinder or +size or +sizeM or +sizeK \
(256-2231, default 2231): +100M

Command (m for help): p

Disk /dev/sdb: 255 heads, 63 sectors, 2231 cylinders
Units = cylinders of 16065 * 512 bytes

   Device Boot    Start       End    Blocks   Id  System
/dev/sdb1   *         1       255   2048256   83  Linux
/dev/sdb2           256       268    104422+  83  Linux

Command (m for help): w
```

After the new partition table has been written, you can format it and mount inside the Container:

```
[root@ct101 root]# mke2fs /dev/sdb2
[Output of mke2fs is skipped…]
[root@ct101 root]# mount /dev/sdb2 /mnt
[root@ct101 root]# df
Filesystem           1k-blocks      Used Available Use% Mounted on
vzfs                   1048576    149916    898660  15% /
ext2                    101107        13     95873   1% /mnt
```

Remember that you have to specify all minors for the devices you want to delegate authority for; allowing to access `/dev/sdb` grants the permission to create, modify and delete partitions on it, but explicit permissions shall be given for partitions you allow the Container to work with.

Moving Network Adapters to Containers

By default, all Containers on a Node are connected among themselves and with the Node by means of a virtual network adapter called venet0. In Parallels Virtuozzo Containers 4.7, there is a possibility for a Container to directly access a physical network adapter (for example, eth1). In this case the adapter becomes inaccessible to the Hardware Node itself. This is done with the help of the vzctl command:

```
# vzctl set 101 --netdev_add eth1 --save
Add network device: eth1
Saved parameters for Container 101
```

Mind that the network device added to a Container in such a way has the following limitations:

- This network device will be accessible only to the Container whereto it has been moved, but not to the Hardware Node (Container 0) and not to all the other Containers on the Node.
- The port redirection mechanism is not supported for this network device.
- The Parallels Virtuozzo Containers class-based traffic shaping, if set for the given Container, does not limit the bandwidth for this network device.
- If such a device is removed from the Container (by means of the vzctl set --netdev_del command) and added to another Container instead, all the network settings of this device are purged. To work around this problem, you should store all the device settings in the ifcfg-dev file and have this file available in the /etc/sysconfig/network-scripts directory inside all the Containers that may have access to this device (including Container 0). After the device has been added to a Container, it will be enough to issue the ifup *dev* command inside the Container to read the settings from the file mentioned above. Mind though that this will still not restore advanced network configuration settings, such as traffic shaping or packet filtering rules.
- The physical device inside a Container has no security restrictions typical for the venet virtual device. Inside the Container it will be possible to assign any IP address to this device and use it, to sniff network traffic in the promiscuous mode, and so on.

Enabling VPN for Containers

Virtual Private Network (VPN) is a technology which allows you to establish a secure network connection even over an insecure public network. Setting up a VPN for a separate Container is possible via the TUN/TAP device. To allow a particular Container to use this device, the following steps are required:

- Make sure the `tun.o` module is already loaded before Parallels Virtuozzo Containers is started:

```
# lsmod
```

- Allow the Container to use the TUN/TAP device:

```
# vzctl set 101 --devices c:10:200:rw
```

- Create the corresponding device inside the Container and set the proper permissions:

```
# vzctl exec 101 mkdir -p /dev/net
# vzctl exec 101 mknod /dev/net/tun c 10 200
# vzctl exec 101 chmod 600 /dev/net/tun
```

Configuring the VPN proper is carried out as a common Linux administration task, which is out of the scope of this guide. Some popular Linux software for setting up a VPN over the TUN/TAP driver includes Virtual TUNnel `<http://vtun.sourceforge.net/>` and OpenVPN `<http://openvpn.sourceforge.net/>`.

Managing Hardware Node Resources Parameters

Parallels Virtuozzo Containers allows you to configure a number of resource management parameters defining the amount of resources to be allocated to the Hardware Node (also known as Container 0). These parameters include all standard UBC parameters (`VMGUARPAGES`, `KMEMSIZE`, `OOMGUARPAGES`, etc.) as well as the `ONBOOT` parameter.

You can edit any of these parameters in the `/etc/vz/conf/0.conf` file on the Hardware Node by means of your favorite text editor (for example, `vi` or `emacs`) or by using the `vzctl set` command and specifying `0` as the Container ID. For example:

```
# vzctl set 0 --kmemsize 12211840:14359296 --save
Saved parameters for Container 0
```

This command sets both the barrier and limit values of unswappable kernel memory (in bytes) which can be allocated to internal kernel structures of the processes on the Node. The specified parameter values will be in force until the Hardware Node restart. If you wish these values to be applied to the Node on its next booting, you should additionally set the `ONBOOT` parameter in the `/etc/vz/conf/0.conf` file to `yes`. This can be done in one of the following ways:

- Passing the `--onboot` option to the `vzctl set` command:

```
# vzctl set 0 --onboot yes
Saved parameters for Container 0
```

- Editing the `/etc/vz/conf/0.conf` file with your favorite text editor (e.g. `vi`) and setting the value of the `ONBOOT` parameter in this file to `yes`.

> **Note:** Detailed information on all resource parameters that can be changed in respect of your Hardware Node is provided in the *Parallels Virtuozzo Containers 4.7 Reference Guide*.

If you have made a number of changes to Hardware Node resource management parameters and wish to reset them to the values specified in the `/etc/vz/conf/0.conf` file, you can proceed as follows:

```
# vzctl set 0 --reset_ub
UBC limits were set successfully
```

Setting Immutable and Append Flags for Container Files and Directories

You can use standard Linux utilities—chattr and lsattr–to set extra flags for files and directories inside your Containers and to query their status, respectively. Currently, two of these extra flags—'append' and 'immutable'—are supported. For example, you can execute the following command to set the 'immutable' flag for the /root/MyFile file inside Container 101:

```
[root@ct101 root] chattr +i /root/MyFile
```

To check that the 'immutable' flag has been successfully set, use the following command:

```
[root@ct101 root] lsattr /root/MyFile
----i-------- /root/MyFile
```

Note: For detailed information on the chattr and lsattr utilities, see their manual pages.

Creating Local Repository Mirrors for vzup2date

The vzup2date-mirror utility allows you to create local mirrors of the Parallels Virtuozzo Containers official repository storing the latest versions of the Parallels Virtuozzo Containers software (i.e. newest versions of the Parallels Virtuozzo Containers core and utilities) and used by vzup2date to keep your current Parallels Virtuozzo Containers installation up-to-date. You can also use this utility to make local mirrors of updated standard and EZ OS and application templates.

When executed, vzup2date-mirror completes a number of tasks (connects to the Parallels Virtuozzo Containers official repository, downloads the specified Parallels Virtuozzo Containers software updates or updated templates to the server where your mirror is located, etc.) resulting in building a local mirror of the Parallels Virtuozzo Containers official repository. The created mirror can then be used to update all your Hardware Nodes from one and the same location on your local network. Building your own local repository mirrors results in less Internet bandwidth consumption and more rapid software updates deployments to your Nodes.

The following subsections provide information on how you can create your own local mirrors of the Parallels Virtuozzo Containers official repository using the vzup2date-mirror utility.

Parallels Virtuozzo Containers Repository Structure

Before starting to create your own local mirror, it is important to you to have a clear idea of the structure of the Parallels Virtuozzo Containers official repository. This knowledge will be of service to you later on while running `vzup2date-mirror` and specifying the part of the Parallels Virtuozzo Containers repository for which you wish to create a mirror (i.e. while deciding on what Parallels Virtuozzo Containers update release or what Parallels Virtuozzo Containers templates are to be downloaded).

The official Parallels Virtuozzo Containers repository is organized as a directory tree at the top of which the `/virtuozzo` directory (the root of the tree) is located. The further repository structure may be described as follows:

- Beneath the root is the directory containing the information about the operating system the packages of which are stored in the Parallels Virtuozzo Containers repository. In our case, it is Linux; so, the full name of the directory is `/virtuozzo/linux`. Please note that you are not allowed to access the root of this directory.
- The next underlying directory represents the microprocessor architecture for which the packages stored in the Parallels Virtuozzo Containers repository are meant. Currently, you can make use of the following directories:
  - `i386`: this directory is meant for Parallels Virtuozzo Containers RPM packages and templates to be used on 32-bit platforms;
  - `x86_64`: this directory is meant for Parallels Virtuozzo Containers RPM packages and templates to be used on x86-64-bit platforms (e.g. on servers with the AMD Opteron and Intel Pentium D processors installed);

  Each of the aforementioned directories contains a number of files holding the information on all update releases for the corresponding architecture (e.g. `index.xml`) and on particular update releases (e.g., `index_4.7.0.xml` or `update_ids.4.7.0`).
- The next underlying directories are the following:
  - The `eztemplates` directory containing a set of OS and application EZ templates for the corresponding microprocessor architecture. This directory contains two files—`index.xml` and `update_ids`—holding the information on all available EZ template updates.
  - The `templates` subdirectory containing a set of OS and application standard templates for the corresponding microprocessor architecture. This directory contains two files—`index.xml` and `update_ids`—holding the information on all available standard template updates.
  - A directory representing the major Parallels Virtuozzo Containers release version for the corresponding microprocessor architecture (e.g., `/virtuozzo/linux/i386/4.7.0` for the Parallels Virtuozzo Containers 4.7 release). This directory contains the `index.xml` and `update_ids` files holding the information on all available updates for the given release, a number of additional `xml` files and subdirectories described below.

- A number of subdirectories containing updated packages for particular Parallels Virtuozzo Containers components (e.g., `/virtuozzo/linux/i386/4.7.0/TU-4.7.0-3` keeping updates for your current Parallels Virtuozzo Containers utilities).

Creating a Local Mirror

The process of creating your local repository mirror which will be locally available to your Hardware Nodes includes the following main stages:

1. Installing the `apache` application on the server where your local mirror will be kept, if it is not yet installed. Currently, you can create HTTP-based mirrors only; so, `apache` is needed to make your server function as a web server.

   > **Note:** We recommend that you always store your mirrors inside individual Containers or on dedicated servers not to compromise the Hardware Node security.

2. Installing the `vzup2date-mirror` RPM package shipped with the Parallels Virtuozzo Containers distribution using the `rpm -i` command.
3. Configuring the `vzup2date-mirror` configuration file that will be used by this utility on the step of connecting to the Parallels Virtuozzo Containers official repository and deciding what updates to download to your local mirror.
4. Running the `vzup2date-mirror` utility on the server where you are going to set up the mirror. This will create a special directory on this server and download all the required packages from the Parallels Virtuozzo Containers official repository to this directory.
5. Telling the `vzup2date` utility to use the local mirror for updating your Parallels Virtuozzo Containers software instead of connecting to the Parallels Virtuozzo Containers official repository. To do this, you should replace the value of the `Server` parameter in the `/etc/sysconfig/vzup2date/vzup2date.conf` file on each Hardware Node where the `vzup2date` utility is to be run with the path to your local mirror.

Let us clear up the aforementioned statements by following the example below. In this example we presume the following:

- You wish to create a local repository mirror that will store system files for the 32-bit version of the Parallels Virtuozzo Containers 4.7 release and use it to update all Hardware Nodes in your local network.
- Your mirror will be located in the `/var/www/html` directory inside Container 101.
- Container 101 is started and has the IP address of `192.168.0.101` assigned to it (i.e. it can be accessed from your local network using this IP address).

  > **Note:** You can also assign a public IP address to the Container and make it accessible from your Hardware Nodes on other networks.

- The `apache` web server is running inside Container 101 and the default document root for `apache` is `/var/www/html`.

To create a local mirror and make it available to your Hardware Nodes, you should perform the following operations:

1 Log in to Container 101 (e.g., via SSH) and install the `vzup2date-mirror` package there. For example:

```
# rpm -ihv vzup2date-mirror-4.7.0-3.noarch.rpm
```

> **Note:** You may need to additionally install a number of Perl packagesto satisfy the `vzup2date-mirror` dependencies. For example, if you are creating a local mirror in a Container based on the `sles-10-x86_64` EZ OS template, you have to install the `perl-Crypt-SSLeay` package before installing the `vzup2date-mirror` package in the Container.

2 Edit the `vzup2date-mirror.conf` file. It is located in the `/etc/vzup2date-mirror` directory inside Container 101. This file is used by the `vzup2date-mirror` utility to:

- retrieve the path and the credentials to access the Parallels Virtuozzo Containers official repository
- define what packages are to be downloaded to your local mirror
- define the place where the mirror is to be located

You can edit this file according to your needs or leave the default settings. For example, your `vzup2date-mirror.conf` file may look like the following:

```
Server=http://vzup2date.parallels.com
User=user1
Password=sample
HTTP_PROXY=http://192.168.1.20
HTTP_PROXY_PASSWORD=wer26sd2
HTTP_PROXY_USER=Peter
LocalRepositoryRoot=/var/www/html
Releases=i386/4.7.0
MirrorName=MyMirror
HTTPD_CONFIG_FILE=/etc/httpd/conf/httpd.conf
```

The aforementioned parameters define the behaviour of the `vzup2date-mirror` utility during the local mirror creation as follows:

- The `Server`, `User` and `Password` parameters are used by the utility when connecting to the Parallels Virtuozzo Containers official repository. As a rule, these parameters are set automatically and do not need to be modified.
- The `HTTP_PROXY` group of parameters should be used if you are connecting to the Internet via a proxy server.
- The `LocalRepositoryRoot` and `MirrorName` parameters define the mirror location and name, respectively.
- The `Releases` parameter determines the list of updates to be downloaded to the local mirror from the Parallels Virtuozzo Containers repository. For more information on how to configure this parameter, please see the **Choosing Updates for Downloading** section (p. 316).
- The `HTTPD_CONFIG_FILE` parameter defines the functioning of your local mirror as an HTTP-based server providing the path to the `httpd` configuration file. By default, this parameter is set to `/etc/httpd/conf/httpd.conf`. If you have not changed the default `httpd.conf` file location, you do not need to modify this parameter.

> **Note:** Detailed information on all the parameters that can be set in the `vzup2date-mirror` configuration file is provided in the *Parallels Virtuozzo Containers 4.7 Reference Guide*.

3 Create a local mirror inside Container 101:

```
# vzup2date-mirror
```

During the command execution, `vzup2date-mirror` will perform the following operations in accordance with the parameters set in the `vzup2date-mirror.conf` file:

- Connect to the Parallels Virtuozzo Containers official repository using the specified URL, credentials, and proxy server settings.
- Create the `/var/www/html/virtuozzo/linux/i386/4.7.0` directory inside Container 101 according to the values of the `LocalRepositoryRoot` and `Releases` parameters and copy all the packages contained in the subdirectories of the `/virtuozzo/linux/i386/4.7.0` directory of the Parallels Virtuozzo Containers official repository to the `/var/www/html/virtuozzo/linux/i386/4.7.0` directory inside Container 101.
- Create a number of files in the `/var/www/html/virtuozzo/linux/i386` directory (e.g., `index.xml` and `index_4.7.0.xml`) containing the information on all major system update releases available for the i386 architecture and on all minor update releases included in the Parallels Virtuozzo Containers 4.7 release.

> **Note:** To create a local mirror storing the latest versions of Parallels Virtuozzo Containers EZ templates, configure the `vzup2date-mirror.conf` file and specify the `-z` option when running the `vzup2date-mirror` utility, respectively. See the **Choosing Updates for Downloading** section (p. 316) and the *Parallels Virtuozzo Containers 4.7 Reference Guide* for details.

4 Set the value of the `Server` parameter in the `/etc/sysconfig/vzup2date/vzup2date.conf` file on each Hardware Node where the `vzup2date` utility is to be run to `http://192.168.0.101`.

From now on, the `vzup2date` utility will use the created local repository mirror to update all Hardware Nodes in your local network.

At any time, you can run `vzup2date-mirror` to check if there are any updates available to your local mirror. The second and all subsequent times you run the utility, it will download only those packages that are currently absent from your mirrored releases or the MD5SUM check sum of which differs from that of the packages in the mirrored releases and will put them to the corresponding directories. As for the aforementioned example, all changed packages for the 4.7 major release will be downloaded to the `/var/www/html/virtuozzo/linux/i386/4.7.0` directory inside Container 101.

Choosing Updates for Downloading

When executed without any options, the vzup2date-mirror utility downloads all the available system updates for all architectures and releases to your local mirror. If you want to download all available EZ templates updates, use the -z option. You can also make the utility download specific system and templates updates only. This can be done by editing the Releases parameter in the vzup2date-mirror.conf file. Let us assume that you want to get the following updates from the Parallels Virtuozzo Containers official repository:

- all system updates for the 32-bit version of Parallels Virtuozzo Containers 4.7
- all updates for the centos-5 and fedora-core-14 EZ templates intended for use on the x64 version of Parallels Virtuozzo Containers

To make the vzup2date-mirror utility download only the aforementioned updates to your local mirror, you need first to create two configuration files for vzup2date-mirror—one file per each update type (system and EZ template). The necessity of creating two separate files is caused by the fact that the format of the Releases parameter for system and EZ templates is different:

- For system updates, the Releases parameter should be set in the *arch*/*Parallels Virtuozzo Containers_release* format where *arch* and *Parallels Virtuozzo Containers_release* denote the microprocessor architecture and the major Parallels Virtuozzo Containers release version, respectively, for which the updates are to be downloaded (e.g., x86_64/4.7.0).
- For EZ templates updates, the Releases parameter should be set in the *arch*/*EZ_template_name* format where *arch* and *EZ_template_name* denote the microprocessor architecture and the name of the EZ template, respectively, for which the updates are to be downloaded (e.g., x86_64/fedora-core-14).

The easiest way to make two configuration files is to use the default /etc/vzup2date-mirror/vzup2date-mirror.conf file for system updates and create a copy of this file for EZ templates updates. Let us name this file vzup2date-mirror-z.conf and put it to the /etc/vzup2date-mirror directory.

Once you create the configuration files, you need to configure the Releases parameter in each file to tell the vzup2date-mirror utility to download certain system and templates updates only:

- Configure the Releases parameter in the vzup2date-mirror.conf file by setting its value to i386/4.7.0:

```
# vi /etc/vzup2date-mirror/vzup2date-mirror.conf
Releases=i386/4.7.0
```

- Configure the Releases parameter in the vzup2date-mirror-z.conf file by setting its value to x86_64/centos-5, x86_64/fedora-core-14:

```
# vi /etc/vzup2date-mirror/vzup2date-mirror-z.conf
Releases=x86_64/centos-5, x86_64/fedora-core-14
```

To set an upper limit on versions of system updates to download, use the optional `<ApproveSystemUpdate arch/release></ApproveSystemUpdate>` tag, where `arch` is system architecture (e.g. `x86_64`) and `release` is Parallels Virtuozzo Containers release (e.g. `4.7.0`). For example, if the current versions are `4.7` (major), `2.6.32-042stab020.1` (kernel), `4.7.0-99` (tools and command-line utilities), and you need to prevent any newer versions from being downloaded from the official repository to a local mirror (e.g. until they are properly tested first), add the following to `vzup2date-mirror.conf`:

```
<ApproveSystemUpdate x86_64/4.7.0>
MU=no
CU=2.6.32-042stab020.1
TU=4.7.0-99
</ApproveSystemUpdate>
```

Note: Major updates are allowed by default (`MU=yes`). This includes configurations where the entire `ApproveSystemUpdate` tag is omitted.

Now you can start downloading the specified updates. To do this, run the following commands on the server where your local mirror resides:

To download all system updates for the 32-bit version of Parallels Virtuozzo Containers 4.7:

```
# vzup2date-mirror
```

To download all updates for the `centos-5` and `fedora-core-14` EZ templates intended for x64 version of Parallels Virtuozzo Containers:

```
# vzup2date-mirror -z -c /etc/vzup2date-mirror/vzup2date-mirror-z.conf
```

The `-c` option in the last command tells the `vzup2date-mirror` utility to use the necessary parameters from the specified configuration files instead of the default one.

Managing iptables Modules

On systems running Parallels Virtuozzo Containers, you can use *iptables* to manage packet filtering and NAT rules for both physical servers and Containers.

Loading iptables Modules to the Hardware Node

You can configure the list of `iptables` modules that will be loaded on the Hardware Node after its startup as follows:

- By using standard means of your Host operating system:
    - On RHEL-based Nodes, by editing the `/etc/sysconfig/iptables-config` file with your favorite text editor (e.g. `vi`) and configuring the value of the `IPTABLES_MODULES` parameter in this file.
    - On SUSE-based Nodes, by editing the `/etc/sysconfig/SuSEfirewall2` file (e.g. by means of the YaST2 configuration tool).

  For example, if your Hardware Node is running Red Hat Linux Enterprise 5, you can make the `ip_conntrack_netbios_ns`, `ip_conntrack`, and `ip_conntrack_ftp` modules load on the Node startup by modifying the `IPTABLES_MODULES` parameter in the `/etc/sysconfig/iptables-config` file as follows:

```
IPTABLES_MODULES="ip_conntrack_netbios_ns ip_conntrack ip_conntrack_ftp"
```

- By editing the `/etc/vz/vz.conf` file on the Hardware Node. The `IPTABLES` parameter in this file determines the `iptables` modules that will additionally be loaded to the Node during the Parallels Virtuozzo Containers service startup. For example, you can indicate the following `iptables` modules as the value of this parameter to have them automatically loaded to your Hardware Node after the Parallels Virtuozzo Containers service startup:

```
IPTABLES="ipt_REJECT ipt_tos ipt_limit ipt_multiport iptable_filter
          iptable_mangle ipt_TCPMSS    ipt_tcpmss ipt_ttl ipt_length"
```

All the specified modules will be loaded on the Node startup after you reboot the Hardware Node.

Sharing a File System Among Containers

This section provides a simple example of what can be done with the help of Container action scripts. You need a basic BASH shell language knowledge to understand the examples.

Remember that when you source configuration files in your action script, you have two environment variables that show the path to Container file areas: $VE_ROOT and $VE_PRIVATE. You need to use $VE_ROOT since the VZFS file system does not follow mount points in the Container private area. In other words, if you mount a directory to the Container private area, the users inside the Container will not see this mount and you should use $VE_ROOT in your scripts.

This example shows how to create a configuration when two environments can share files and the necessary setup is automatically created at Containers startup. Let us assume that both environments want to have their user home directories in sync. For the sake of simplicity, let Container 102 (called test2) hold actual user directories and Container 101 (called test1) use them as well.

In this case, Container 102 does not need any action scripts. All the necessary setup is done by the mount script of Container 101. It can look like the following:

```
#!/bin/bash
#
# 101.mount - script to mount home dir of Container 102
# if one of these files does not exist then something is
# really broken
[ -f /etc/sysconfig/vz ] || exit 1
[ -f $VE_CONFFILE ] || exit 1
[ -f /etc/sysconfig/vz-scripts/$veid.conf ] || exit 1

# source these files. Note the order, it is important
. /etc/sysconfig/vz
. $VE_CONFFILE

# If home dirs are not mounted we exit with error
mount --bind /vz/root/102/home $VE_ROOT/home
exit $?
```

This script is intentionally simplified to focus on the main idea of mounting one Container directories inside another. However, it can be developed further by adding checkups for the Container 102 mount status (it is possible to call vzctl from the mount script, but do not call vzctl with the same Container ID as the Container the mount script is being executed for). It can source the Container 102 configuration file to determine correctly the VE_ROOT directory of Container 102.

In order to be able to stop Container 101, you have to create the umount script dismounting $VE_ROOT/home:

```
#!/bin/bash
#
# 101.umount - a script to umount home directory of Container 102
# If one of these files does not exist then something is
# really broken
[ -f /etc/sysconfig/vz ] || exit 1
[ -f $VE_CONFFILE ] || exit 1

# Source configuration files to access $VE_ROOT
. /etc/sysconfig/vz
. $VE_CONFFILE

# Dismount shared directory
umount $VE_ROOT/home
```

After starting Container 102 and 101, Containers will have a common `/home` directory.

It is possible to use the same technique for mounting the Hardware Node file system sub tree into a Container, to mount a block device into a Container (for example, a hard drive partition or a CD-ROM), and so on.

Creating Configuration Files for New Linux Distributions

Distribution configuration files are used to distinguish among Containers running different Linux versions and to determine what scripts should be executed when performing the relevant Container-related operations (e.g. assigning a new IP address to the Container). Detailed information on distributions configurations files is provided in the *Parallels Virtuozzo Containers 4.7 Reference Guide*.

All Linux distributions shipped with Parallels Virtuozzo Containers have their own configuration files located in the `/etc/vz/conf/dists` directory on the Hardware Node. However, you may wish to create your own distribution configuration files to support new Linux versions released. Let us assume that you wish your Container(s) to run the CentOS 5 Linux distribution and, therefore, have to make the `centos-5.conf` distribution configuration file to define what scripts are to be executed while performing major tasks with Containers running this Linux version. To this effect, you should do the following:

1 In the Container configuration file (with the name of `/etc/vz/conf/`*`CT_ID`*`.conf`), specify `centos-5` as the value of the `DISTRIBUTION` variable (for example, `DISTRIBUTION="centos-5"`).

2 Create the `centos-5.conf` configuration file in the `/etc/vz/conf/dists` directory. The easiest way to do it is copy one of the existing configuration files by executing the following command in the `/etc/vz/conf/dists` directory, for example:

```
# cp fedora.conf centos-5.config
```

In the example above, we assume that the `fedora.conf` file is present in the `/etc/vz/conf/dists` directory on the Hardware Node. In case it is not, you may use any other distribution configuration file available on your Node.

3 Open the `centos.conf` file for editing with the help of any text editor:

```
# vi centos-5.conf
```

4 In the `centos-5.conf` file, go to the first entry and, in the right part of the entry, specify the name of the script you wish to be run on issuing the `vzctl` command with the parameter specified in the left part of the entry. For example, if you wish the script to be executed while assigning a new IP address to your Container and the script has the `my_centos_script` name, your entry should look as follows:

```
ADD_IP=my_centos_script-add_ip.sh
```

> **Note:** The information on all acceptable parameters and their description are provided in the *Parallels Virtuozzo Containers 4.7 Reference Guide*.

5 Repeat **Step 4** for all entries in the file.

6 Place the scripts for the new Linux distribution to the `/etc/vz/conf/dists/scripts` directory on the Node. Make sure the names of these scripts coincide with those specified in the `centos-5.conf` file.

Rebooting Containers

When you issue the `reboot` command at your Linux box console, the command makes the reboot system call with argument ‘`restart`’, which is passed to the server BIOS. The Linux kernel then reboots the server. For obvious reasons this system call is blocked inside Containers: no Container can access BIOS directly; otherwise, a reboot inside a Container would reboot the whole Hardware Node. That is why the `reboot` command inside a Container actually works in a different way. On executing the `reboot` command inside a Container, the Container is stopped and then started by Parallels Agent, which handles this situation.

If you want a Container to be unable to initiate reboot itself, add the `ALLOWREBOOT=”no”` line to the Container configuration file (`/etc/vz/conf/`*`CT_ID`*`.conf`). If you want to have Container reboot disabled by default and want to specify explicitly which Containers are allowed to reboot, add the `ALLOWREBOOT=”no”` line to the global configuration file (`/etc/vz/vz.conf`) and explicitly specify `ALLOWREBOOT=”yes”` in the corresponding Container configuration files.

If the Parallels Agent software is not running on your Hardware Node, an auxiliary way to allow Containers to reboot themselves is to uncomment the following line in the `/etc/cron.d/vereboot` file:

```
# vi /etc/cron.d/vereboot
[beginning of file]
#* * * * * root /etc/vz/conf/vereboot
```

You can use any editor of your choice instead of the `vi` command. Remove the hash mark on the last line to read:

```
* * * * * root /etc/vz/conf/vereboot
```

Now you can issue the `reboot` command in a Container, and the latter will be started on the next `vereboot` run.

Managing Graphical Applications In Containers

The given section provides information on how you can run X applications inside Containers located somewhere on a TCP/IP network and display them on your local server, exploit window managers to customize the appearance of running X applications, and use the `vnc` desktop software to remotely launch graphical applications.

Running Graphical Applications in X Windows

Overview

You may wish to run X applications (X clients) such as `xclock`, `xmms`, etc. inside your Containers on a TCP/IP network and display the resulting output on your local server. This can be done with the help of the X Window System. The X Window System is based on the client/server model where an X server is the program responsible for controlling the display of the server on which you are working and an X client denotes an application program that communicates with the server, sending it various requests, such as "draw a line" or "pay attention to keyboard input".

To run X applications inside your Container located on a TCP/IP network and to display them on your local server, you should take care of the following:

- Install and configure a special software called an X server on the server where you wish X clients to be displayed.

  Note: In the following subsections, we assume that you have successfully installed and configured an X server on your local server. In case you have not, please download the X server software packages (e.g. from http://www.xfree86.org) and install them by following the instructions shipped with this software.

- Configure X clients (X applications) to direct their output to your local server where the X server is running.
- You may also wish to specify a window manager of your choice to be used for displaying your X clients.

A central concept of the X Window System is the display, an abstraction for the screen managed by an X server. When an X client is invoked, it needs to know which display to use. Displays are named by strings in the form of *`hostname`*`:`*`displaynumber`*`.`*`screennumber`* and should be set as the `DISPLAY` environment variable on the server where X clients are to be run (in our case inside the corresponding Container):

- *`hostname`* specifies the hostname or the IP address of the machine to which the display is physically connected, i.e. the server where the X server is running (e.g. `198.112.45.11:0.0`). An omitted hostname (e.g. `DISPLAY=:0.0`) would mean the local host.
- `displaynumber` is usually used to refer to a collection of monitors that share a common keyboard and pointer (mouse, tablet, etc.). Most workstations tend to have only one keyboard and pointer, and therefore, only one display. In case a workstation has several displays (i.e. several keyboards or pointer sets), each display on this server is assigned a display number (beginning at 0) when the X server for that display is started. The display number must always be given in a display name.
- `screennumber`. Some displays share a single keyboard and pointer among two or more monitors. Since each monitor has its own set of windows, it is assigned a screen number (beginning at 0) when the X server for that display is started. If the screen number is not given, screen 0 will be used.

For example, if your local server is known to the outside world as `my_local_computer` and located in the `my-domain.org` domain and you are running a normal X server on this server, the value of the `DISPLAY` variable in the Container environment where you wish to remotely run X clients should be set to `my_local_computer.my-domain.org:0.0`.

Using X Windows to Run Graphical Applications

The X Window System lets you start any X application inside any Container on a TCP/IP network and have it show up on your local server where an X server is installed. To run remote X applications, you should first of all tell the X applications running inside your Container to direct their output to the display of your local server. You can do it by specifying the DISPLAY environment variable inside the Container. For example, to run the xfig drawing program inside your Container and display its output on your local server with the IP address of 199.199.199.199, you should issue the following commands inside the Container:

```
# DISPLAY=199.199.199.199:0
# export DISPLAY
# xfig &
```

Along with setting the DISPLAY environment variable inside your Container, you should also open permissions to your X server so that X applications are allowed to use your local display. You can do it in one of the following ways:

- By using the host list mechanism (xhost). In this case the X server maintains a list of hosts which are allowed to connect to it.
- By using the magic cookie mechanism (xauth). In this case the X server allows access from any host having an authorization record (a magic cookie) stored inside the server.
- By forwarding X connections via ssh.

You can choose any of these ways to remotely run your X applications. However, by using the xhost and xauth mechanisms, authority records needed to establish a connection between an X server and X application are transmitted over the network with no encryption, whereas using ssh enables you to run X applications over encrypted connections. So, if you are worried that someone might snoop on your connections, you can use the X forwarding mechanism, as is shown in the example below.

Let us assume that you wish to run the xclock application inside Container 101 and display its output on your local server with the name of my_local_computer.my-domain.org. To do this, perform the following operations:

Note: Before running X applications inside a Container on a public network, check that this Container is accessible from your local server where the X server is to be run.

1 On the local server, execute the startx command:

```
# /usr/X11R6/bin/startx
```

This starts an X server with a basic terminal window (the default xterm application) on your server.

2 Once xterm is open, you should establish an ssh connection to a Container where you wish to run the xclock application:

```
# ssh CT_IP_Address
```

where *CT_IP_Address* denotes the IP address or hostname of the Container where your X client is to be run. As has been mentioned above, an ssh connection is used to provide security and stronger authentication for an X protocol connection between the X server and the X client by tunneling the X protocol, which is called X forwarding. Moreover, X forwarding automatically sets the DISPLAY variable inside the Container to point to your local server and directs the output of X clients running inside the Container to the X server on the local server. X forwarding is enabled in ssh1 and ssh2 by default; however, you may additionally use the -X option to enable X forwarding in case you are not sure that it is on.

3 After executing the command, you will be prompted for the password to log in to the Container. Provide the root user name and their password to log in to the Container and press **Enter**.

4 Now that you have successfully logged in to the Container, execute the echo $DISPLAY command to check the value of the DISPLAY variable in your Container environment. It should read: my_remote_computer.parallels.com:10.0. Unlike the xhost and xauth mechanisms where the display number in the DISPLAY variable reflects a real number of displays connected to a server (beginning at 0), ssh always uses the 10th display number - a special X display created by ssh itself - to pass X protocol information to your local server.

If you do not see any value when typing this command or the value is incorrect, set the DISPLAY variable in your Container environment as follows:

```
# DISPLAY=my_remote_computer.parallels.com:10.0
# export DISPLAY
```

5 Launch the xclock application displaying the current time in an analog form by issuing the following command:

```
# xclock
```

If a clock is shown on the screen of your remote server, you have successfully run the xclock application.

Note: While running the commands in our example, we assume that you work in the bash shell. While working in other Linux shells, you may need to use different commands to start your X server or to set the DISPLAY variable on your local server.

Defining a Window Manager to Run X Applications

The layout of windows on the screen in the X Window system is controlled by special programs called window managers. Window managers (like twm, wmaker, fvwm2, etc.) are programs that sit between an X server and normal X clients and control the way the running X clients are positioned, resized, or moved on your screen. Although a window manager decides to a great extent how X clients look and feel, it does not affect what client applications do within the window defined by this window manager.

The main operations that can be performed by means of window managers are the following:

- Start and terminate X clients.
- Move, resize, and rearrange the "vertical" stacking of windows.
- Refresh the screen(s).
- Determine which window is to receive input from your keyboard or mouse.
- Create and customize pop-up menus to complete any of the aforementioned tasks.

You can change the default window manager used to control the appearance of your X clients by editing the Xclients and xinitrc scripts located in the /usr/X11R6/lib/X11/xinit/ directory either inside your Container or on your local server. However, you can launch only one window manager at any time. So, if you are already running a local window manager, you cannot start the remote one (i.e. it will complain and exit).

Let us assume that you wish to run several X applications (xterm, oclock, emacs) inside your Container and to use the remote fvwm2 window manager to manage their output on the screen. To this effect, you can edit the /usr/X11R6/lib/X11/xinit/Xclients script inside your Container in the following way:

Note: We assume that you have successfully installed the fvwm2 window manager inside your Container. In case you have not, please download the needed software packages (e.g. from http://www.fvwm.org) and install them by following the instructions shipped with this software.

1 Log in to your Container and open the /usr/X11R6/lib/X11/xinit/Xclients file for editing:

```
vi /usr/X11R6/lib/X11/xinit/Xclients
```

This file is just a shell script containing commands that you wish to run when your X session starts (e.g. xterm, xclock).

2 Remove the existing text in the file and add the following strings to it:

Note: We recommend that you make a copy of the Xclients file in case something goes wrong.

```
#!/bin/sh
oclock -geometry 75x75-1-1 &
xterm -C -geometry 80x12+0+0 &
emacs &
fvwm2
```

The clients will be launched in the order in which they are listed in the file; the last line should specify the window manager where the started X clients will run.

3 Save the file.

In our example, the `Xclients` file starts three applications - `xterm`, `oclock`, and `emacs` - and the `fvwm2` window manager where these application are to be run. The `–geometry` options used in the example specify the size and shape of the window. `80x12+0+0` means a window that is 80 characters wide and 12 lines high, positioned at the upper left. The + and – numbers give the location of the window. The first number gives the X coordinate and the second one gives the Y coordinate. The + numbers start from the upper left of the screen; the – numbers start from the lower right of the screen. So, `+0+0` means to put the `xterm` application at the upper left corner. Numbers greater than `0` are used to put things in the middle of the screen as in case with the `oclock` window (a round clock) in our example.

Running Graphical Applications via VNC

You may also wish to use VNC (Virtual Network Computing) to remotely run graphical applications inside your Container and display them on your local server. The main features of VNC are the following:

- The server and the client may be on different computer and on different types of computers. The protocol which connects the server and the viewer is simple, open, and platform independent.
- No state is stored at the viewer. Breaking the viewer's connection to the server and then reconnecting will not result in any loss of data. Because the connection can be remade from somewhere else, you have easy mobility.
- The VNC protocol is designed to adapt to the amount of bandwidth available which makes it ideal for thin client deployments.

To start using VNC, you should perform the following operations:

- Install a virtual X server - `vnc` - inside your Container. The `vnc` servers are not associated with a physical display, but provide a "fake" one X clients (`xterm`, `mozilla`, etc.) can attach to.
- Install a `vnc` client - `vncviewer` - on your local server to connect to the `vnc` server from anywhere on the network.
- Connect to the `vnc` server with the `vnc` viewer.

Let us run the `xclock` application inside Container 101 with the hostname of `Container101.com` located on a TCP/IP network and display it on your local server by using VNC.To this effect, you should do the following:

> **Note:** We assume that you have successfully installed a `vnc` server inside your Container and a `vnc` client on your local server. If you have not, please download the needed software packages (e.g. from http://www.realvnc.com) and install them by following the instructions shipped with this software or available on the web site.

1 Log in to Container 101 and start your `vnc` server by issuing the following command:

```
# vncserver
```

If you have never run a `vnc` server before, you will be prompted for a password, which you will need to use when connecting to this server. All the `vnc` servers on your remote server will use the same password; you can change it at a later time by using the `vncpasswd` command. Type the password you consider suitable and press **Enter**.

2 Execute the `echo $DISPLAY` command to check what display number will be used by the `vnc` server to run graphical applications. As you have learnt in the previous subsections, the main X display of a workstation is usually indicated as `0` (in our case it will read `:0`; the hostname is omitted because the `vnc` server is running inside the Container itself). When you run a `vnc` server inside your Container, it will appear as `:1`, as if it were just an additional display. Normally, the `vnc` servers will choose the first available display number and tell you what it is. However, you can specify your own display number (for example, `2`) by typing the following:

```
# vncserver :2
```

You can also cause graphical applications to use a `vnc` server rather than the normal X display by setting the `DISPLAY` variable in the Container environment to the `vnc` server you want (in the examples below, we assume that the display number for the `vnc` server is set to `2`):

```
# export DISPLAY=CT101:2
```

or by starting a graphical application with the `-display` option:

```
# xterm -display CT101:2 &
```

3 Now you should connect the `vnc` viewer running on your local server to the `vnc` server. You can do it by executing the following command on your local server:

```
# vncviewer CT101.com:2
```

where `CT101.com` is the hostname of Container 101 where the `vnc` server is running and `1` denotes the number of the display used by the `vnc` server to run graphical applications.

> **Note:** While using hostnames for connecting to a Container, make sure that your Container has a valid DNS entry. Otherwise, you should replace its hostname with the corresponding IP address.

You can control the way graphical applications are positioned, resized, or moved on the screen of your local server by specifying different options for the `vncserver` command, as you do it by using window managers while running X applications. For example, you can pass the `-geometry` option to `vncserver` to set the size of the desktop to be created (by default, it is 1024x768). You can get a list of all options for the `vncserver` command by giving `-h` as its option.

Assigning External IP Addresses to Containers

In Parallels Virtuozzo Containers 4.7, you can assign external IP addresses to Containers. External IP addresses are considered valid IP addresses by the `venet0` adapter so that the adapter does not drop IP packets coming from Containers and having external IP addresses set as source IP addresses. However, unlike normal IP addresses, external IP addresses are not set as alias addresses in Containers and are not announced via Address Resolution Protocol (ARP). Moreover, you can assign the same external IP address to several Containers, which you cannot do with a normal IP address, and the Containers may reside on both the same or different Hardware Nodes.

You may wish to assign external IP addresses to Containers when implementing load-balancing configurations. As a rule, such configurations require that you assign one and the same virtual IP address—the IP address where the service you plan to provide will live—to several Containers. You cannot do this using normal IP addresses, but you can use the functionality provided by external IP addresses and assign the same virtual IP address to all necessary Containers.

To assign an external IP address to a Container, you can use the `--ext_ipadd` option of the `vzctl set` command. For example, to set the external IP address of `10.10.10.101` for Container 101, you can execute the following command:

```
# vzctl set 101 --ext_ipadd 10.10.10.101 --save
Adding external IP addresses: 10.10.10.101
Saved parameters for Container 101
```

The specified IP address should be set as the value of the `EXT_IP_ADDRESS` parameter in the Container configuration file. So you can check the configuration file of Container 101 to ensure that the operation has finished successfully, for example:

```
# grep EXT_IP_ADDRESS /etc/vz/conf/101.conf
EXT_IP_ADDRESS="10.10.10.101"
```

At any time, you can delete the added external IP address from Container 101 using this command:

```
# vzctl set 101 --ext_ipdel 10.10.10.101 --save
Deleting IP addresses: 10.10.10.101
Saved parameters for Container 101
```

If Container 101 has more than one external IP address assigned, you can delete all its external IP addresses as follows:

```
# vzctl set 101 --ext_ipdel all --save
Saved parameters for Container 101
```

CHAPTER 10

Mastering Parallels Management Console

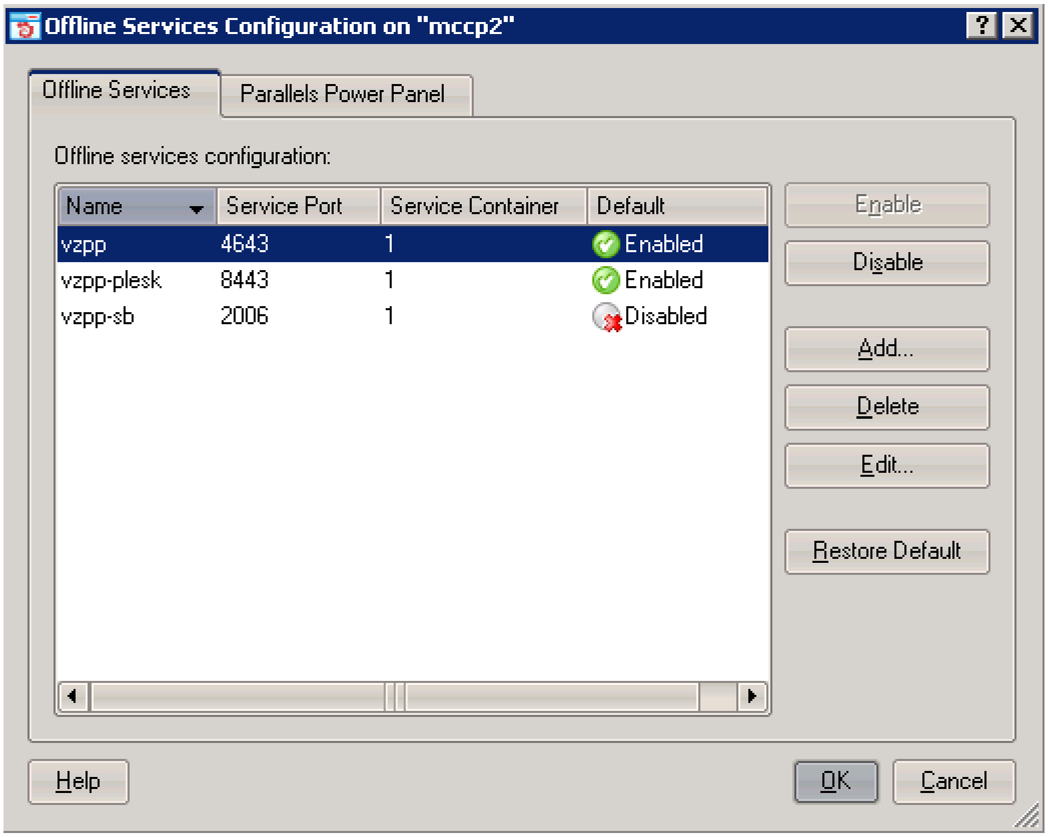
To leverage the full power of Parallels Management Console, it is important to be aware of those tasks that are much more convenient to perform through the Management Console interface than through the command line. The current chapter centers on the advanced Management Console features you can make use of while administering your Parallels Virtuozzo Containers system.

In This Chapter

Configuring Offline Management Parameters

The offline management functionality ensures the Container manageability by means of one or more offline services from any browser at its own IP address. When offline management is enabled for a Container, this Container is said to be subscribed to one or more offline services, which means that one or more ports of its IP address are permanently active whatever the Container state. This is needed to ensure the Container manageability in its down state.

The currently supported services are `vzpp` (for managing Containers by means of Parallels Power Panel) and `vzpp-plesk` (for managing Containers by means of the Plesk control panel integrated with Parallels Power Panel). You can view the names of accessible services on your Hardware Node in Parallels Management Console by right-clicking the needed Hardware Node name and choosing **Tasks** > **Manage Offline Services**.



Offline services configuration:

Name	Service Port	Service Container	Default
vzpp	4643	1	✓ Enabled
vzpp-plesk	8443	1	✓ Enabled
vzpp-sb	2006	1	Disabled

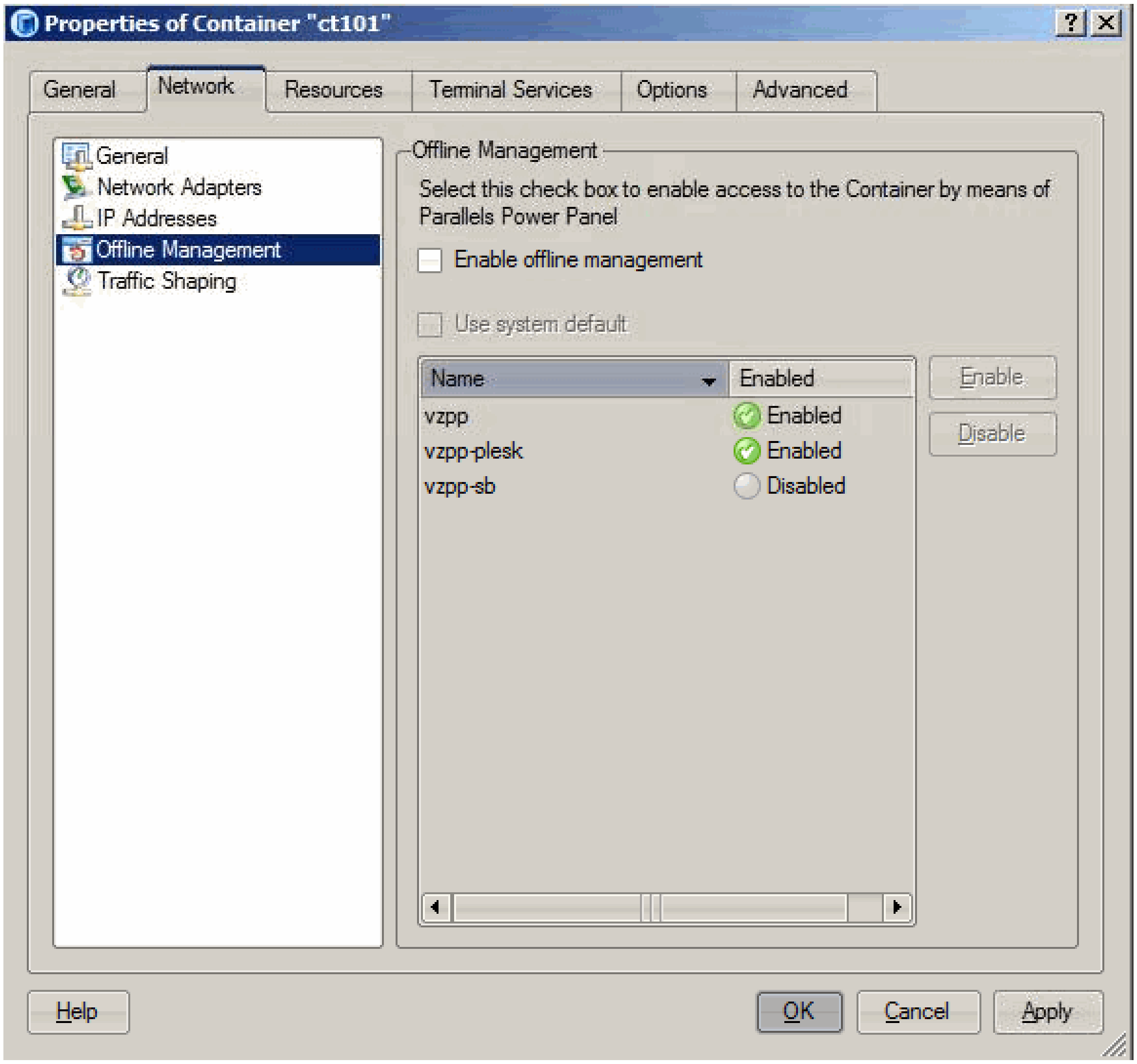
All offline services currently available on the Hardware Node are listed in the **Offline services configuration** table. By default, offline management is enabled for all Containers residing on the Node.

To disable the offline management for a Container, do the following:

1 In the left pane of the Management Console window, select the **Parallels Containers** item under the corresponding Hardware Node name.

2 In the right pane, right-click the Container on the Container list, and choose **Properties**.

3 On the **Network** tab of the displayed window, select the **Offline Management** item, and clear the **Enable offline management** check box.



Offline Management

Select this check box to enable access to the Container by means of Parallels Power Panel

□ Enable offline management

□ Use system default

Name	Enabled
vzpp	Enabled
vzpp-plesk	Enabled
vzpp-sb	Disabled

You can also manage the offline services which will be available to the Container. To do this:

a Leave the **Enable offline management** check box selected.

b Click the name of the corresponding offline service and use the **Enable**/**Disable** buttons to subscribe the Container to or unsubscribe it from this service.

If you have made some changes to any of the offline services and wish to restore the system default values, click the **Apply System Defaults** button at the bottom of the **Properties** window.

4 Click **OK**.

You can disable the offline management for all Containers residing on the Node at once:

1 Right-click the Hardware Node name, and choose **Tasks** > **Manage Offline Services**.

2 On the **Parallels Power Panel** tab of the **Offline Services Configuration** window, clear the **Enable Parallels Power Panel and Parallels Virtual Automation services** check box.

On the **Offline Services** tab, you can also manage the offline services which will be available to all Containers on the Hardware Node:

a Select the corresponding offline service from the list of available services and use the **Enable**/**Disable** buttons to enable/disable this offline service to the Containers on the Node.

b Use the **Add**/**Delete**/**Edit** buttons to add a new offline service, to remove an existing offline service, or to configure the properties of any offline service in the **Offline services configuration** table, respectively.

If you have made some changes to any of the offline services and wish to restore the system default values, click the **Restore Defaults** button.

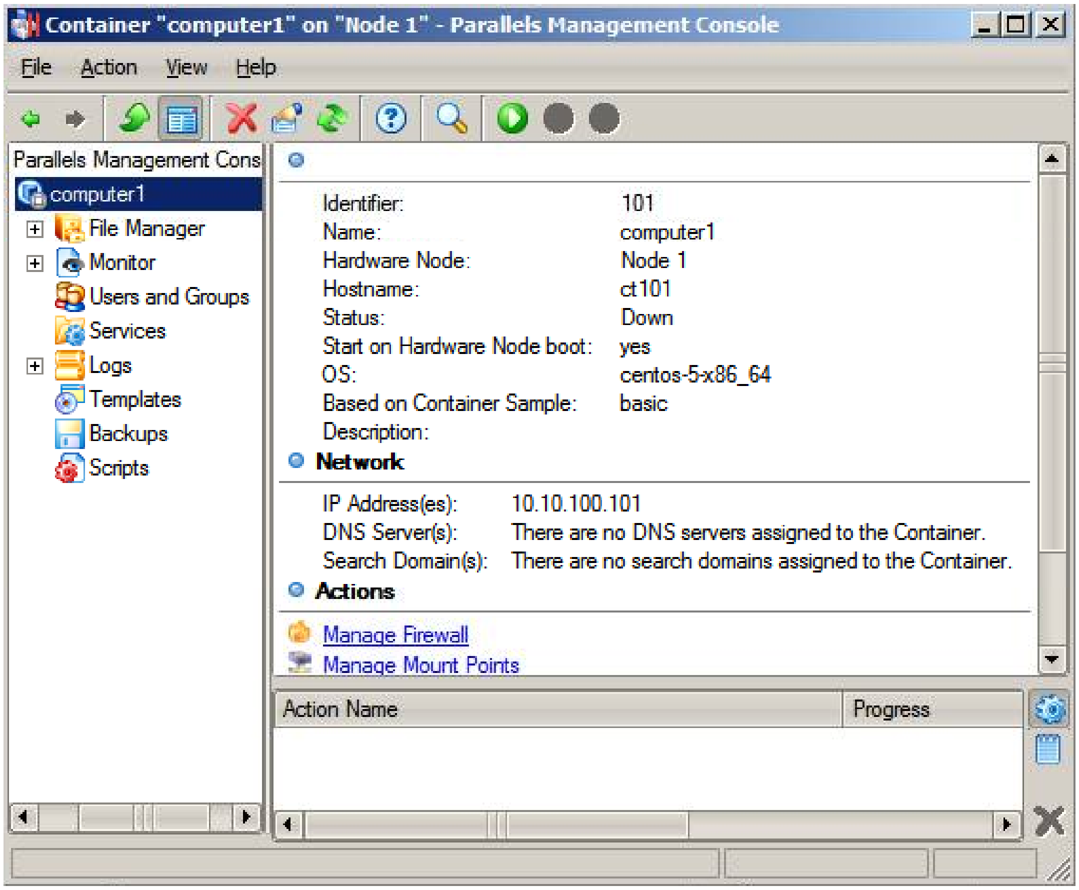
3 Click **OK**.

Viewing Summary Pages

You can view the summary page for every Hardware Node. Click on the name of the Hardware Node you are interested in in the tree in the left pane of the Parallels Management Console main window or double–click the name of the Hardware Node in the list of Nodes in the right pane.

The upper part of the information pane contains shortcuts to the most important tasks you are likely to do. However, all the actions and operations are accessible via the Management Console toolbar, **Action** menu and context menus. The bottom part of the Hardware Node summary page includes three tabs: **System**, **Network**, and **Disks**. The **System** tab describes the OS distribution and kernel version, CPU(s), RAM, and swap information. The **Network** tab describes the Hardware Node network configuration: interfaces and IP addresses. The **Disks** tab describes available disks and their utilization.

You can also view summary pages for each and every Container. To open the summary page in the Container Manager, click on the name of the Container in the tree pane. The summary page is similar to that in the main Parallels Management Console window.



Identifier: 101
Name: computer1
Hardware Node: Node 1
Hostname: ct101
Status: Down
Start on Hardware Node boot: yes
OS: centos-5-x86_64
Based on Container Sample: basic
Description:

Network

IP Address(es): 10.10.100.101
DNS Server(s): There are no DNS servers assigned to the Container.
Search Domain(s): There are no search domains assigned to the Container.

Actions

Manage Firewall
Manage Mount Points

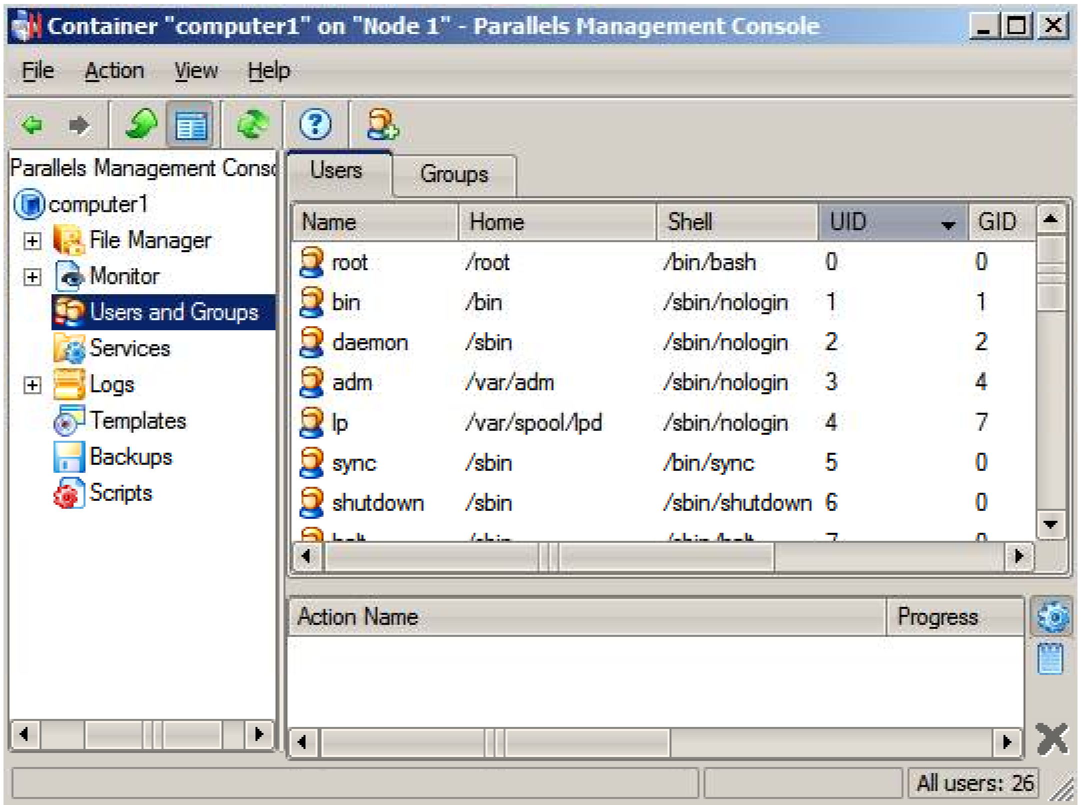
It contains information about the Container ID, type of the Container, OS template, status (e. g. 'mounted', 'running'), Container class, and hostname. There is also a **Network** section describing the network configuration of the Container.

The shortcuts to the most common operations are located at the bottom of the summary page, in the **Actions** section.

Managing Users and Groups In Containers

Parallels Management Console does not allow you to manage users or groups of the Host OS not to compromise the security of the Hardware Node. However, you can manage users and groups inside regular Containers with the help of Container Manager. All users and groups are adjustable. You can also add new users and groups.

To manage groups or users inside a Container, open the main tree for this Container, select the **Users and Groups** item, and click either the **Groups** or **Users** tab, respectively.



Name	Home	Shell	UID	GID
root	/root	/bin/bash	0	0
bin	/bin	/sbin/nologin	1	1
daemon	/sbin	/sbin/nologin	2	2
adm	/var/adm	/sbin/nologin	3	4
lp	/var/spool/lpd	/sbin/nologin	4	7
sync	/sbin	/bin/sync	5	0
shutdown	/sbin	/sbin/shutdown	6	0
halt	/sbin	/sbin/halt	7	0

All users: 26

To open the group properties dialog, double-click on the group name in the table of groups or select **Properties** on the context menu. To add a new user to the group, click the **Add** button. To remove a user from the group, select the user name and click the **Remove** button.

To add a new group, click the **New group** button on the toolbar (note that this button appears only if you are currently working with Container groups). Then enter the group name and press **OK**.

To delete a group, select its name in the table of groups and click the **Delete** button on the toolbar or select the **Delete** item from the context menu.

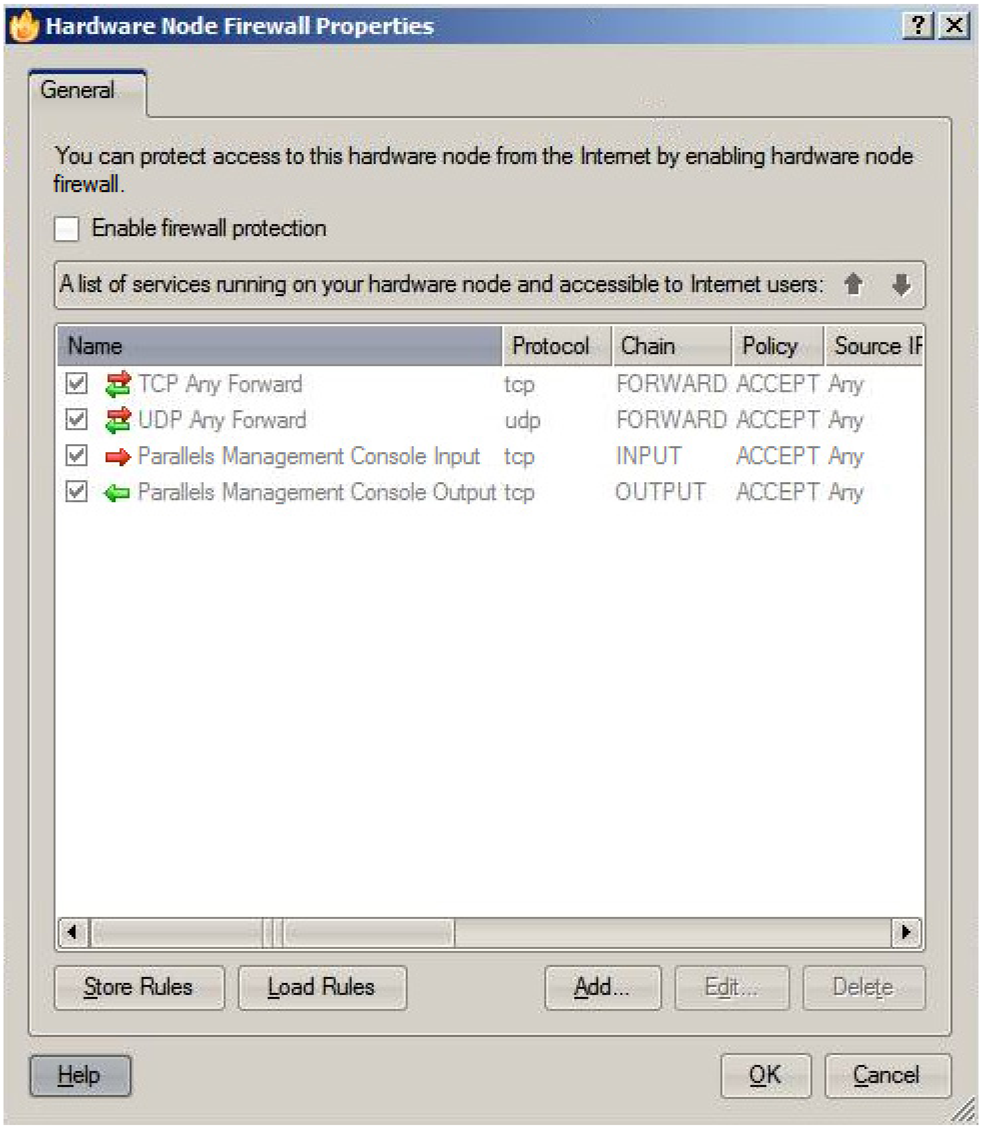
To add a new user, open the list of users and click the **New user** button at the top toolbar. Enter the user login (user name). This is the only mandatory parameter. You may also specify the home directory, the login shell, set the user description and password, add the user to one or more groups (see the **Member Of** tab). Then click **OK**.

To edit an existing user, double-click on the user name in the table of users or use the **Properties** item from the context menu. The user properties dialog is analogous to the **New User** dialog.

To delete a user, select its name in the table of users and click the **Delete** button at the top toolbar or select the **Delete** option in the context menu.

Configuring Firewall

You can limit access of Internet users to your Hardware Node. To enable the Hardware Node firewall, right-click the needed Node, and choose **Tasks** > **Manage Firewall Settings**.



You can protect access to this hardware node from the Internet by enabling hardware node firewall.

□ Enable firewall protection

A list of services running on your hardware node and accessible to Internet users:

	Name	Protocol	Chain	Policy	Source IF
☑	TCP Any Forward	tcp	FORWARD	ACCEPT	Any
☑	UDP Any Forward	udp	FORWARD	ACCEPT	Any
☑	Parallels Management Console Input	tcp	INPUT	ACCEPT	Any
☑	Parallels Management Console Output	tcp	OUTPUT	ACCEPT	Any

Several default rules are set for the Hardware Node, which are read-only. These rules are used to allow the Hardware Node to receive/send IP packets from/to different networks via TCP and UDP protocols and to enable Management Console connections to the Node.

In the **Hardware Node Firewall Properties** window, you can do the following:

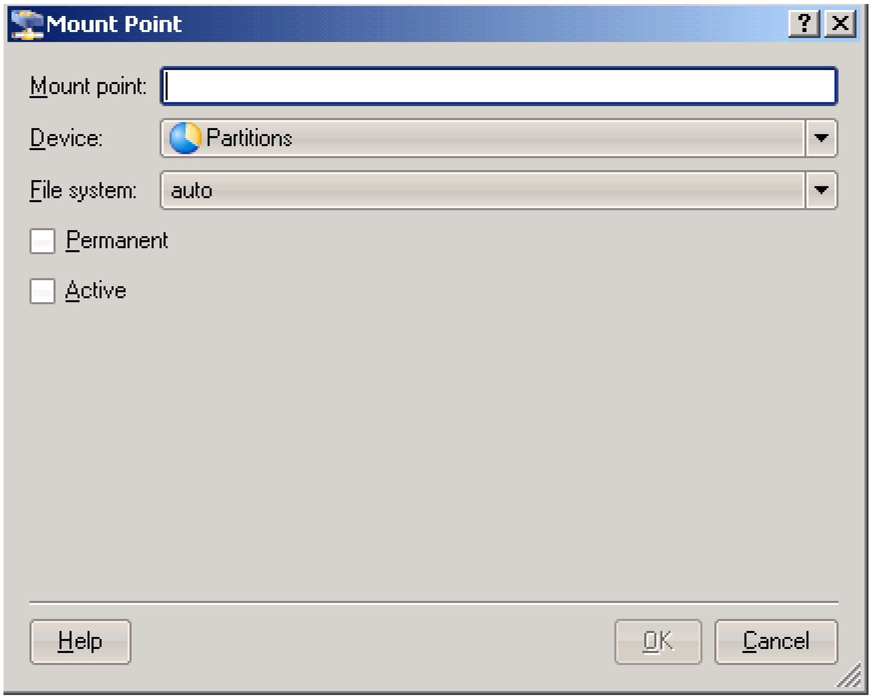
- Add your own rules with the **Add** button, for example, to provide access to certain services like SSH, Telnet, POP3, SMTP, HTTP, and FTP. You can also define rules that are more specific. Refer to your Linux documentation for more details on firewall configuration.
- Remove any rules (except for the default ones) form the existing list with the **Delete** button. To disable the rule temporary, unmark the check box opposite the rule name.
- Change any of the existing rules (except for the default ones) using the **Edit** button.
- Save any of the existing rules on your local computer with the **Store Rules** button or load new rules from a local file with the **Load Rules** button.

Managing the firewall configuration for a Container is identical to managing the firewall configuration for the Hardware Node in respect of adding or removing rules. To manage the firewall configuration for a Container, click the **Manage Firewall** link on the summary page of the Container Manager.

Each IP packet coming to a particular Container passes 2 firewalls: the `iptables` rules of the Host OS and the firewall rules of the given Container. An administrator of the Hardware Node sets up the Host OS `iptables` rules, and the end-users have no access to these rules.

Managing Mount Points

You can manage mount points through Parallels Management Console both for the Hardware Node and for each and every Container. To view the current list of mount points, click the **Manage Mounts** link on the summary page of either the Hardware Node or the necessary Container. Then use the **Add** button to add a new mount point, the **Remove From List** button to delete an existing mount point, or the **Edit** button to change an existing mount point. For example, after clicking the **Add** button, you will be presented with the following window.



Mount point:

Device: Partitions

File system: auto

□ Permanent

□ Active

In this window, do the following:

- Specify the directory where your file system is to be mounted the **Mount point** field (if the directory does not exist, it will be automatically created after clicking the **OK** button).
- Choose the physical device where your file system resides in the **Device** list box.

If you mark a mount point permanent (the **Permanent** check box is selected), it means that this mount point will be automatically mounted on the system boot. If you mark a mount point active (the **Active** check box is selected), it will be mounted after you click the **OK** button in the **Mount Point** window.

Viewing System and Parallels Virtuozzo Containers Logs

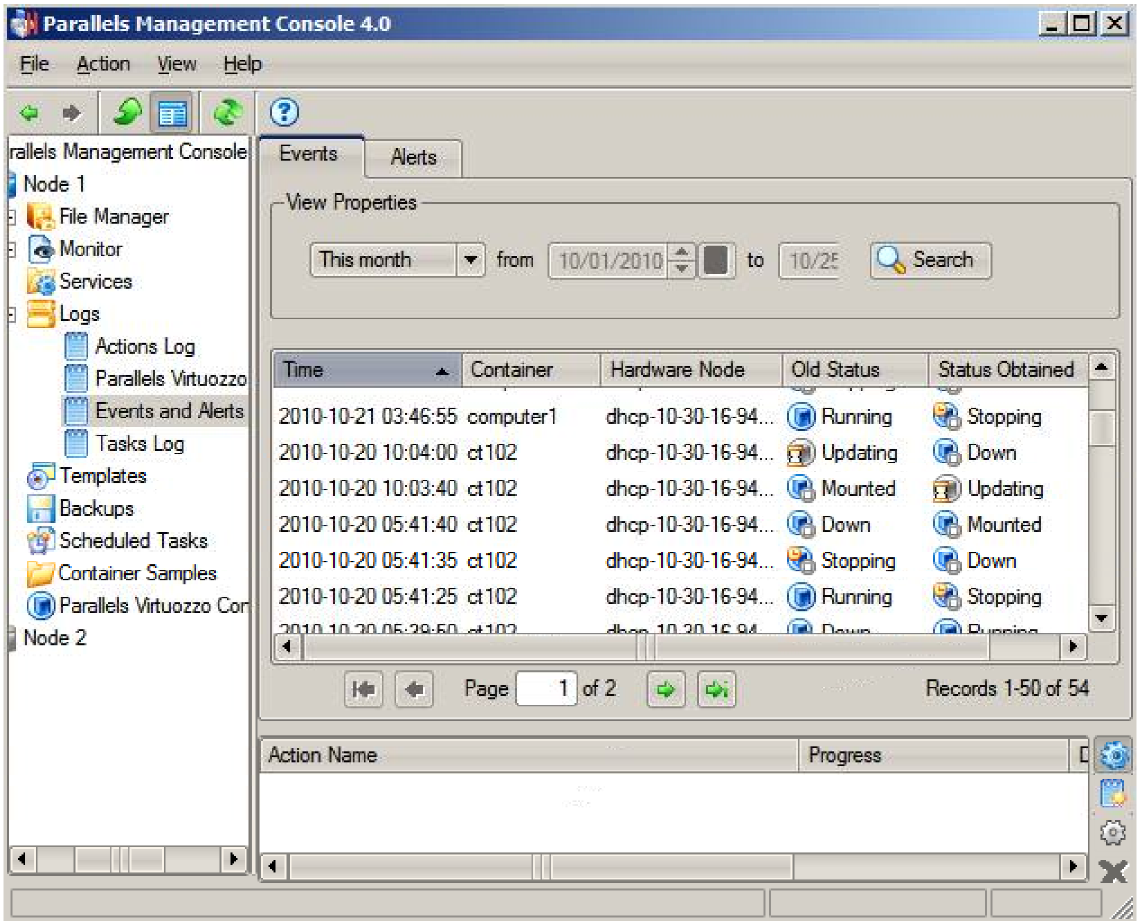
Parallels Management Console allows to view the logs which are maintained on the corresponding Hardware Node both for the Hardware Node itself and for a particular Container. The following log types are available for a particular Hardware Node in the Management Console main window:

Log type	**Description**
Alerts	Resource management system messages generated in case a Container exceeds its resources limits or disk quotas.
Events	All Container-related events (start, stop, migrate, mount, unmount, etc.).
Operations	Asynchronous tasks performed with any Container of the Hardware Node.
Parallels Virtuozzo Containers	Parallels Virtuozzo Containers system messages.
Actions	All actions performed with the main Parallels Virtuozzo Containers management utility `vzctl`: creating a new Container destroying an existing Container, starting and stopping a Container, running commands in a Container and adjusting the configuration parameters and limits for a Container.
Tasks	All tasks performed in the Hardware or Container context.

For Containers, only the **Events and Alerts** and **Tasks Log** logs are available in the corresponding Container manager window.

To view the logs, do the following:

1. Expand the **Logs** folder in the main tree under either the Hardware Node name or the Container name, and click the needed log type.
2. Specify the time period for which you would like to view the logs.
3. Click **Search** to display the list of log entries in the right pane of the window.



This month from 10/01/2010 to 10/25

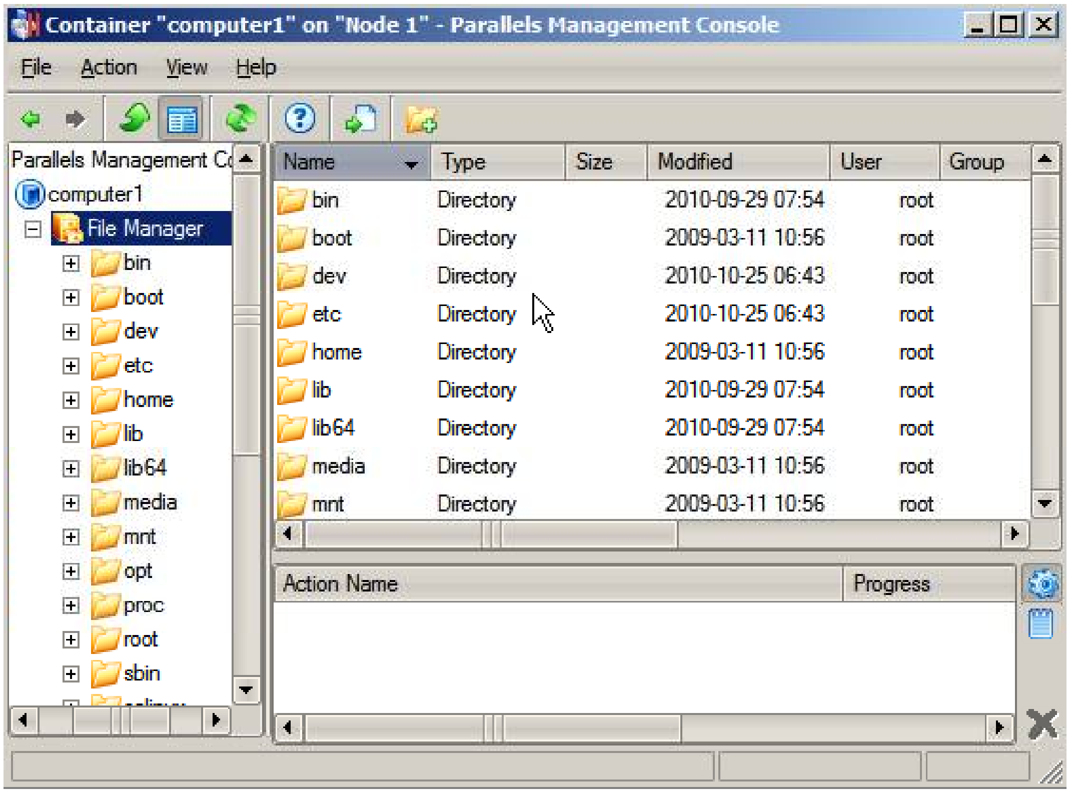
Time	Container	Hardware Node	Old Status	Status Obtained
2010-10-21 03:46:55	computer1	dhcp-10-30-16-94...	Running	Stopping
2010-10-20 10:04:00	ct102	dhcp-10-30-16-94...	Updating	Down
2010-10-20 10:03:40	ct102	dhcp-10-30-16-94...	Mounted	Updating
2010-10-20 05:41:40	ct102	dhcp-10-30-16-94...	Down	Mounted
2010-10-20 05:41:35	ct102	dhcp-10-30-16-94...	Stopping	Down
2010-10-20 05:41:25	ct102	dhcp-10-30-16-94...	Running	Stopping

Page 1 of 2 Records 1-50 of 54

Note: You can adjust the level of logging verbosity by defining the **log_level** parameter (from 0 to 2) in the global configuration file (adjustable by selecting the **Configuration** item in the Hardware Node main tree).

Managing Files In Containers

You cannot manage files directly on the Hardware Node by means of Parallels Management Console, but you can do it inside each and every Container by means of the Container manager window. After you click on the **File Manager** item in the Container main tree, you will see the list of folders and files of the Container root directory. Thus, this item corresponds to the / directory of the selected Container.



Name	Type	Size	Modified	User	Group
bin	Directory		2010-09-29 07:54	root	
boot	Directory		2009-03-11 10:56	root	
dev	Directory		2010-10-25 06:43	root	
etc	Directory		2010-10-25 06:43	root	
home	Directory		2009-03-11 10:56	root	
lib	Directory		2010-09-29 07:54	root	
lib64	Directory		2010-09-29 07:54	root	
media	Directory		2009-03-11 10:56	root	
mnt	Directory		2009-03-11 10:56	root	

The principles of working with the Container file manager are standard. You can move through the hierarchy of Container folders by double-clicking the folders names or selecting the necessary folders in the left pane. Use the menu items, toolbar buttons, table view, and context menus to perform the following tasks:

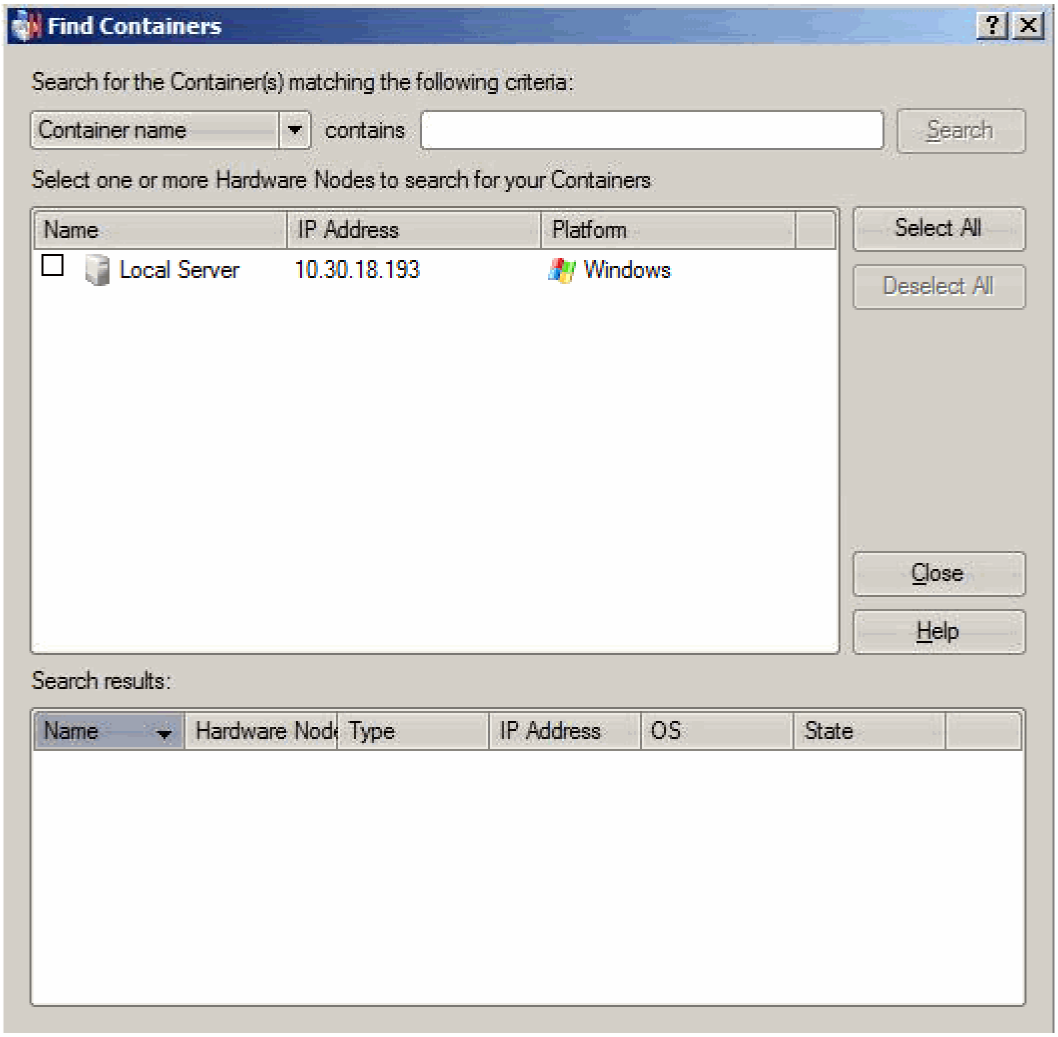
- View the contents of simple text files.
- View the principal information about a file/folder/symlink located in every directory and subdirectory of any depth in the given Container.
- Upload any number of files or whole directories from the local computer (the computer where Parallels Management Console is installed) to any folder of the given Container.
- Download any number of files from the given Container to the local computer.
- Create new folders in the Container.
- Copy files to another directory in the given Container.
- Move files to another directory in the given Container.
- Delete Container files.
- Rename Container files.
- Set permissions for Container files.

Parallels Management Console provides a user-intuitive interface for performing all these tasks.

Searching for Containers

Sometimes, there are a great number of Containers on your Hardware Nodes. To quickly find the necessary Container:

1 Right-click the **Parallels Containers** item, and choose **Task** > **Search for Containers**. The **Find Containers** window opens.



Search for the Container(s) matching the following criteria:

Container name ▾ contains

Select one or more Hardware Nodes to search for your Containers

Name	IP Address	Platform
□ Local Server	10.30.18.193	Windows

Search results:

Name	Hardware Nod	Type	IP Address	OS	State

2 Indicate the parameter by which you want to search for Containers on the upper left drop-down menu, and then the value of the parameter. If you choose to search for Containers by their state (status) or ID, you will be presented with a list of predefined values of these parameters. It is connected with the fact that there is a fixed number of Container statuses, and Container IDs can be only of the integer type. By searching for Containers by their name or IP address, you can enter any string in the corresponding field. In this case, the search results will display all the Containers whose name/IP address contain the specified string, even if only as a part.

3 Select the Hardware Nodes where you want to search for Containers with the specified characteristics. Containers from different Nodes matching the search criterion will be displayed in one and the same search result table. Once you have selected the Hardware Nodes, click the **Search** button. The table will be populated at the bottom of the window.

The Containers in the **Search Results** table corresponding to the specified search criterion may also be sorted by a number of parameters, among which are their ID, name, the Hardware Node they belong to, their IP address, etc. To sort the Containers by a parameter, click the corresponding column name. Another click will reverse the sorting order.

From the **Search Results** table, you may also open the Container manager window by double-clicking the corresponding Container.

Managing Container Search Domains

Search domains is the list for hostname lookup. The search list is normally determined by the local domain name; by default, it contains only the local domain name. You can add other host names for a particular Container. A search query is performed by attempting to use each item in the list in turn until a match is found. Note that this process may be slow and may generate a lot of network traffic if the servers for the listed domains are not local, and that the query might time out if no server is available for one of the domains. The search list is currently limited to six domains with a total of 256 characters.

To view and/or edit the list of search domains for a particular Container, do the following:

1 Click on the **Parallels Virtuozzo Containers** item in the Parallels Management Console main tree.

2 As soon as the list of the Containers on this particular Hardware Node is displayed, right-click on the necessary Container name and select **Properties** on the context menu. (In case you are working with the Container Manager, click on the **Manage Container Configuration** link at the Container dashboard).

3 Click the **Network** tab in the **Properties of Containers** window.

4 Under the **Search domains** group in the right part of the window, use the **Add**, **Remove**, and **Properties** buttons to add, delete, or edit search domains, respectively.

CHAPTER 11

Troubleshooting

This chapter provides the information about those problems that may occur during your work with Parallels Virtuozzo Containers and suggests the ways to solve them, including getting technical support from Parallels.

In This Chapter

General Considerations

The general issues to take into consideration when troubleshooting your system are listed below. You should read them carefully before trying to solve more specific problems.

- Make sure a valid license is always loaded on the Node. If your license has expired and the grace period is over, all the Containers on your Node will be stopped.
- You should always remember where you are currently located in your terminal. Check it periodically using the `pwd`, `hostname`, `ifconfig`, `cat /proc/vz/veinfo` commands. One and the same command executed inside a Container and on the Node can lead to very different results. You can also set up the `PS1` environment variable to show the full path in the `bash` prompt. To do this, add these lines to `/root/.bash_profile`:

```
PS1="[\u@\h \w]$ "
export PS1
```

- If the Node slows down, use `vmstat`, `ps` (`ps axfw`), `dmesg`, `top` (`vztop`) to find out what is happening, never reboot the machine without investigation. If no thinking helps restore the normal operation, use the Alt+SysRq sequences to dump the memory (`showMem`) and processes (`showPc`).
- If the Node was incorrectly brought down, on its next startup all the partitions will be checked and quota recalculated for each Container, which dramatically increases the startup time.
- Do not run any binary or script that belongs to a Container directly from the Node, for example, do not ever do that:

```
cd /vz/root/99/etc/init.d
./httpd status
```

Any script inside a Container could have been changed to whatever the Container owner chooses: it could have been trojaned, replaced to something like `rm -rf`, etc. You can use only `vzctl exec`/`vzctl enter` to execute programs inside a Container.

- Do not use init scripts on the Node. An init script may use `killall` to stop a service, which means that all similar processes will be killed in all Containers. You can check `/var/run/Service.pid` and kill the correspondent process explicitly.
- You must be able to detect any rootkit inside a Container. It is recommended to use the `chkrootkit` package for detection (you can download the latest version from www.chkrootkit.org), or at least run

```
rpm -Va|grep "S.5"
```

to check up if the MD5 sum has changed for any RPM file.

You can also run `nmap`, for example:

```
# nmap -p 1-65535 192.168.0.1

Starting nmap V. 2.54BETA22 ( www.insecure.org/nmap/ )
Interesting ports on  (192.168.0.1):
(The 65531 ports scanned but not shown below are in
  state: closed)
```

```
Port        State       Service
21/tcp      open        ftp
22/tcp      open        ssh
80/tcp      open        http
111/tcp     open        sunrpc

Nmap run completed -- 1 IP address (1 host up) scanned
  in 169 seconds
```

to check if any ports are open that should normally be closed.

That could however be a problem to remove a rootkit from a Container and make sure it is 100% removed. If you're not sure, create a new Container for that customer and migrate his/her sites and mail there.

- Check the `/var/log/` directory on the Node to find out what is happening on the system. There are a number of log files that are maintained by the system and Parallels Virtuozzo Containers (the `boot.log`, `messages`, etc.), but other services and programs may also put their own log files here depending on your distribution of Linux and the services and applications that you are running. For example, there may be logs associated with running a mail server (the `maillog` file), automatic tasks (the `cron` file), and others. However, the first place to look into when you are troubleshooting is the `/var/log/messages` log file. It contains the boot messages when the system came up as well as other status messages as the system runs. Errors with I/O, networking, and other general system errors are reported in this file. So, we recommend that you read to the `messages` log file first and then proceed with the other files from the `/var/log/` directory.
- Subscribe to bug tracking lists. You should keep track of new public DoS tools or remote exploits for the software and install them into Containers or at Nodes.
- When using `iptables`, there is a simple rule for Chains usage to help protect both the Node and its Containers:
  - use INPUT, OUTPUT to filter packets that come in/out the Node
  - use FORWARD to filter packets that are designated for Containers

Kernel Troubleshooting

Using ALT+SYSRQ Keyboard Sequences

Press ALT+SYSRQ+H (3 keys simultaneously) and check what is printed at the Node console, for example:

```
SysRq: unRaw Boot Sync Unmount showPc showTasks showMem loglevel0-8 tErm kIll killalL
Calls Oops
```

This output shows you what ALT+SYSRQ sequences you may use for performing this or that command. The capital letters in the command names identify the sequence. Thus, if there are any troubles with the machine and you're about to reboot it, please press the following sequences before pressing the **Power** button:

ALT+SYSRQ+M to dump memory info

ALT+SYSRQ+P to dump processes states

ALT+SYSRQ+S to sync disks

ALT+SYSRQ+U to unmount filesystems

ALT+SYSRQ+L to kill all processes

ALT+SYSRQ+U try to unmount once again

ALT+SYSRQ+B to reboot

If the server is not rebooted after that, you can press the **Power** button.

Saving Kernel Faults (OOPS)

You can use the following command to check for the kernel messages that should be reported to Parallels Virtuozzo Containers developers:

```
grep -E "Call Trace|Code" /var/log/messages*
```

Then, you should find kernel-related lines in the corresponding log file and figure out what kernel was booted when the oops occurred. Search backward for the "Linux" string, look for strings like that:

```
Sep 26 11:41:12 kernel: Linux version 2.6.18-8.1.1.el5.028stab043.1 (root@rhel5-32-
build) (gcc version 4.1.1 20061011 (Red Hat 4.1.1-30)) #1 SMP Wed Aug 29 11:51:58 MSK
2007
```

An oops usually starts with some description of what happened and ends with the Code string. Here is an example:

```
Aug 25 08:27:46 boar BUG: unable to handle kernel NULL pointer dereference at virtual
address 00000038
Aug 25 08:27:46 boar printing eip:
Aug 25 08:27:46 boar f0ce6507
Aug 25 08:27:46 boar *pde = 00003001
Aug 25 08:27:46 boar Oops: 0000 [#1]
Aug 25 08:27:46 boar SMP
Aug 25 08:27:46 boar last sysfs file:
Aug 25 08:27:46 boar Modules linked in: snapapi26(U) bridge(U) ip_vzredir(U) vzredir(U)
vzcompat(U) vzrst(U) i
p_nat(U) vzcpt(U) ip_conntrack(U) nfnetlink(U) vzfs(U) vzlinkdev(U) vzethdev(U)
vzevent(U) vzlist(U) vznet(U) vzstat(U) vzmo
n(U) xt_tcpudp(U) ip_vznetstat(U) vznetstat(U) iptable_mangle(U) iptable_filter(U)
ip_tables(U) vztable(U) vzdquota(U) vzdev(U) autofs4(U) hidp(U) rfcomm(U) l2cap(U)
bluetooth(U) sunrpc(U) ipv6(U) xt_length(U) ipt_ttl(U) xt_tcpmss(U) ipt_TCPMSS(U)
xt_multiport(U) xt_limit(U) ipt_tos(U) ipt_REJECT(U) x_tables(U) video(U) sbs(U)
i2c_ec(U) button(U) battery(U) asus_acpi(U) ac(U) lp(U) floppy(U) sg(U) pcspkr(U)
i2c_piix4(U) e100(U) parport_pc(U) i2c_core(U) parport(U) cpqphp(U) eepro100(U) mii(U)
serio_raw(U) ide_cd(U) cdrom(U) ahci(U) libata(U) dm_snapshot
(U) dm_zero(U) dm_mirror(U) dm_mod(U) megaraid(U) sym53c8xx(U) scsi_transport_spi(U)
sd_mod(U) scsi_mod(U) ext3(U) jbd(U) ehci_hcd(U) ohci_hcd(U) uhci_hcd(U)
Aug 25 08:27:46 boar CPU: 1, VCPU: -1.1
Aug 25 08:27:46 boar EIP: 0060:[<f0ce6507>] Tainted: P VLI
Aug 25 08:27:46 boar EFLAGS: 00010246 (2.6.18-028stab043.1-ent #1)
Aug 25 08:27:46 boar EIP is at clone_endio+0x29/0xc6 [dm_mod]
Aug 25 08:27:46 boar eax: 00000010   ebx: 00000001   ecx: 00000000   edx: 00000000
Aug 25 08:27:46 boar esi: 00000000   edi: b6f52920   ebp: c1a8dbc0   esp: 0b483e38
Aug 25 08:27:46 boar ds: 007b   es: 007b   ss: 0068
Aug 25 08:27:46 boar Process swapper (pid: 0, veid: 0, ti=0b482000 task=05e3f2b0
task.ti=0b482000)
Aug 25 08:27:46 boar Stack: 0b52caa0 00000001 00000000 b6f52920 00000000f0ce64de
00000000 02478825
Aug 25 08:27:46 boar 00000000 c18a8620 b6f52920 271e1a8c 024ca03800000000 00000000
00000000
Aug 25 08:27:46 boar 00000000 00000000 c18a3c00 00000202 c189e89400000006 00000000
05cb7200
Aug 25 08:27:46 boar Call Trace:
Aug 25 08:27:46 boar [<f0ce64de>] clone_endio+0x0/0xc6 [dm_mod]
Aug 25 08:27:46 boar [<02478825>] bio_endio+0x50/0x55
Aug 25 08:27:46 boar [<024ca038>] __end_that_request_first+0x185/0x47c
Aug 25 08:27:46 boar [<f0c711eb>] scsi_end_request+0x1a/0xa9 [scsi_mod]
Aug 25 08:27:46 boar [<02458f04>] mempool_free+0x5f/0x63
Aug 25 08:27:46 boar
Aug 25 08:27:46 boar [<f0c713c3>] scsi_io_completion+0x149/0x2f3 [scsi_mod]
Aug 25 08:27:46 boar [<f0c333b9>] sd_rw_intr+0x1f1/0x21b [sd_mod]
Aug 25 08:27:46 boar [<f0c6d3b9>] scsi_finish_command+0x73/0x77 [scsi_mod]
Aug 25 08:27:46 boar [<024cbfa2>] blk_done_softirq+0x4d/0x58
Aug 25 08:27:46 boar [<02426452>] __do_softirq+0x84/0x109
Aug 25 08:27:46 boar [<0242650d>] do_softirq+0x36/0x3a
Aug 25 08:27:46 boar [<024050b7>] do_IRQ+0xad/0xb6
Aug 25 08:27:46 boar [<024023fa>] default_idle+0x0/0x59
Aug 25 08:27:46 boar [<0240242b>] default_idle+0x31/0x59
Aug 25 08:27:46 boar [<024024b1>] cpu_idle+0x5e/0x74
Aug 25 08:27:46 boar ======================
Aug 25 08:27:46 boar Code: 5d c3 55 57 89 c7 56 89 ce 53 bb 01 00 00 00 83 ec 0c 8b 68
3c 83 7f 20 00 8b 45 00 8b 00 89 44 24 04 8b 45 04 89 04 24 8b 40 04 <8b> 40 28 89 44
24 08 0f 85 86 00 00 00 f6 47 10 01 75 0a 85 c9
Aug 25 08:27:46 boar EIP: [<f0ce6507>] clone_endio+0x29/0xc6 [dm_mod]
SS:ESP0068:0b483e38
Aug 25 08:27:46 boar Kernel panic - not syncing: Fatal exception in interrupt
```

All you need is to put the oops into a file and then send this file as part of your problem report to the Parallels support team.

Finding a Kernel Function That Caused the D Process State

If there are too many processes in the D state and you can't find out what is happening, issue the following command:

```
# objdump -Dr /boot/vmlinux-`uname -r` >/tmp/kernel.dump
```

and then get the process list:

```
# ps axfwln
  F UID   PID  PPID PRI NI  VSZ  RSS  WCHAN STAT TTY TIME COMMAND
100   0 20418 20417  17  0 2588  684      - R    ?   0:00 ps axfwln
100   0     1     0   8  0 1388  524 145186 S    ?   0:00 init
040   0  8670     1   9  0 1448  960 145186 S    ?   0:00 syslogd -m 0
040   0  8713     1  10  0 1616 1140 11ea02 S    ?   0:00 crond
```

Look for a number under the **WCHAN** column for the process in question. Then, open `/tmp/kernel.dump` in an editor, find that number in the first column and then scroll backward to the first function name, which can look like this:

```
"c011e910 <sys_nanosleep>:"
```

Then you can tell if the process “lives” or is blocked into the found function.

Problems With Container Management

This section includes recommendations on how to settle some problems with Containers.

Failure to Start a Container

An attempt to start a Container fails.

Solution 1

If there is a message on the system console: `parameters missing`, and the list of missed parameters follows the message, set these parameters using the `vzctl set --save` command (see **Configuring Container** (p. 35) for instructions). Try to start the Container once again.

Solution 2

If there is a message on the system console: `IP address is already used`, issue the `cat /proc/vz/veinfo` command. The information about the Container numeric identifier, Container class, number of Container's processes and Container IP address shall be displayed for each running Container. This shall also demonstrate that your Container is up, i.e. it must be running without any IP address assigned. Set its IP address using the command:

```
vzctl set CT_ID --ipadd IP_addr --save
```

where *`CT_ID`* represents the Container numeric identifier and *IP_addr* represents an actual IP address.

Solution 3

The Container might have used all its disk quota (either disk space or disk inodes). Check the Container disk quota and increase the quota parameters if needed. See the **Managing Disk Quotas** (p. 146) and **Setting Up Per-Container Disk Quota Parameters** (p. 152) sections for details.

Solution 4

Run the `vzfsutil` utility to make sure that the VZFS symlinks inside the Container work correctly. For example:

```
vzfsutil --call -t /vz/template /vz/private/<CT_ID>
```

The complete reference on the `vzfsutil` utility is provided in the *Parallels Virtuozzo Containers 4.7 Reference Guide*.

Solution 5

The Container administrator might have inadvertently modified, replaced, or deleted any file that is part of an application or OS template, which has brought about the Container malfunction. In this case, restore the file(s) with the `vzctl recover` command. See the **Reinstalling Containers** section (p. 104) for details.

Solution 6

Restore the latest operable copy of the Container by means of the `vzarestore` utility. See the **Backing Up and Restoring Containers** section (p. 61) for details.

Failure to Access a Container From Network

Solution 1

The IP address assigned to the Container might be already in use in your network. Make sure it is not. The problem Container address can be checked by issuing the following command:

```
# grep IP_ADDRESS /etc/vz/conf/<CT_ID>.conf
IP_ADDRESS="10.0.186.101"
```

The IP addresses of other Containers, which are running, can be checked by running

```
cat /proc/vz/veinfo
```

Solution 2

Make sure the routing to the Container is properly configured. Containers can use the default router for your network, or you may configure the Node as rooter for its Containers.

Failure to Log In to a Container

The Container starts successfully, but you cannot log in.

Solution 1

You are trying to connect via SSH, but access is denied. Probably you have not set the password of the `root` user yet or there is no such user. In this case, use the `vzctl set --userpasswd` command. For example, for Container 101 you might issue the following command:

```
# vzctl set 101 --userpasswd root:secret
```

Solution 2

Check forwarding settings by issuing the following command:

```
# cat /proc/sys/ipv4/conf/venet0/forwarding
```

If it is 0 then change it to 1 by issuing the following command:

```
# echo 1 > /proc/sys/ipv4/conf/venet0/forwarding
```

Failure to Back Up a Container in Parallels Management Console

An attempt to back up a Container with a large amount of disk space (e.g. 6 Gb) by means of Parallels Management Console finishes with the following error message: `The request was timed out`. However, the backup process continues running and the Container backup is successfully created on the Backup Node after a while, which can be checked by exploring the `/vz/backup` directory on this Node, where all Container backups are stored by default.

Solution

The problem is caused by the fact that the timeout limit set by Parallels Agent for the Container backup process in Management Console has been reached. This limit is equal to 3600 seconds by default. You can increase the maximal backup timeout value by performing the following operations:

1. In Management Console, right-click on the Hardware Node name and select **Tasks** > **Manage Parallels Agent Configuration** on the context menu.
2. In the left part of the displayed window, choose **backm** > **configuration** > **timeouts**.
3. Double-click the `backup` parameter in the right part of the **Parallels Agent Configuration** window, and specify the needed time (in seconds) in the **Parameter value** field.
4. Click **OK**.

Failure to Display the List of Container Backups

You created a number of Container backups on the Backup Node and now wish to view them. However, the process of displaying your Container backups takes a very long time or even goes into infinity.

Solution

By default, the timeout limit for the Container backup search process is set to a very high value - 3600 seconds, which makes the search process to run for 60 minutes before showing a list of available backups on the Backup Node. To reduce the time needed to display your Container backup list, you should decrease the backup search value. You can do it in the following way:

1. In Parallels Management Console, right-click on the Hardware Node name and select **Tasks** > **Manage Parallels Agent Configuration** on the context menu.
2. In the left part of the displayed window, choose **backm** > **configuration** > **timeouts**.
3. Double-click the `search` parameter in the right part of the **Parallels Agent Configuration** window and specify the desired time (in seconds) in the **Parameter value** field.

   Note: You are recommended to set the value of the `search` parameter to 300 seconds.
4. Click **OK**.

Miscellaneous Problems

Corrupted Pseudographics in Parallels Virtuozzo Containers Utilities

Some Parallels Virtuozzo Containers utilities (for example, `install`, `vzup2date`, and others) employ pseudographical instead of simple character output during their operation. Certain terminal clients fail to display the pseudographics the way it was intended to be displayed. This has nothing to do with Parallels Virtuozzo Containers, but with locale settings either on the Hardware Node or in the terminal client. You may try to solve this problem in one of the following ways:

Solution 1

Set the correct locale for your terminal.

Solution 2

Try to run the utility as

```
# LC_ALL=C utility_name
```

Solution 3

If you are connecting to the Node via a remote shell, please make sure the locale set in the remote terminal is the same as in the local one.

Timeout When Accessing Remote Hosts

A host is unreachable by the Hardware Node or its Containers, though it can be reached from other computers.

Solution

Often these timeouts occur due to the fact that the Explicit Congestion Notification (ECN) mechanism of the TCP/IP protocol is on by default in Parallels Virtuozzo Containers and off in some other systems, which leads to their incompatibility. ECN is used to avoid unnecessary packet drops and for some other enhancements. If Parallels Virtuozzo Containers cannot connect to a host, turn off this mechanism:

```
# sysctl -w net.ipv4.tcp_ecn=0
net.ipv4.tcp_ecn = 0
```

Failure to Start iptables Modules After Physical Server Migration

`iptables` is broken in the Container after a physical server has been migrated.

Solution

The `iptables` service can work properly inside the Container that has resulted from a physical server migration only if the `ipt_state` module is loaded both on the Hardware Node and in the Container in question. The simplest way to do it is the following:

1 Stop Parallels Virtuozzo Containers on the Node:

```
# service vz stop
```

2 Add `ipt_state` as another module name to the `IPTABLES_MODULES` parameter in the `/etc/sysconfig/iptables-config` file on the Node.

3 Restart `iptables` on the Node:

```
service iptables restart
```

4 Start Parallels Virtuozzo Containers:

```
# service vz start
```

5 Add `ipt_state` as another module name to the `IPTABLES` parameter in the `/etc/vz/vz.conf` file on the Node.

6 Restart the Container:

```
#  vzctl restart CT_ID
```

To learn more on loading `iptables` modules, see the **Loading iptables Modules** section (p. 317).

Getting Technical Support

This section provides information on how to get technical support from Parallels.

Getting Assistance With Parallels Virtuozzo Containers Installation

Parallels provides installation assistance for the Parallels Virtuozzo Containers software. Assistance with installation can be offered via e-mail or by using the *Parallels Virtuozzo Containers Support Tunnel* tool:

- While communicating via e-mail, the Parallels support will attempt to answer any relevant questions you may have before the installation process is initiated. This includes the following:
  - pre-requisites list
  - hardware compatibility
  - software compatibility
- You can also install the *Parallels Virtuozzo Containers Support Tunnel* tool on your physical server and use it for getting installation assistance from the Parallels support. Detailed information on the *Parallels Virtuozzo Containers Support Tunnel* tool is provided in **Establishing a Secure Channel to Parallels Support** (p. 364).

Preparing and Sending Questions to Technical Support

In most cases, the support team must rely on the customer's observations and communications with the customer to diagnose and solve the problem. Therefore, the detailed problem report is extremely important. You can submit a support report by visiting the http://www.parallels.com/en/support/virtuozzo/request/ web page and filling in the Online Support Form. When describing the problem, please do mention the following:

- symptoms of the problem
- when the problem began including the circumstances of the failure
- any changes you made to your system
- other information that may be relevant to your situation (e.g., the installation method)
- specific hardware devices that may be relevant to your problem

You can also make use of the Parallels Helpdesk support tool. To do this:

1. Follow the https://helpdesk.parallels.com/ link.
2. Register with the Parallels Helpdesk (if you have not done so before) by clicking the **Get Access to Parallels RT** link on the Helpdesk login page and following the instructions provided on the **Activate Your Support Account** screen.
3. Log in to the Helpdesk using the received credentials.
4. At the top of the **RT At Glance** screen, select the Parallels Virtuozzo Containers component your problem relates to on the drop-down menu, and click the **New Ticket in** button:
5. On the **Create New Ticket** screen, fill in the appropriate fields, describe your problem, and click the **Create** button to make a new support ticket.

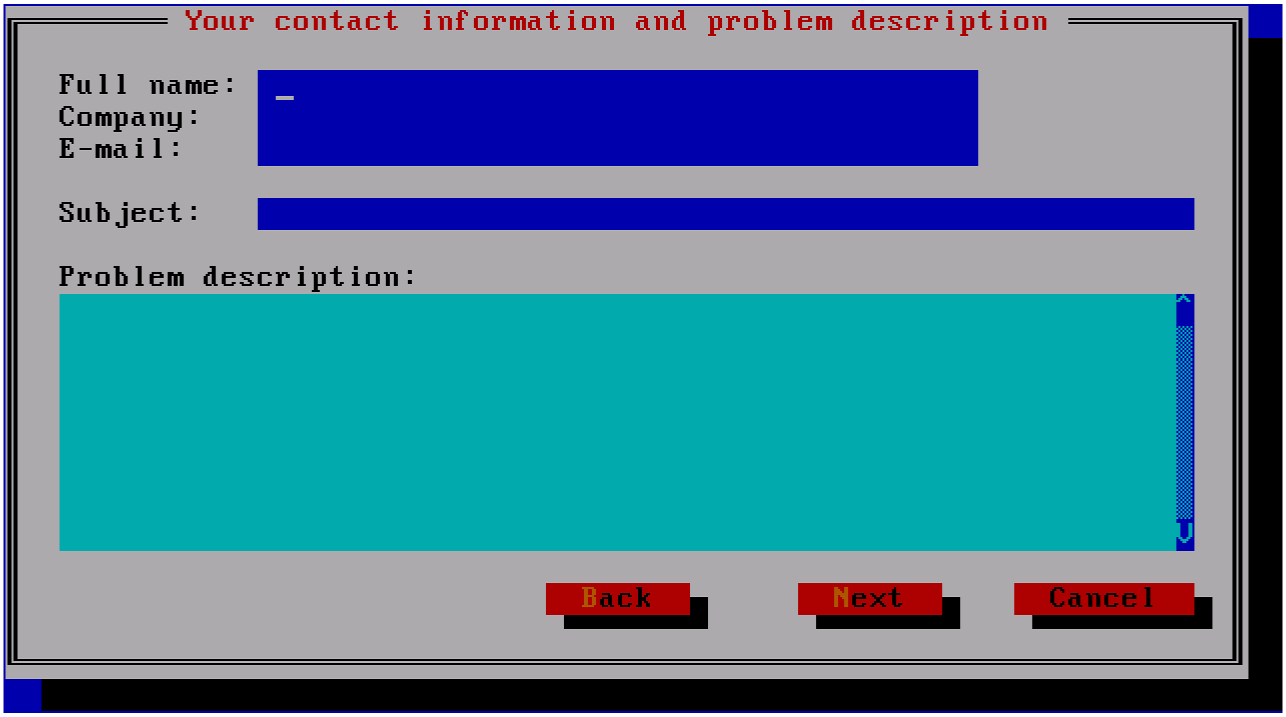
Another way of getting help is to directly call us or visit one of our offices. The information about phone numbers, contact people and office addresses is available on the contact pages at http://www.parallels.com/en/contact and http://www.parallels.com/en/support/phone/.

Submitting a Problem Report to Technical Support

Parallels Virtuozzo Containers is shipped with the `vzreport` utility allowing you to compile a detailed report if you have any Parallels Virtuozzo Containers-related problems and to automatically send it to the Parallels support team. After receiving your report, the support team will closely examine your problem and make its best to solve the problem as quickly as possible.

`vzreport` has two modes of execution—full screen and command line. By default, the utility starts in the full screen mode. However, you can force the utility to run in the command line mode by specifying any option containing your contact information (e.g., `-n` denoting your name) or the problem report description (e.g., `-m` used to provide additional information on your problem). Detailed information on all the options that can be passed to `vzreport` in the command line is provided in the *Parallels Virtuozzo Containers 4.7 Reference Guide*.

After running the `vzreport` utility in the full screen mode, the **Problem Report Wizard** is opened, which will guide you through a number of steps asking you to provide the necessary information to generate a problem report. In the **Welcome** window, just click **Next** to proceed with the wizard. You will see the following window:


Your contact information and problem description

Full name: _
Company:
E-mail:

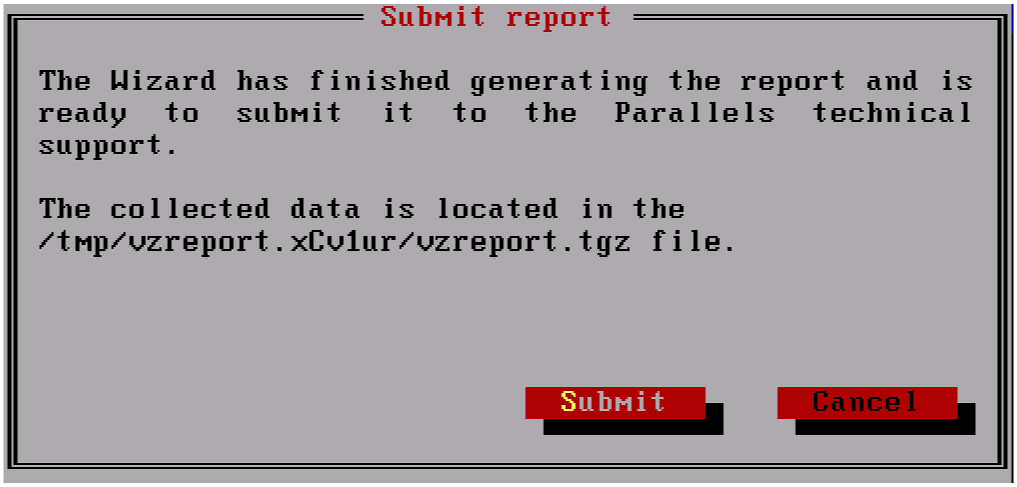
Subject:

Problem description:

In this window, enter your name, e-mail, and the name of your company into the corresponding fields. Make sure that you type a valid e-mail address; otherwise, the Parallels support team will not be able to contact you. In the **Subject** field, you should also specify what Parallels Virtuozzo Containers problem you encountered and may provide additional information in the **Problem description** field which, in your opinion, can help solve the problem.

Clicking **Next** in the **Your contact information and issue description** window starts collecting Parallels Virtuozzo Containers logs and information on your system and network settings into a special file. This file will be sent to the Parallels support team upon the completion of the wizard. The file does not contain any private information!

After the utility has gathered all the necessary information on the Node, the **Submit report** window is displayed.



The Wizard has finished generating the report and is ready to submit it to the Parallels technical support.

The collected data is located in the /tmp/vzreport.xCv1ur/vzreport.tgz file.

In this window, do one of the following:

- Click the **Submit** button to send your problem report to the Parallels technical support team. The report is dispatched directly to Parallels by using the HTTP protocol and port 80. However, if you use an HTTP proxy server for handling all your HTTP requests and wish your problem report to be sent via this server, you should specify the hostname or IP address of the server in the `/etc/vz/vz.conf` configuration file on the Hardware Node as the value of the `HTTP_PROXY` parameter. After the problem report has been successfully sent to the Parallels support, the **Congratulations** window is displayed informing you:
  - Of the ID assigned to your report. Use this ID every time you communicate with the Parallels support via e-mail or the Parallels Helpdesk support tool.
  - That an e-mail message providing you with detailed information on your problem report has been sent to the e-mail address you specified in the **E-mail** field of the **Your contact information and issue description** window.
- Click the **Cancel** button if you do not wish to dispatch the problem report to the support team at the moment for some reason or other. You can do it later on by manually sending the generated zip file to the Parallels support team. The full path to this file is indicated in the **Submit report** window.

Establishing a Secure Channel to Parallels Support

Parallels Virtuozzo Containers provides a special tool—*Parallels Virtuozzo Containers Support Tunnel*—which allows you to establish a private secure channel to the Parallels support team server. After establishing the channel, the support team will be able to quickly and securely connect to your Node and diagnose and solve your problem. The secure connection to your server is achieved through a Virtual Private Network (VPN) created between the Parallels support team server and your Hardware Node.

To start using the *Parallels Virtuozzo Containers Support Tunnel* tool:

- Make sure the `openvpn` (version 2.0 and above) and `vzvpn` packages are installed on your Node. These packages are automatically installed on the Node during the Parallels Virtuozzo Containers installation.
- Make sure that port 80 is opened on the Hardware Node.
- Edit the `/etc/vzvpn/vzvpn.conf` file to specify the correct parameters for your proxy server, if you use any. Detailed information on these parameters is given in the *Parallels Virtuozzo Containers 4.7 Reference Guide*.

After you have completed the tasks above and in case you encountered a Parallels Virtuozzo Containers-related problem, you can do the following to get assistance from the Parallels support:

1. Obtain a special certificate from Parallels which will uniquely identify you as a Parallels Virtuozzo Containers user. Certificates are issued by Parallels in the form of files. Once you obtain a certificate, install it on the Node using the `vzvpn.sh key-install` *`certificate`* command where *`certificate`* denotes the name of the certificate file obtained from Parallels. You can get a certificate in one of the following ways:
   - Visit the http://www.parallels.com/en/support/virtuozzo/certificates web site, fill up the **Request Parallels Virtuozzo Containers Support Certificate** form, and click the **Submit** button. After a while, a certificate will be generated and sent to the email address you provided in the **Request Parallels Virtuozzo Containers Support Certificate** form.
   - Contact the Parallels support team via email or by telephone and ask for a valid certificate.
2. After installing the certificate, make sure your Hardware Node is connected to the Internet.
3. On the Node, execute the `/etc/init.d/vzvpn.sh start` command to establish a VPN between your Node and the Parallels support server.
4. Contact the Parallels support team (by telephone or via e-mail) and inform them of the problem you encountered. Also mention that you have launched the Parallels Virtuozzo Containers Support Tunnel tool and established a VPN to the Parallels support server.
5. After that, the Parallels support team will connect to your Node by using the secure VPN established, closely examine your problem, and make its best to solve the problem as quickly as possible.

Notes:

1. Parallels Virtuozzo Containers Support Tunnel is implemented as a standard Linux service running in the background of your system. Therefore, to have this service running after your Hardware Node reboot, you should set it to the `autoboot` mode or start it manually again by executing the `/etc/init.d/vzvpn start` command.

2. To close the VPN session with the Parallels support server, you should issue the `/etc/init.d/vzvpn stop` command on the Node.

Setting Up the Monitor Node

A regular monitoring of Hardware Nodes is an important part of their maintaining, administering, and troubleshooting. Parallels Virtuozzo Containers enables you to check the state of your Nodes in one of the following ways:

- By using the Monitor Node as a serial console to log the kernel state of the Hardware Node. This way of logging kernel messages is the most preferable one since it allows you to start monitoring the system and collecting messages right after the kernel boot process is started.
- By running the `vzrmond` daemon on the Monitor Node. This daemon provides the remote monitoring of the Hardware Node by constantly checking up the current state of the Node, verifying that the main Hardware Node parameters do not exceed their specified limits, and sending instant alerts via e-mail, ICQ, or SMS if anything goes wrong on the Node.
- By running the `vzstatrep` utility on the Monitor Node. This utility periodically analyzes the main resources consumption of one or several Hardware Nodes, generates statistic reports and graphics based on the analyzed information, and sends these reports and graphics at your e-mail address. You can then examine the received e-mail message to find out whether the Hardware Node is functioning trouble-free or a number of corrective actions should be performed in relation to some of its components.
- By using the `netconsole` module. This module can be configured to send console messages from the Parallels Virtuozzo Containers kernel on the Hardware Node to the Monitor Node. However, in this case the process of monitoring the system and collecting kernel messages is started only after the kernel has been successfully loaded on the Hardware Node.

The following subsections describe each of these ways in detail.

Configuring a Serial Console on the Monitor Node

To set up a serial console on the Monitor Node, you have to complete the following tasks:

- Install Linux on a dedicated server that is to be served as the Monitor Node. This server shall meet one requirement: you must be able to install a Linux distribution on it. Logging messages even from several Hardware Nodes requires neither a powerful CPU nor a large amount of RAM. However, if you plan to be connected to more than two Hardware Nodes, you may need a special multi-port serial card. Among the popular makes of multi-port serial cards are Cyclades-Z, Digiboard, Specialix, and Stallion. Consult your Linux distribution vendor on multi-port serial card compatibility issues.
- Connect the Hardware Nodes to the Monitor Node via a null-modem cable.
- Configure serial parameters on the Monitor Node and the Hardware Node.
- Configure the Hardware Node to send kernel messages to the Monitor Node.
- Start the message collector on the Monitor Node.
- Reboot the Hardware Node.

Configuring Serial Parameters on the Monitor and Hardware Nodes

First, find out the serial port number used on the Monitor Node. The first serial port (COM1 in DOS) is represented by `/dev/ttyS0`, the second one (COM2 in DOS) – by `/dev/ttyS1`, and so on. If you are not sure about which serial port the cable is connected to, you may try on your own risk different ports in the commands given in this and next subsections. It may not be completely safe if you have some other hardware attached to a different serial port.

If you have the null-modem cable connected to the `/dev/ttyS1` port, issue the following command on the Monitor Node:

```
# stty 115200 cs8 -hupcl -cstopb cread clocal -crtscts -icrnl ixon \
       ixoff -opost -isig -icanon -iexten -echo \
       </dev/ttyS1 >/dev/ttyS1
```

This command will correctly configure the second serial port (`/dev/ttyS1`). Use the appropriate serial terminal name instead of `/dev/ttyS1` if the actual configuration differs.

Start the following command on the Monitor Node:

```
# cat /dev/ttyS1
```

Now find out which serial port is connected on the Hardware Node side. Issue the following commands to configure the serial line parameters on the Hardware Node and to send a message to the Monitor Node:

```
# stty 115200 cs8 -hupcl -cstopb cread clocal -crtscts ixon ixoff \
       -opost </dev/ttyS0 >/dev/ttyS0
# echo 123 > /dev/ttyS0
```

The commands above assume that `/dev/ttyS1` is used on the Monitor Node and `/dev/ttyS0` is used on the Hardware Node. Change the commands appropriately if the actual configuration differs.

If you did everything right, you shall see “123” on the Monitor Node now.

Preparing the Hardware Node for Sending Messages

Now you should pass the `console=ttyS0,115200 console=tty` parameters to the kernel on each start of the Hardware Node. In case you are using the LILO boot loader, add the following line into the Parallels Virtuozzo Containers section of the `/etc/lilo.conf` configuration file:

```
append="console=ttyS0,115200 console=tty"
```

and run `/sbin/lilo` to activate the changes.

With the GRUB loader, it is enough to modify the `/boot/grub/grub.conf` configuration file by adding the needed parameters to the line beginning with `kernel` inside the Virtuozzo section of the file. For example:

```
kernel /vmlinuz-2.6.32-stab1.2.777 ro console=ttyS0,115200 console=tty
```

> **Note:** You must not remove any of the existing parameters in the kernel line of the `grub.conf` configuration file.

Parallels Virtuozzo Containers includes a special watchdog module, which is off by default. However, if you set up a Monitor Node, it is very important to have this module running since it logs the kernel state every minute. In order to make Parallels Virtuozzo Containers load this module automatically, edit the `/etc/vz/vz.conf` file and change the value of the `VZWDOG` parameter from `no` to `yes`. The corresponding line should look like the following:

```
# grep ^VZWDOG /etc/vz/vz.conf
VZWDOG=yes
```

Starting Messages Collection on the Monitor Node

The kernel messages from the Hardware Node may be collected by reading from the serial terminal on the Monitor Node. The simplest way to collect and to store them is by executing the following command:

```
# cat /dev/ttyS1 > /var/log/vzmessages.hn1 &
```

on the Monitor Node. This way the messages will be stored in the /var/log/vzmessages.hn1 file.

However, it is recommended to use the ttylogd serial console daemon to maintain serial log files. This daemon is launched by the /etc/init.d/ttylogd script on the system startup and uses the /etc/ttylogd.conf file for the correct parameters. Thus, all you need to do to automate the messages collection on the Monitor Node is to install ttylogd and edit appropriately its configuration file.

First, install the daemon on the Monitor Node. The corresponding package can be found on your Parallels Virtuozzo Containers CD, DVD, or in your local distribution directory in the /virtuozzo/RPMS subdirectory:

```
# rpm -ihv ttylogd-3.0.0-2.swsoft.i386.rpm
Preparing...                ########################################### [100%]
   1:ttylogd                ########################################### [100%]
```

Now, take a look at the /etc/ttylog.conf file. It must comprise a number of string sections of the following type:

```
# Settings for ttyS0
# PORT1=/dev/ttyS0
# HOST1=ts2
# LOG1="/var/log/console-${HOST1}.log"
```

The value of the PORTX parameter is the serial console device on the Monitor Node;

The value of the HOSTX parameter is the name of the Hardware Node to be monitored. This parameter is optional, it is used for convenience.

The value of the LOGX parameter is the path to the file that will accumulate messages coming to the specified serial console from the Hardware Node. You may use the ${HOSTX} variable to synchronize the name of the file with the name of the Hardware Node.

You must have as many such sections as the number of Nodes you wish to monitor. Copy and paste the needed number of these sections in the ttylogd.conf configuration file. Apply one and the same number after "PORT", "HOST", and "LOG" throughout each section, and increment this number with each new section. Edit the values of the "PORT", "HOST", and "LOG" parameters appropriately for each and every Hardware Node to be monitored and remove the hash marks before them. Then modify the DAEMONS="1 2" line to include all the numbers (separated by spaces) you used in your sections after the "PORT", "HOST", and "LOG" parameters. Save the file.

You may also consult the ttylogd(8) and ttylog.conf(5) manual pages.

Checking That Logging Works

Now reboot the Hardware Node. After the Hardware Node is up, check the file on the Monitor Node where the messages are stored (for example, `/var/log/vzmessages.hn1`). The file should contain the messages printed by the kernel during the boot-up.

Upon loading, the watchdog module should produce to the log file the output similar to the one below:

```
MODULES="$PRELOAD_MODULES vzfs vzmon vzdquota vzdev vzwdog"
*** VZWDOG: time 1034715427.628385 uptime 994993 \
CPU 0 $Revision: 1.1.2.1 $ ***
           CPU0
  0:     994995    IO-APIC-edge  timer
  1:          2    IO-APIC-edge  keyboard
  8:          1    IO-APIC-edge  rtc
 14:          2    IO-APIC-edge  ide0
 21:       1999    IO-APIC-level  eth0
 26:      11037    IO-APIC-level  aic7xxx
 27:         16    IO-APIC-level  aic7xxx
[a lot of lines suppressed]
```

Setting Up netconsole

The `netconsole` module allows you to send the console messages from the Parallels Virtuozzo Containers kernel installed on the Hardware Node to the Monitor Node. To prepare this module for use in your network environments, you should perform the following operations:

1. Set up the `netconsole` module on the Hardware Node to be monitored.
2. Configure the Monitor Node to collect messages from the `netconsole` module on the Hardware Node.

Both operations are described in the following subsections in detail.

Notes:

1. The `netconsole` module uses the UDP (User Datagram Protocol) transport protocol to send kernel messages from the Hardware Node to the Monitor Node. As this protocol provides simple but unreliable message services, you are highly recommended to have both Nodes located as close to each other as possible (best of all - in one and the same network segment) to ensure that all kernel messages can reach the Monitor Node.

2. Since the `netconsole` module allows you to monitor the system and collect kernel messages only after the kernel is successfully loaded and the corresponding NIC card is initialized, we recommend that you set up a serial console and use it as the primary tool for monitoring your system. Configuring the Monitor Node as a serial console enables you to start collecting the Node kernel logs right after the kernel boot process is started.

Preparing the Hardware Node for Sending Kernel Messages

First, set up the `netconsole` module on the Hardware Node you wish to monitor. Depending on the Linux distribution installed on your Node, the operations you have to perform to configure this module may slightly differ. Listed below are examples of how to set up the `netconsole` module for some Linux distributions:

To configure the `netconsole` module on a Hardware Node running Red Hat Enterprise Linux 3 or 4:

1 Specify the IP address of the Monitor Node as the value of the `SYSLOGADDR` parameter in the `/etc/sysconfig/netdump` file. Assuming that your Monitor Node has the `192.168.0.100` IP address assigned, you can do it as follows:

```
SYSLOGADDR=192.168.0.100
```

2 Execute the following command on the Hardware Node:

```
# service netdump restart
```

To configure the `netconsole` module on a Hardware Node running Red Hat Enterprise Linux 5.1 and Fedora 8:

> **Note:** For instructions on how to load the `netconsole` module on Hardware Nodes running Red Hat Enterprise 5.0, please see the information below.

1 Specify the IP address of the Monitor Node as the value of the `SYSLOGADDR` parameter in the `/etc/sysconfig/netconsole` file. Assuming that your Monitor Node has the `192.168.0.100` IP address assigned, you can do it as follows:

```
SYSLOGADDR=192.168.0.100
```

2 Execute the following command on the Hardware Node:

```
# service netconsole restart
```

To configure the `netconsole` module on a Hardware Node running SUSE Linux Enterprise Server 10:

1 Make sure that the `netconsole-tools` RPM package is installed on the Hardware Node.

2 Run the `netconsole-server` utility on the Hardware Node and specify the Monitor Node IP address as its parameter. For example:

```
# netconsole-server 192.168.0.100
```

To configure the `netconsole` module on Hardware Nodes running other Linux distributions, please see the documentation shipped with these distributions.

Another way of loading and configuring the netconsole module on your Hardware Node is to use the modprobe utility. The procedure of setting up netconsole using this utility is identical for all Linux distributions and can be used for the netconsole configuration irrespective of a Linux distribution installed on the Node. However, to configure the netconsole module with modprobe, you have to manually specify a number of parameters when running this utility (e.g. the Node IP address and the name of the network card installed on this Node). For example, you can issue the following command to prepare the netconsole module on your Node for sending kernel logs to the Monitor Node:

```
# /sbin/modprobe netconsole \
  netconsole=6666@192.168.0.50/eth0,514@192.168.0.100/00:17:31:D9:D7:C8
```

The parameters used in this command are explained below:

- 6666: the port on the Hardware Node used for sending UDP messages.
- 192.168.0.50: the IP address assigned to the Hardware Node.
- eth0: the name of the network interface card installed on the Hardware Node.
- 514: the port on the Monitor Node used to listen to incoming UDP messages from the Hardware Node.
- 192.168.0.100: the IP address assigned to the Monitor Node.
- 00:17:31:D9:D7:C8: the MAC address of the Monitor Node (if you do not know how to find out the Monitor Node MAC address, please turn to the next subsection).

If you wish the netconsole module to automatically load on the Hardware Node boot up, you need to add the following string to the /etc/rc.d/rc.local script on the Node:

```
/sbin/modprobe netconsole \
netconsole=6666@192.168.0.50/eth0,514@192.168.0.100/00:17:31:D9:D7:C8
```

Determining the Monitor Node MAC Address

You can execute the following command on your Hardware Node to learn the MAC address assigned to the Monitor Node (we assume that the Monitor Node has the 192.168.0.100 IP address assigned):

```
# /sbin/arp -n 192.168.0.100
Address          HWtype   HWaddress           Flags Mask   Iface
192.168.0.100    ether    00:17:31:D9:D7:C8   C            eth0
```

In the example above, the Monitor Node has the MAC address of 00:17:31:D9:D7:C8 assigned.

Starting Messages Collection on the Monitor Node

The kernel messages sent by the netconsole module on the Hardware Node may be collected by dumping the data received on a UDP port on the Monitor Node. The simplest way to collect this data is by executing the following command on the Monitor Node:

```
# nc -l -u 514 > /var/log/netconsole_logs
```

This way the messages will be collected on the 514 UDP port (this is the same port you specified when setting up netconsole on the Hardware Node) and stored in the /var/log/netconsole_logs file on the Monitor Node. However, the collected messages will have no time stamps and the redirection to the file will become broken in the case of a Monitor Node reboot. So, we recommend that you use the ttylogd serial console daemon to maintain kernel messages on the Monitor Node.

Note: Some Linux distributions (e.g., SLES 10 SP1) include the netcat utility in their distributions instead of nc. If this is your case, use netcat to collect kernel messages coming from netconsole in the same way you would use the nc utility.

The ttylogd serial console daemon is used to effectively process kernel messages received from netconsole on the Monitor Node. This daemon is launched by the /etc/init.d/ttylogd script on the system startup and uses the /etc/ttylogd.conf file for the correct control parameters. Thus, all you need to do to automate the kernel messages collection on the Monitor Node is to install ttylogd and to edit appropriately its configuration file.

First, you should install the daemon on the Monitor Node if you have not done so before. The corresponding package can be found in the /virtuozzo/RPMS subdirectory on your Parallels Virtuozzo Containers CD, DVD, or in your local distribution directory:

```
# rpm -ihv ttylogd-4.7.0-2.swsoft.i386.rpm
Preparing...                ########################################### [100%]
   1:ttylogd                ########################################### [100%]
```

Now take a look at the /etc/ttylog.conf file. It must comprise a number of string sections of the following type:

```
# Settings for netconsole
# PORT3=514
# HOST3=ts4
# LOG3="/var/log/console-${HOST3}.log"
```

The value of the PORT*X* parameter is the UDP port number on the Monitor Node used to listen to incoming kernel messages from your Hardware Node.

The value of the HOST*X* parameter is the name of the Hardware Node to be monitored. This parameter is optional, it is used for convenience.

The value of the LOG*X* parameter is the path to the file that will accumulate messages coming to the specified serial console from the Hardware Node. You may use the ${HOST*X*} variable to synchronize the name of the file with the name of the Hardware Node.

You must have as many such sections as the number of Nodes you wish to monitor. Copy and paste the needed number of these sections in the ttylogd.conf configuration file. Apply one and the same number after "PORT", "HOST", and "LOG" throughout each section, and increment this number with each new section. Edit the values of the "PORT", "HOST", and "LOG" parameters appropriately for each and every Hardware Node to be monitored and remove the hash marks before them. Then modify the DAEMONS="1 2" line in this file to include only those numbers (separated by spaces) that are used in your sections after the "PORT", "HOST", and "LOG" parameters. Save the file.

After you have configured the /etc/ttylog.conf file, you should restart the ttylogd daemon for the changes made to this files to come into effect:

```
# service ttylogd restart
Shutting down ttylogd:                    [OK]
Starting ttylogd 514:                     [OK]
```

You may also consult the ttylogd(8) and ttylog.conf(5) manual pages.

Increasing Kernel Log Level

To increase the kernel verbosity on the Hardware Node to get more informative kernel messages on the Monitor Node, you can proceed as follows:

1. Check the current kernel log level:

```
# cat /proc/sys/kernel/printk
6       4       1       7
```

2. Set the log level to the maximum possible value:

```
# echo 8 4 1 8 >/proc/sys/kernel/printk
```

3. On Hardware Nodes running RHEL-based distributions, additionally edit the KLOGD_OPTIONS parameter in the /etc/sysconfig/syslog file as follows:

```
KLOGD_OPTIONS="-x -c 8"
```

4. If your Hardware Node has an SMP kernel installed, additionally execute the following command on the Node:

```
# echo 8 >/proc/sys/kernel/silence-level
```

You can permanently save the changes made to the kernel log level configuration by doing the following:

1. Adding the following string to the /etc/sysctl.conf file on the Hardware Node:

```
kernel.printk = 8 4 1 8
```

2. Specifying the debug parameter in the boot loader configuration file (/etc/grub.conf or /etc/lilo.conf) on the Hardware Node.

On Hardware Nodes with SMP kernels, you should also add the silencelevel=8 string to the boot loader configuration file on the Node.

Checking That netconsole Logging Works

You can check that you have successfully set up netconsole by loading and unloading a certain kernel module on the Hardware Node and viewing the file on the Monitor Node where the messages are stored. The file should contain the messages printed by the kernel during the module loading/unloading. Assuming that all messages coming from netconsole are to be stored in the /var/log/netconsole_logs file For example, netconsole will send messages like the following during the loop module loading on the Hardware Node:

```
Jan 22 17:49:57 ts4 ttylogd v.2.1.0-5 started
Jan 22 06:14:58 ts4 loop: loaded (max 8 devices)
```

Preparing the Monitor Node for Sending Alerts

The Monitor Node can also be configured to remotely check up the state of the Hardware Nodes – if they are running or down, as well as a number of vital parameters – and to send instant alerts via e-mail if anything goes wrong.

To do this, it is necessary to install the `vzrmon` package on the Monitor Node, which are located on your Parallels Virtuozzo Containers CD, DVD, or in your local distribution directory in the `/virtuozzo/RPMS` subdirectory. For example:

```
# rpm -ihv vzrmon-4.7.0-3.i386.rpm
Preparing...                ########################################### [100%]
   1:vzrmon                 ########################################### [100%]
```

Note: You might also need to install the `gnuplot` and `mutt` packages, if they are not already installed. If this is the case, you will receive the corresponding notification. These packages are not included with Parallels Virtuozzo Containers, as they are part of a standard Red Hat Linux distribution.

After the `vzrmon` package is installed, the `vzrmond` daemon is started on the Monitor Node. You should manually edit the `vzrmond` configuration file (see the next subsection for details) to define the list of Nodes to monitor and the way the alerts are sent. However, `vzrmond` needs to be able to remotely log in to the specified Node(s) without having to provide a root password. Therefore, you should provide each Node to be monitored with your authorized public SSH RSA key. It can be done in the following way. First, you should 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):
/root/.ssh/id_rsa already exists.
Overwrite (y/n)? y
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 should leave an empty passphrase in the above procedure.

Next, transfer your public key to each Hardware Node you are going to monitor to the `/root/.ssh` directory (use some intermediary name for the file not to overwrite the corresponding file on the Hardware Node):

```
# 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
```

Finally, you should add the contents of the transferred file to the `authorized_keys` file in this very directory of the Hardware Node. Log in to the Hardware Node, go to the `/root/.ssh` directory and issue the following command in it:

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

Now the Monitor Node should be able to log in to this Hardware Node as root without having to provide the root password. You should copy the public RSA file of the Monitor Node to every Hardware Node to be monitored and add its contents to the `authorized_keys` file in the `/root/.ssh` directory.

Using External Applications for Sending Alerts

Along with sending e-mail messages, `vzrmond` allows you to use external instant messaging applications for sending alerts via other means of communication (e.g. via ICQ or SMS). Let us assume that you wish to configure the `Centericq` application to send notifications about the Hardware Node state to your ICQ. To this effect, you should perform the following operations on the Monitor Node:

- Install the `centericq` package, for example:

```
# rpm -ihv centericq-4.21.0-1.i386.rpm
Preparing...       #################################### [100%]
   1:centericq     #################################### [100%]
```

- Configure the `CUSTOM_ACTION` and `CUSTOM_LIST` parameters in the `/etc/vzrmond.conf` configuration file to inform `vzrmond` that it should use the `Centericq` application for sending messages. For example:

```
...
CUSTOM_ACTION="centericq -s msg -p icq"
CUSTOM_LIST="-t 24359283"
...
```

The parameters specified above mean the following:

- The `-s` option is used to denote the type of event to be sent (in our case it is a message - 'msg').
- The `-p` option is used to specify the destination instant messaging network (`icq`).
- The `-t` option is used to indicate the ICQ UIN (Unified Identification Number) where the message is to be sent (`24359283`).

Note: Detailed information on all parameters that can be specified in the `vzrmond.conf` file is provided in the *Parallels Virtuozzo Containers 4.7 Reference Guide*.

Using vzstatrep to Monitor Hardware Nodes

The `vzstatrep` utility allows you to analyze the main resources consumption of one or several Hardware Nodes and to receive information on this consumption in the form of statistic reports and graphics at your e-mail address(es). `vzstatrep` is included in the `vzrmon` package and automatically installed on the Monitor Node during the `vzrmon` package installation. For more information on how to install `vzrmon`, please see the previous subsection.

To start using `vzstatrep`, you should manually edit the `vzstatrep.conf` configuration file located in the `/etc` directory on the Monitor Node to define a list of Hardware Nodes whose resources consumption is to be analyzed and specify one or several e-mail addresses where the Hardware Node statistic reports and graphics are to be sent. In this file, you can also set a number of other parameters (e.g. the resources the usage of which will be presented in the graphical form with the help of the `gnuplot` utility or the path to the directory on the Hardware Node where `vzstatrep` will search for the logs to be analyzed). Detailed information on the `vzstatrep.conf` file and all its options is provided in the *Parallels Virtuozzo Containers 4.7 Reference Guide*.

By default, the `vzstatrep` utility is scheduled as a `cron` job to automatically run once a day. When launched, the `vzstatrep` utility performs the following operations:

1. Connects to the Hardware Node(s) to be monitored.
2. Downloads the logs collected by the `vzlmond` utility and stored in the `/var/log/vzstat` directory on the Hardware Node by default.
3. Analyzes the downloaded logs and generates the statistic report and graphics on the basis of these logs.
4. Sends the generated statistic report and graphics at the specified e-mail address(es).

Let us assume that you wish to analyze the resources statistics from the Hardware Node having the hostname of `my_hardware_node.com` and to periodically (i.e. once a day) receive this statistics report at the `peter@my_domain.com` e-mail address. To do this:

1. On the Monitor Node, open the `/etc/vzstatrep.conf` file for editing:

```
# vi /etc/vzstatrep.conf
```

2. In the file, set the `STATS_EMAIL` and `NODES` parameters as follows:

```
NODES="my_hardware_node.com"
STATS_EMAIL="peter@my_domain.com"
```

3. Save the `/etc/vzstatrep.conf` file.

From now on, an e-mail message containing information on the Hardware Node resources consumption will be sent every day at the `peter@my_domain.com` e-mail address. However, if you wish to get the Hardware Node statistic report at the current moment, you can manually run the `vzstatrep` command on the Monitor Node:

```
# vzstatrep --plot --sendmail
```

As a result of this command, an e-mail message will be instantly sent to the `peter@my_domain.com` address containing the text information on the Hardware Node resources consumption (on the memory and CPU consumption on the Node, network statistics, etc.). Besides, you will get a number of attached files where the resources usage is presented in the form of graphics generated by the `gnuplot` utility. Detailed information on all `vzstatrep` options (including the `--plot` and `--sendmail` options used in the example above) is provided in the *Parallels Virtuozzo Containers 4.7 Reference Guide*.

Glossary

Application template is a template used to install a set of applications in *Containers*. See also *Template*.

Container (or *regular Container*) is a virtual private server, which is functionally identical to an isolated standalone server, with its own IP addresses, processes, files, its own users database, its own configuration files, its own applications, system libraries, and so on. Containers share one *Hardware Node* and one OS kernel. However, they are isolated from each other. A Container is a kind of ‘sandbox’ for processes and users.

Container 0 is used to designate a *Hardware Node* where the *Parallels Virtuozzo Containers* software is installed.

EZ template is a template file that points to a repository with the packages that comprise the template. Unlike *standard templates*, EZ templates cannot be updated because the repository stays the same. However, the packages in the repository can be updated.

Hardware Node (or *Node*) is a server where the Parallels Virtuozzo Containers software is installed for hosting *Containers*. Sometimes, it is marked as *Container 0*.

Host Operating System (or *Host OS*) is an operating system installed on the *Hardware Node*.

OS template (or *Operating System template*) is used to create new *Containers* with a preinstalled operating system. See also *Template*.

Package set is a synonym for *Template*.

Parallels Virtual Automation is a tool designed for managing Hardware Nodes and all *Containers* residing on them with the help of a standard Web browser on any platform.

Parallels Management Console (or *Management Console*) is a Parallels Virtuozzo Containers management and monitoring tool with graphical user interface. It is used to control individual *Hardware Nodes* and their *Containers*. *Management Console* is cross-platform and runs on both Microsoft Windows and Linux workstations.

Parallels Power Panel is a means for administering personal Containers with the help of a standard Web browser (Internet Explorer, Mozilla, etc.) on any platform.

Parallels Virtuozzo Containers is a complete server automation and virtualization solution allowing you to create multiple isolated *Containers* on a single physical server to share hardware, licenses, and management effort with maximum efficiency.

Private area is a part of the file system where *Container* files that are not shared with other *Containers* are stored.

Standard template (obsolete) is a template file that has inside itself all the re-usable files of all the packages comprising the template. If newer versions of any of these packages appear, a standard template can be correspondingly updated. Compare *EZ template*.

Template (or *package set*) is a set of original application files (packages) repackaged for mounting over Virtuozzo File System. There are two types of templates. OS Templates are used to create new *Containers* with a preinstalled operating system. Application templates are used to install an application or a set of applications in *Containers*. See also *Standard template* and *EZ template*.

UBC is an abbreviation of *User Beancounter*.

User Beancounter is the subsystem of the *Parallels Virtuozzo Containers* software for managing *Container* memory and some system-related resources.

venet0 is a virtual networking device, a gateway from a *Container* to the external network.

Virtual Environment (or *VE*) is an obsolete designation of a *Container*.

Virtuozzo Control Center (or *VZCC*) is an obsolete designation of *Parallels Virtual Automation*.

Virtuozzo File System (*VZFS*) is a virtual file system for mounting to Container private areas. VZFS symlinks are seen as real files inside *Containers*.

Parallels Virtuozzo Containers license is a special license that you should load to the *Hardware Node* to be able to start using the *Parallels Virtuozzo Containers* software. Every *Hardware Node* shall have its own license.

Virtual Private Server (or *VPS*) is an obsolete designation of a *Container*.

Parallels Agent (or *Parallels Agent Protocol*) is an XML-based protocol used to monitor and manage a *Hardware Node*. The *Parallels Agent* software implements this protocol and is a backend for the *Parallels Management Console*.

Index