

---

Parallels

# Parallels Virtuozzo Containers 4.6 for Windows

Reference Guide



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

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

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

*This product is protected by United States and international copyright laws. The product's underlying technology, patents, and trademarks are listed at <http://www.parallels.com/trademarks>.*

*Microsoft, Windows, Windows Server, Windows NT, Windows Vista, and MS-DOS are registered trademarks of Microsoft Corporation.*

*Linux is a registered trademark of Linus Torvalds.*

*Mac is a registered trademark of Apple, Inc.*

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

# Contents

<b>Preface</b>	<b>5</b>
About Parallels Virtuozzo Containers .....	6
About This Guide .....	7
Organization of This Guide .....	7
Documentation Conventions .....	8
Getting Help .....	9
Feedback .....	10
<b>Parallels Virtuozzo Containers Utilities Overview</b>	<b>11</b>
<b>General Utilities</b>	<b>13</b>
vzctl .....	14
vzctl create .....	16
vzctl delete and vzctl destroy .....	17
vzctl mount and vzctl umount .....	17
vzctl start, vzctl stop, vzctl restart, and vzctl status .....	17
vzctl set .....	18
vzctl exec, vzctl exec2, and vzctl enter .....	23
vzctl mounttext and vzctl umounttext .....	24
vzctl partadd and vzctl partdel .....	24
vzctl reinstall .....	25
vzctl shrink .....	25
vzctl defrag .....	26
vzctl addrole, vzctl delrole, vzctl enumrole .....	27
vzlist .....	28
vzlist Output Parameters and Their Modifiers .....	29
vzquota .....	31
vzquota setlimit .....	32
vzquota stat and vzquota show .....	32

**Licensing Utilities 33**

vzlicload ..... 34  
 vzlicview ..... 35

**Migration Utilities 36**

vzmigrate ..... 37  
 vzp2v ..... 39  
 vzmlcal ..... 40

**Backing-Up Utilities 41**

vzabackup ..... 42  
 vzarestore ..... 44  
 vzvssctl ..... 45

**Template Management Utilities 46**

vzpkgls ..... 46  
 vzpkgdeploy ..... 47  
 vzpkgadd ..... 47  
 vzpkgrm ..... 48

**Supplementary Utilities 49**

vzkeygen ..... 49  
 vzcache ..... 50  
 vzlscache ..... 51  
 vzuncache ..... 52  
 vznetcfg ..... 53  
 vznetstat ..... 54  
 vzdevctl ..... 55  
 vzcpucfg ..... 57  
 vzquery ..... 58  
 vzwinupdatecmd ..... 59

**Glossary 60**

**Index 62**

## CHAPTER 1

# Preface

## In This Chapter

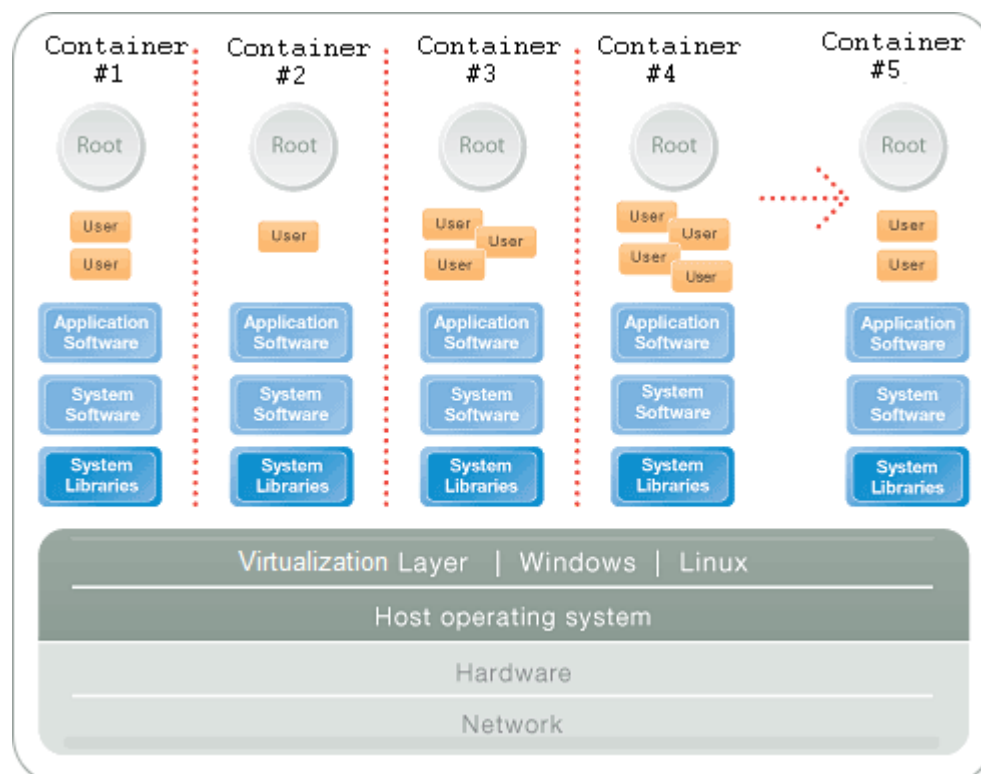
About Parallels Virtuozzo Containers.....	6
About This Guide.....	7
Getting Help.....	9
Feedback .....	10

# About 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**—Division of 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 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.

---

## About This Guide

This guide is a complete reference on all Parallels Virtuozzo Containers configuration files and Hardware Node command-line utilities. It familiarizes you with the way to configure Parallels Virtuozzo Containers to meet your requirements and to perform various tasks by using the corresponding Parallels command line utilities.

The primary audience for this guide is anyone who is looking for an explanation of a particular configuration option, does not understand a Parallels file format, needs help for a particular command, or is seeking for a command to perform a certain task.

## Organization of This Guide

Chapter 2, **Parallels Virtuozzo Containers Utilities Overview**, lists all utilities supported in the current version of Parallels Virtuozzo Containers.

Chapter 3, **General Utilities**, describes utilities intended for performing day-to-day maintenance tasks.

Chapter 4, **Licensing Utilities**, provides the information on utilities for managing Parallels Virtuozzo Containers licenses.

Chapter 5, **Migration Utilities**, focuses on utilities for migrating Containers between Hardware Nodes or within one Hardware Node.

Chapter 6, **Backing-Up Utilities**, describes utilities for backing up and restoring Container private areas and configuration files.

Chapter 7, **Template Management Utilities**, focuses on utilities for creating and managing templates.

Chapter 8, **Miscellaneous Utilities**, concentrates on utilities for performing different tasks in the Hardware Node and Container context.

## Documentation Conventions

Before you start using this guide, it is important to understand the documentation conventions used in it. For information on specialized terms used in the documentation, see the glossary at the end of this document.

### Typographical Conventions

The following kinds of formatting in the text identify special information.

Formatting convention	Type of Information	Example
Preformatted	On-screen computer output in your command-line sessions; source code in XML, C++, or other programming languages.	Saved parameters for Container 101
Preformatted Bold	What you type, as contrasted with on-screen computer output.	C:\Documents and Settings\Administrator> vzlist
Moonscape	The names of commands, files, and directories.	Use <code>vzctl start</code> to start a Container.
Monospace Italics	Designates a command line placeholder, which is to be replaced with a real name or value.	To delete a Container, type <code>vzctl delete CT_ID</code> .
Special Bold	All elements of the graphical user interface (GUI): menu items, menu options, menu buttons, etc.	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 or to introduce a term.	<i>Host operating system</i> is an operating system installed on the Hardware Node.
CAPITALS	Names of keys on the keyboard.	SHIFT, CTRL, ALT
KEY+KEY	Key combinations for which the user must press and hold down one key and then press another.	CTRL+P, ALT+F4

### General Conventions

Be aware of the following conventions used in this book.

- Chapters in this guide are divided into sections, which, in turn, are subdivided into subsections. For example, **Documentation Conventions** is a section, and **General Conventions** is a subsection.
- When following steps or using examples, be sure to type double-quotes (") and single-quotes (') exactly as shown.

---

## Getting Help

In addition to this guide, there are a number of other resources available for Parallels Virtuozzo Containers 4.6 which can help you use the product more effectively. These resources include:

- *Getting Started With Parallels Virtuozzo Containers 4.6*. This guide provides basic information on how to install Parallels Virtuozzo Containers 4.6 on your server, create new Containers, and perform main operations on them.
- *Parallels Virtuozzo Containers 4.6 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 4.6* guide, it contains a more detailed description of all the operations needed to install and set Parallels Virtuozzo Containers 4.6 to work including planning the structure of your Parallels Virtuozzo Containers network, performing the Parallels Virtuozzo Containers unattended installation, etc. Besides, it does not include the description of any Container-related operations.
- *Parallels Virtuozzo Containers 4.6 User's Guide*. This guide provides comprehensive information on Parallels Virtuozzo Containers 4.6 covering the necessary theoretical conceptions as well as all practical aspects of working with Parallels Virtuozzo Containers. However, it does not deal with the process of installing and configuring your Parallels Virtuozzo Containers system.
- *Parallels Virtuozzo Containers 4.6 Templates Management Guide*. This guide is meant to provide complete information on Parallels Virtuozzo Containers templates, an exclusive Parallels Virtuozzo Containers technology allowing you to efficiently deploy standard Windows applications inside your Containers and to greatly save the Hardware Node resources (physical memory, disk space, etc.).
- *Deploying Microsoft Clusters in Parallels-Based Systems*. This document provides information on creating Microsoft failover and Network Load Balancing clusters in Parallels Virtuozzo Containers-based systems.
- *Parallels Management Console Help*. This help system provides detailed information on Parallels Management Console, a graphical user interface tool for managing Hardware Nodes and their 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 their 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/>).

# Parallels Virtuozzo Containers Utilities Overview

This chapter documents the utilities supported in the current version of Parallels Virtuozzo Containers. For every utility, all available command-line options are described.

The Parallels Virtuozzo Containers command-line utilities can be subdivided into the following categories: general utilities, licensing utilities, Container migration utilities, Container backup utilities, template management utilities, and miscellaneous utilities.

*General utilities* are intended for performing day-to-day maintenance tasks:

<code>vzctl</code>	Utility to control Containers.
<code>vzlist</code>	Utility to view a list of Containers existing on the Node with additional information.
<code>vzquota</code>	Utility to control Parallels Virtuozzo Containers disk quotas.

*Licensing utilities* are used for managing Parallels Virtuozzo Containers licenses:

<code>vzlicview</code>	Utility to show the Parallels Virtuozzo Containers license status and parameters.
<code>vzlicload</code>	Utility to install Parallels Virtuozzo Containers licenses.

*Container migration utilities* allow you to migrate Containers between Hardware Nodes or within one Hardware Node:

<code>vzmigrate</code>	Utility for migrating Containers to another Hardware Node.
<code>vzmlocal</code>	Utility for migrating Containers within the same Hardware Node.
<code>vzp2v</code>	Utility for migrating physical servers to Containers.

*Container backup utilities* allow you to back up and restore Container private areas and configuration files:

<code>vzabackup</code>	Utility to back up Containers.
<code>vzarestore</code>	Utility to restore Containers.
<code>vzvsctl</code>	Utility to suspend and resume the Container activity during the backup operation.

*Template management utilities* allow you to create new templates, install them on the Hardware Node, and perform various operations on them:

<code>vzpkgls</code>	Utility to view the templates installed on the Hardware Node and/or applied to Containers.
<code>vzpkgdeploy</code>	Utility to install/remove templates on/from the Hardware Node.
<code>vzpkgadd</code>	Utility to add application templates to Containers.
<code>vzpkgrm</code>	Utility to remove templates from Containers.

*Miscellaneous utilities* perform different tasks in the Hardware Node and Container context:

<code>vzkeygen</code>	Utility to show the Hardware Node ID.
<code>vzcache</code>	Utility to cache common files across Containers.
<code>vzlscache</code>	Utility to list the folders created by <code>vzcache</code> .
<code>vzuncache</code>	Utility to detach a Container from its cache and copy the cached files back to the Container private area.
<code>vznetcfg</code>	Utility to manage Virtual Networks and network classes on the Hardware Node.
<code>vznetstat</code>	Utility to display the traffic usage statistics for Containers.
<code>vzdevctl</code>	Utility to forward the hardware devices (SCSI, iSCSI, USB flash drives, etc.) from the Hardware Node to Containers.
<code>vzcpucfg</code>	Utility to manage CPU pools on the Hardware Node.
<code>vzquery</code>	Utility to determine Container IDs using process and session IDs.
<code>vzwinupdatecmd</code>	Utility to manage Windows Server updates inside Containers.

## CHAPTER 2

# General Utilities

This chapter describes utilities intended for performing day-to-day maintenance tasks.

## In This Chapter

vzctl .....	14
vzlist .....	28
vzquota .....	31

---

## vzctl

`vzctl` is the primary tool for Container management. To use it, you have to log in to the Hardware Node as the administrator. The syntax of `vzctl` is:

```
vzctl [--quiet | --verbose] command CT_ID
vzctl --version
vzctl --help
```

`command` can be one of the following:

<code>create</code>	Creates a Container.
<code>delete</code>	Removes a Container.
<code>destroy</code>	Removes a Container.
<code>mount</code>	Mounts the Container private area.
<code>umount</code>	Unmounts the Container private area.
<code>start</code>	Starts a Container.
<code>stop</code>	Stops a Container.
<code>restart</code>	Restarts a Container.
<code>status</code>	Displays the Container status.
<code>set</code>	Sets Container parameters.
<code>enter</code>	Provides a way for the Hardware Node administrator to “enter” a Container without knowing the Container administrator password.
<code>exec</code> , <code>exec2</code>	Run arbitrary commands inside a Container without logging in to the Container. The difference between the command is their returned status.
<code>mountext</code>	Mounts folders and drives of the Hardware Node to Containers.
<code>umountext</code>	Unmounts external folders and drives of the Hardware Node from Containers.
<code>partadd</code>	Creates a new Container drive by mounting it to a loopback file on the Hardware Node.
<code>partdel</code>	Unmounts the Container drive mounted by means of the <code>vzctl partadd</code> command.
<code>reinstall</code>	Restores the original state of the Container system and application files.
<code>shrink</code>	Shrinks compact virtual disk drives inside Containers.
<code>defrag</code>	Defragments Container virtual disks.
<code>addrole</code>	Adds a new server role to the Container.
<code>delrole</code>	Removes a server role from the Container.
<code>enumroles</code>	Lists the server roles installed or ready to be installed inside Containers.

Verbosity options can be used with any of the above commands:

<code>--quiet</code>	Disables logging to the display and to the log file.
<code>--verbose</code>	Sets the log level to the maximum possible value.

You can also pass to `vzctl` one of the following options:

- `--version` Displays the Parallels Virtuozzo Containers version installed on the Hardware Node.
- `--help` Displays the `vzctl` usage information.

## vzctl create

This command is used to create new Containers. It has the following syntax:

```
vzctl create CT_ID --pkgset name [options]
```

A Container ID is required for this command and shall be unique within a Hardware Node. Container IDs from 1 to 100 are reserved for internal Parallels Virtuozzo Containers needs. Do not use IDs from 1 to 100 for your Containers.

Command options are as follows:

Name	Description
<code>--pkgset name</code>	Mandatory. Denotes the OS template to base the Container on.
<code>--ipadd ip_address</code>	Optional. Adds the specified IP address to the list of IP addresses the Container can use and brings up the network interface with this address inside the Container.
<code>--hostname hostname</code>	Optional. The hostname to assign to the Container.
<code>--config filename</code>	Optional. Applies the resource parameters from the specified configuration file to the Container. Configuration files are located in the <code>X:\Program Files\Parallels\Containers\Configs</code> folder on the Hardware Node and have the form <code>ve-&lt;name&gt;.conf-sample</code> .
<code>--name name</code>	An arbitrary name to assign to the Container. This name can be used, along with the Container ID, to refer to the Container while performing Container-related operations.
<code>--private path</code>	Optional. Specifies the location of the Container private area. The Container private area can be placed anywhere on your hard disk. While choosing a path for this folder, keep in mind the following: <ul style="list-style-type: none"> <li>▪ This folder cannot be a mount point, i.e. you cannot mount external disk partitions to this folder.</li> <li>▪ This folder cannot be a network share, i.e. it cannot be located on a server network drive.</li> <li>▪ The hard disk partition where this folder will be located must have no less than 10 Gb of free disk space.</li> </ul>
<code>--diskspace size</code>	The total size of disk space to allocate to the Container, in 1 Kb blocks.
<code>--disktype plain compact</code>	Optional. The type of the Container virtual disk drive. It can be one of the following: <ul style="list-style-type: none"> <li>▪ <code>plain</code>: in this case, the size of a virtual hard disk is fixed, i.e. all disk space is allocated during the virtual disk creation.</li> <li>▪ <code>compact</code> (default): in this case you set only the maximal size of a virtual hard disk. A Container hard disk grows in size each time new data is written to the hard disk and can increase up to the maximum size specified during the hard disk creation.</li> </ul>

## vzctl delete and vzctl destroy

These commands are used to delete a Container, which is no longer needed, from the Hardware Node. The syntax of the commands is as follows:

```
vzctl delete <CT_ID>
vzctl destroy <CT_ID>
```

When executed, `vzctl delete/vzctl destroy` physically removes all the files located in the Container private area with the default path of `C:\vz\private\CT_ID`. These commands do not take any additional arguments and require the Container to be stopped and its private area to be unmounted.

## vzctl mount and vzctl umount

These commands take no additional arguments:

```
vzctl mount <CT_ID>
vzctl umount <CT_ID>
```

The first command mounts the Container private area to the Container root folder (`X:\vz\root\<CT_ID>`) without starting it and the Container registry to the `HKEY_LOCAL_MACHINE\<CT_ID>` key in the registry of the Node. Normally, you do not have to use this command because the `vzctl start` command mounts the Container private area and registry automatically.

The `vzctl umount` command unmounts the Container private area and registry. Usually, there is no need in using this command either because `vzctl stop` unmounts the Container private area and registry automatically.

## vzctl start, vzctl stop, vzctl restart, and vzctl status

These four commands have the same syntax and take no obligatory arguments:

```
vzctl start <CT_ID>
vzctl stop <CT_ID>
vzctl restart <CT_ID>
vzctl status <CT_ID>
```

The first command is used to start a Container.

`vzctl stop` shuts the Container down. There are some commands that can be performed only after running the `vzctl stop` command (for example, `vzctl delete`).

The `vzctl restart <CT_ID>` command consecutively performs the stopping and starting of a Container.

The `vzctl status` command shows the current Container state.

## vzctl set

This command is used for setting Container parameters. It has the following syntax:

```
vzctl set <CT_ID> <setting_name> <value> [--save]
```

An optional `--save` switch tells `vzctl` whether to save changes into the Container configuration file. Most of the Container settings can be changed dynamically without the necessity of the Container reboot.

The settings that can be passed to the `vzctl set` command are subdivided into the following categories: miscellaneous, networking, and resource management parameters.

### *Miscellaneous settings:*

<code>--onboot yes no</code>	This setting requires the <code>--save</code> switch. If you set it to “yes”, Parallels Virtuozzo Containers will automatically start the Container on the next system start-up.
<code>--userpasswd user:password</code>	This setting is used to set a new password for the specified user inside the Container (this user must already exist).  Omitting the user name (e.g., <code>:XXXXXXXXXX</code> ) will set the password for the built-in Administrator account inside the Container, even though it was renamed.
<code>--offline_management yes no</code>	This setting enables/disables the direct managing of the Container through a common Internet browser by means of Parallels Power Panel. To enable/disable the offline management feature for the Hardware Node and, consequently, for all Containers, set this parameter for Container 0.
<code>--showctid yes no</code>	This setting defines the appearance of the Container ID string in the top right corner of the Container desktop: <ul style="list-style-type: none"> <li>▪ If set to <code>yes</code>, the ID string is displayed in the top right corner of the Container desktop helping you identify the ID of the Container you are currently logged in to.</li> <li>▪ If set to <code>no</code>, the ID string is not shown on the Container desktop.</li> </ul> <p>By default, this option is set to <code>yes</code>. Keep in mind that you must log off from the Container and log in to it anew for the changes to take effect.</p> <p>To disable the appearance of the ID string for all Containers that you will create on the Hardware Node, specify 0 as the Container ID when executing the <code>vzctl set</code> command.</p>
<code>--showhostname yes no</code>	This setting defines the appearance of the Container hostname string in the top right corner of the Container desktop: <ul style="list-style-type: none"> <li>▪ If set to <code>yes</code>, the hostname string is displayed in the top right corner of the Container desktop helping you identify the hostname of the Container you are currently logged in to.</li> <li>▪ If set to <code>no</code>, the string is not shown on the Container desktop.</li> </ul>

	By default, this option is set to <code>no</code> . Keep in mind that you must log off from the Container and log in to it anew for the changes to take effect.
	To disable the appearance of the hostname string for all Containers that you will create on the Hardware Node, specify <code>0</code> as the Container ID when executing the <code>vzctl set</code> command.
<code>--name</code>	An arbitrary name to assign to the Container. This name can then be used, along with the Container ID, to refer to the Container while performing Container-related operations.
<code>--description</code>	Sets the Container description. You are allowed to use only symbols in the 'A -z' and '0-9' ranges in your descriptions.
<code>--bootorder number</code>	Sets the start-up priority for Containers, starting from 1. The lower number is assigned to the Container, the higher priority it has. Specifying <code>0</code> after <code>--bootorder</code> restores the default start order of the Container (defined by its ID).
<code>--regowner name</code>	Sets the registered owner name for the Container. By default, this name is set to <code>User</code> .
<code>--regorganization name</code>	Sets the registered organization name for the Container. By default, this name is set to <code>Organization</code> .

*Resource management settings* control the amount of resources a Container can consume.

<code>-p, --numproc number</code>	The number of processes allowed to simultaneously run inside the Container. Upon hitting this limit, the Container will not be able to start a new process.
<code>-t, --numsessions number</code>	The number of terminal sessions that can be opened simultaneously to the Container.
<code>--cpuunits units</code>	CPU power. This is a positive integer number that defines how much CPU time one Container will receive in comparison with the other Containers on the Hardware Node in case all the CPUs of the Node are fully used.
<code>--cpuguarantee percent</code>	CPU guarantee. This is a positive integer number that determines the minimal guaranteed share of the CPU time, in percent, the Container is guaranteed to receive.
<code>--cpulimit percent</code>	CPU limit. This is a positive number indicating the CPU time, in percent, the Container is not allowed to exceed. By default, this parameter is disabled for all Containers on the Hardware Node, i.e. any application inside any Container can use all the free CPU power of the Node.
<code>--numa yes no</code>	Enables/disables the Non-Uniform Memory Access (NUMA) support inside the Container.
<code>--diskspace number</code>	The total size of disk space that can be consumed by the Container, in 1 Kb blocks. You can additionally use the <code>--drive</code> option to specify for which Container drive the disk space is to be set. If the <code>--drive</code> option is omitted, the disk space limit is set for the <code>C:\</code> drive.
<code>--drive drive_name</code>	The name of the drive inside the Container for which the disk space limit is to be set. This option should be used in conjunction with the <code>--diskspace</code> option.

---

<code>--vprvmem <i>number</i></code>	The size of private or potentially private memory that can be allocated to all applications inside the Container, in megabytes. Shared or potentially shared memory (e.g., memory mapped files) is not included in this resource parameter.
<code>--disktype <i>plain compact</i></code>	The type of the Container virtual disk drive. You can set the disk type to one of the following: <ul style="list-style-type: none"> <li>▪ <code>plain</code>: in this case, the size of a virtual hard disk is fixed, i.e. all disk space is allocated during the virtual disk creation.</li> <li>▪ <code>compact</code> (default): in this case, you set only the maximal size of a virtual hard disk. A Container hard disk grows in size each time new data is written to the disk and can increase up to the maximum size specified during the disk creation.</li> </ul>
<code>--pagedpoollimit <i>number</i></code>	The amount of paged-pool memory, in megabytes, that can be allocated to the Container.
<code>--nonpagedpoollimit <i>number</i></code>	The amount of non-paged-pool memory, in megabytes, that can be allocated to the Container.
<code>--cpus <i>number</i></code>	If the Hardware Node has more than one CPU installed, sets the number of CPUs to be available to the Container. The changes made to the Container will take effect on the next Container start.
<code>--rate <i>class:Kbits</i></code>	If traffic shaping is turned on, this parameter specifies the bandwidth limit for the Container. The format is <code>class:Kbits</code> where <code>class</code> is the network class (group of IP addresses) and <code>Kbits</code> is the traffic bandwidth.

*Network-related settings* allow you to set the hostname, IP address(es), DNS server address(es), etc. for a Container:

---

**Note:** Most of the network options listed below (`--ipadd`, `--ipdel`, `--nameserver`, `--winserver`, `--mac`, `--gateway`, `--vpn`, `--nat`, etc.) can be used to configure both the default and additional network adapters inside a Container. In the latter case, you need to additionally specify the `--netif` option when running the `vzctl set` command.

---

<code>--hostname <i>name</i></code>	Sets the hostname for the Container.
<code>--ipadd <i>addr</i></code>	Adds the IP address to the list of IP addresses the Container can use and brings up the network interface with this address inside the Container. You can set several IP addresses and separate them by spaces.
<code>--ipadd <i>addr/netmaskbits</i></code>	Along with adding the IP address to the Container, sets its subnet mask.
<code>--ipadd <i>addr:netmask</i></code>	
<code>--ipdel <i>addr all</i></code>	Removes the specified IP address from the Container. Specifying <code>all</code> after the option, removes all IP addresses assigned to the Container.
<code>--nameserver <i>IP_addr</i></code>	Sets the DNS server for the Container. More than one server can be specified in the space-separated format.
<code>--searchdomain <i>domain</i></code>	Sets the DNS search domain for the Container. You can specify several domains and separate them by spaces.

<code>--psched on off</code>	<p>Enables/disables the Quality of Service packet scheduler inside the Container. By default, the scheduler is disabled.</p> <hr/> <p><b>Note:</b> If you have enabled the QoS packet scheduler for a running Container, restart the Container for the changes to take effect.</p> <hr/>
<code>--isolate on off</code>	<p>Defines whether the Container should be isolated within its subnet or can also be accessed from other subnets:</p> <ul style="list-style-type: none"> <li>▪ Use the <code>on</code> command switch to allow the Container to communicate only with the servers from the same subnet.</li> <li>▪ Use the <code>off</code> command switch to enable the Container to access/be accessed by the servers on both the same subnet and other ones.</li> </ul>
<code>--preferred_adapter Adapter_ID</code>	<p>If several network adapter cards are installed on the Hardware Node, you can use this option to specify the preferred network adapter for a Container. Specifying the preferred adapter for a Container means that this network adapter will be used to connect the Container to the network and handle the whole network traffic of the Container.</p> <p>In Parallels-based systems, network adapters are identified by the Media Access Control (MAC) addresses assigned to them. To set a preferred adapter for a Container, specify its MAC address as the value of this option.</p> <p>While working with network adapters in Parallels Virtuozzo Containers-based systems, keep in mind the following:</p> <ul style="list-style-type: none"> <li>▪ After the Parallels Virtuozzo Containers installation, all active adapters on the Hardware Node are used to control network traffic for all Containers on this Node.</li> <li>▪ You can specify several network adapters for a Container and separate them by spaces. In this case, the Container will be able to access all networks to which the specified adapters are connected.</li> <li>▪ If you use the option without specifying any value (i.e. execute the <code>vzctl set CT_ID --preferred_adapter ""</code> command), the network adapters of Container 0 (i.e. of the Node itself) will be used to control network flows for the corresponding Container. If there is no adapter set for Container 0, then all the adapters currently active on the Node will handle the Container network traffic.</li> </ul>
<code>--nettype bridged routed</code>	<p>Sets the operating mode (either bridged or host-routed) for the Container virtual network adapter.</p>
<code>--network network_ID</code>	<p>Connects the Container virtual network adapter to the specified Virtual Network.</p>
<code>--mac XX-XX-XX-XX-XX-XX</code>	<p>The MAC address to be assigned to the Container virtual network adapter.</p>
<code>--gateway addr</code>	<p>Specifies the IP address of the device to be used for routing the traffic from the Container virtual network adapter to external networks.</p>

<code>--winsserver <i>addr</i></code>	The IP address of the WINS server to be used by the Container virtual network adapter.
<code>--vpn on off</code>	Enables/disables the Virtual Private Network (VPN) support by the Container virtual network adapter. By default, the VPN support is disabled.
<code>--nat on off</code>	Enables/disables the Network Address Translation (NAT) functionality for the Container virtual network adapter. By default, NAT is disabled for any newly created Container network adapter.
<code>--cluster_ip <i>IP_addr</i></code>	<p>Sets the virtual (Network Load Balancing cluster) IP address for the Container virtual network adapter. Can be set in one of these forms:</p> <ul style="list-style-type: none"> <li>▪ <code>IP_address</code></li> <li>▪ <code>IP_address:subnet_mask</code></li> <li>▪ <code>IP_address/subnet_mask</code></li> </ul>
<code>--nlb yes no</code>	Enables/disables the support for the Network Load Balancing (NLB) feature inside the Container. By default, the NLB support is disabled.
<code>--nlb_mode multicast unicast</code>	Sets the operating mode of Network Load Balancing inside the Container. By default, the mode is set to <code>multicast</code> .
<code>--tsmode admin app_device app_user</code>	<p>Sets the Terminal Services mode for the Container:</p> <ul style="list-style-type: none"> <li>▪ If set to <code>admin</code>, enables the Remote Desktop for Administration mode for the Container.</li> <li>▪ If set to <code>app_device</code> (or <code>app</code>), enables the device-based licensing scheme of the Terminal Server mode for the Container.</li> <li>▪ If set to <code>app_user</code>, enables the user-based licensing scheme of the Terminal Server mode for the Container.</li> </ul> <p>By default, all newly created Containers is set to work in the Remote Desktop for Administration mode during its creation. Container restart is needed for the new setting to come into effect.</p>
<code>--tslicservers <i>name</i></code>	The NETBIOS name or the IP address of the Terminal Server License (TSL) server to be used by the Container. You can specify several TSL servers and separate them by spaces.
<code>--promiscuous yes no</code>	Enables/disables the promiscuous mode for the Container network adapter.
<code>--broadcasts on off</code>	Enables/disables the Container network adapter to receive network broadcasts. This option is valid only for Containers operating in the host-routed mode.
<code>--failover_cluster yes no</code>	<p>Enables/disables the support for the Microsoft clustering software inside the Container.</p> <p>To enable/disable the possibility of using the clustering software for all Containers on the Hardware Node at once, specify 0 as <code>&lt;CT_ID&gt;</code>.</p>
<code>--scsi yes no</code>	<p>Enables/disables SCSI-aware drivers to load inside the Container.</p> <p>To enable/disable the loading of SCSI-aware drivers for all</p>

	Containers on the Hardware Node at once, specify 0 as <code>&lt;CT_ID&gt;</code> .
<code>--dhcp on off</code>	When set to <code>on</code> , enables the Container network adapter to get IPv4 configuration settings via the DHCP protocol. By default, this option is set to <code>off</code> .
<code>--dhcp_ipv6 on off</code>	When set to <code>on</code> , enables the Container network adapter to get IPv6 configuration settings via the DHCP protocol. By default, this option is set to <code>off</code> .
<code>--dns_suffix suffix</code>	Sets the Domain Name System (DNS) suffix for the Container.
<code>--iplock on off</code>	Enables/disables the possibility of configuring IP address settings and DHCP support from inside the Container.
<code>--netcfglock on off</code>	Enables/disables the possibility of configuring WINS server and DNS suffix settings from inside the Container.
<code>--netif_add name</code>	Creates a new virtual network adapter for the Container with the name specified.
<code>--netif_del name/GUID</code>	Removes the virtual network adapter with the specified name or GUID (globally unique identifier) from the Container.
<code>--netif name</code>	Specifies the name of the Container virtual network adapter whose settings you want to configure. Omit this option if you have only one network adapter inside a Container.

## vzctl exec, vzctl exec2, and vzctl enter

These commands are used to run arbitrary commands inside a Container. The syntax of these commands is as follows:

```
vzctl {exec, exec2} <CT_ID> <command>
vzctl enter <CT_ID>
```

where *command* is a string to be executed in the Container.

The difference between `exec` and `exec2` is the exit code. `vzctl exec` returns 0 in case `vzctl` has been able to launch the command and does not take into account the exit code of the command itself. `vzctl exec2` returns the exit code of the command executed in the Container.

`vzctl enter` is similar to `vzctl exec`. The difference between the two is that `vzctl enter` makes the command line believe that it is connected to a terminal. As such, you receive a command line prompt and are able to execute multiple commands as if you were logged in to the Container.

## vzctl mountext and vzctl umountext

The `vzctl mountext` command is used to mount a drive on the Hardware Node to a drive or folder inside a Container. It has the following syntax:

```
vzctl mountext CT_ID --srcdir path --dstdir path
```

The command requires the following arguments:

Name	Description
<code>CT_ID</code>	The ID of the Container where to mount the drive/folder.
<code>--srcdir path</code>	The full path to the drive/folder inside the Container where to mount the drive. The drive from the Node can be mounted to any drive/folder inside the Container. If the specified drive/folder does not exist, it is automatically created.
<code>--dstdir path</code>	The full path to the drive on the Node.

## vzctl partadd and vzctl partdel

The `vzctl partadd` command is used to create a new drive inside a Container by mounting it to a loopback file on the Hardware Node. The `vzctl partadd` command has the following syntax:

```
vzctl partadd CT_ID --drive CT_drive [--size drive_size]
[--file File_Name]
```

The options that can be passed to `vzctl partadd` are explained in the table below:

Option	Description
<code>--drive CT_drive</code>	Mandatory. The name to assign to the drive in one of these formats: <code>X</code> , <code>X:</code> , <code>X:\</code> .
<code>--size drive_size</code>	Optional. The size of the drive, in kilobytes, to be created inside the Container.
<code>--file File_Name</code>	Optional. The name of the file to be created on the Node. Upon the command completion, this name will be visible as the name of the file in the <code>.efd</code> format and located in the <code>X:\vz\private\CT_ID</code> folder on the Node. If you omit this option, the disk will be assigned a name in the <code>lpbkXXXXX.efd</code> format.

The `vzctl partdel` command is used to unmount the Container drive mounted by means of the `vzctl partadd` command from a file on the Hardware Node. It has the following syntax:

```
vzctl partdel CT_ID --drive CT_drive [--delete]
```

The command requires only the `--drive` option to be specified. However, you can also use the `--delete` option to remove the file from the `X:\vz\private\CT_ID` folder on the Node which, otherwise, remains intact.

## vzctl reinstall

The `vzctl reinstall` command is used to restore the original state of the Container system and application files. The Container is restored to the state as it was at the time of its creation and/or when other applications were added to the Container afterwards.

The syntax of this command is as follows:

```
vzctl reinstall <CT_ID> [options]
```

The available options are listed below:

Option	Description
<code>--resetpwdb</code>	Removes the Administrator credentials from the users' database and creates a clean database as for any new installation.
	<b>Note:</b> During the Container reinstallation, the user login name is always reset to Administrator regardless of the presence of this option.
<code>--skipbackup</code>	The contents of the old private area are not saved in the <code>reinstall</code> folder.
<code>--skipapps</code>	Skips reinstalling the application templates inside the Container.

When executed, the `vzctl reinstall` command creates a new private area for the Container and rewrites the Container from scratch using its configuration file (thus retaining the Container IP address, hostname, resource control parameters, and all the other settings). The contents of the Container old private area are then copied to the `C:\reinstall` folder in the new private area, to retain the user files.

## vzctl shrink

The `vzctl shrink` command is used to shrink compact virtual disk drives inside Containers. It has the following syntax:

```
vzctl shrink CT_ID --drive CT_drive_name
```

The command removes unused space from the specified virtual disk, thus reducing the amount of space these virtual disks occupy on the Hardware Node.

When using this command, keep in mind the following:

- You can shrink virtual disks inside both running and stopped Containers.
- Shrinking a Container virtual disk does not reduce the maximum capacity of the virtual disk itself.
- You cannot shrink plain virtual disks.

## vzctl defrag

The `vzctl defrag` command is used to check the degree of the virtual disk fragmentation and, if necessary, defragment the disk. It has the following syntax:

```
vzctl defrag CT_ID --drive drive [--force] [--analysis]
```

The options that can be used with `vzctl defrag` are explained in the following table:

Name	Description
<code>CT_ID</code>	The ID of the Container that owns the virtual disk drive to defragment/analyze.
<code>--drive drive</code>	The name of the drive to defragment/analyze. A name can be specified in one of the following formats: <code>X</code> , <code>X:</code> , <code>X:\</code> .
<code>--force</code>	Forces the disk defragmentation. You can use this option if you do not have enough free space on the virtual disk. Keep in mind, however, that forcing a defragmentation when there is insufficient disk space may result in partial disk defragmentation.
<code>--analysis</code>	Analyze the virtual disk without performing its defragmentation.

## vzctl addrole, vzctl delrole, vzctl enumrole

These commands are used to manage server roles, role services, and features inside Containers running the Windows Server 2008 or Windows Server 2008 R2 operating system.

### **vzctl addrole**

This command is used to add a new role to a Container. It has the following syntax:

```
vzctl addrole CT_ID --role role_name [role_name] ... [--restart]
```

You can install several roles inside the Container at once by specifying their names after `--role` and separating them by spaces. If a Container restart is required to complete the feature installation, you can additionally specify the `--restart` option. In this case, the Container will be automatically restarted once the feature installation is complete.

### **vzctl delrole**

This command is used to remove an installed role from a Container. It has the following syntax:

```
vzctl delrole CT_ID --role role_name [role_name] ... [--restart]
```

You can remove several roles from the Container at once by specifying their names after `--role` and separating them by spaces. If a Container restart is required to complete the operation, you can additionally specify the `--restart` option. In this case, the Container will be automatically restarted once the feature is removed from the Container.

### **vzctl enumroles**

This command lists the server roles already installed or ready to be installed inside a Container. It has the following syntax:

```
vzctl enumroles CT_ID
```

## vzlist

The `vzlist` utility is used to list the Containers existing on the Hardware Node together with additional information about them. It has the following syntax:

```
vzlist [-a] [-S] [-n] [-o specifier[,specifier...]] [-H] [CT_ID [CT_ID...]]
        [-s specifier | -specifier] [-h pattern] [-N pattern] [-d pattern]
        [--quiet] [--verbose]
vlist -L
vlist --help
```

The options that can be used with `vzlist` are explained in the following table:

Option	Description
<code>-a, --all</code>	List all the Containers existing on the Hardware Node. By default, only running Containers are shown.
<code>-S, --stopped</code>	List only stopped Containers.
<code>-o</code> <code>parameter[.specifier]</code>	Display only particular information about the Containers. The parameters and their specifiers that can be used after the <code>-o</code> option are listed in the following subsection. To display a number of parameters in a single output, separate them with commas.
<code>-H, --no-header</code>	Do not display column headers.
<code>-L, --list</code>	List all parameters for <code>vzlist</code> .
<code>--help</code>	Display the <code>vzlist</code> usage information and exit.
<code>-s, --sort</code> <code>[-</code> <code>]parameter[.specifier]</code>	Sort the Containers in the list by the specified parameter. If "-" is given before the name of the parameter, the sorting order is reversed.
<code>-h, --hosthame pattern</code>	Display only those Containers whose hostnames correspond to the specified pattern. The following wildcards can be used: *.
<code>-n, --name</code>	If used without any parameters, displays information on all Containers on the Node together with their names. If you indicate the Container ID after this option, displays information on the specified Container only.
<code>-N, --name_filter</code> <code>pattern</code>	Displays only the Container that corresponds to the specified pattern.
<code>-d, --description</code> <code>desc_pattern</code>	Displays only the Container whose description corresponds to the specified pattern.
<code>--quiet</code>	Disables logging to the display and to the log file.
<code>--verbose</code>	Sets the log level to the maximum possible value.

## vzlist Output Parameters and Their Modifiers

Some parameters that can be used after the `-o` and `-s` switches of the `vzlist` utility can be specified by the "dot+letter" combination following the parameter and denoting one of the following things:

Specifier	Description
<code>.b</code>	The barrier on using the corresponding resource set for the given Container.
<code>.h</code>	The hard limit on using the corresponding resource set for the given Container.

The following parameters are available for using with the utility:

Parameter	Output Column	Description
<code>ctid</code>	CTID	The Container ID.
<code>hostname</code>	HOSTNAME	The Container hostname.
<code>ip</code>	IP_ADDR	The Container IP address.
<code>status</code>	STATUS	Specifies whether the Container is running or stopped.
<code>templates</code>	TEMPLATES	Displays the OS and application templates applied to the Container. If the <code>ctid</code> parameter is not specified, all templates available on the Hardware Node are shown.
<code>rates</code>	RATES	The bandwidth limit set for the Container.
<code>name</code>	NAME	The name assigned to the Container.
<code>description</code>	DESCRIPTION	The Container description.
<code>numproc</code>	NPROC	The number of processes allowed to simultaneously run in the Container. Can be used with the <code>.b</code> specifier.
<code>numsessions</code>	SESSIONS	The number of terminal sessions opened to the Container. Can be used with the <code>.b</code> specifier.
<code>diskspace</code>	DQBLOCKS	The total size of disk space consumed by the Container, in 1 Kb blocks. Can be used with the <code>.h</code> specifier defining the hard limit on using disk space.
<code>cpuunits</code>	CPUUNI	Allowed CPU power. This is a positive integer number that defines how much CPU time one Container will receive in comparison with the other Containers on the Hardware Node in case all the CPUs of the Node are fully used.
<code>cpuguarantee</code>	CPUGUARANTEE	CPU guarantee. This is a positive integer number that determines the minimal guaranteed share of the CPU time, in percent, the Container is guaranteed to receive.

cpulimit	CPULIMIT	CPU limit. This is a positive number indicating the CPU time, in percent, the Container is not allowed to exceed.
cpus	CPUS	The number of CPUs available to the Container.
cpupool	CPUPOOL	The CPU pools assigned to the Container.
vprvmem	VPRVMEM	The size of private or potentially private memory that can be allocated to all applications inside the Container, in megabytes. Shared or potentially shared memory (e.g., memory mapped files) is not included in this resource parameter.
pagedpool	PAGEDPOOL	The amount of paged-pool memory, in megabytes, that can be allocated to the Container.
disktype	DISKTYPE	The type of the Container virtual disk drive. It can be one of the following: <ul style="list-style-type: none"> <li>▪ <code>plain</code>. The size of a virtual hard disk is fixed and all disk space is allocated during the virtual disk creation.</li> <li>▪ <code>compact</code>. Only the maximal size of a virtual hard disk is set. The Container hard disk grows in size each time new data is written to the hard disk and can increase up to the maximum size specified during the hard disk creation.</li> </ul>
bootorder	BOOTORDER	The start-up priority set for the Container. The lower number is assigned to the Container, the higher priority it has.
regowner	REGOWNER	The registered owner name set for the Container.
regorganization	REGORGANIZATION	The registered organization name set for the Container.
flags	FLAGS	Displays miscellaneous options (offline management, the support for using the Microsoft Cluster Server software and loading SCSI-aware drivers, etc.) currently enabled for the Container.
network	NETWORK	The Virtual Network to which the Container virtual adapter is connected.
nettype	NETTYPE	The network operating mode set for the default Container.
gateway	GATEWAY	The IP address of the device used for routing the traffic from the Container to external networks.
nameserver	NAMESERVER	The DNS servers set for the Container.
winsserver	WINSSERVER	The WINS server set for the Container.
clusterip	CLUSTERIP	The virtual (Network Load Balancing cluster) IP address assigned to the Container.
ifname	INTERFACES	The list of virtual network adapters available inside the Container.
tsmode	TSMODE	The Terminal Services mode set for the Container: <ul style="list-style-type: none"> <li>▪ <code>admin</code>: the Remote Desktop for Administration mode is currently enabled for</li> </ul>

the Container.

- `app_device` (or `app`): the device-based licensing scheme of the Terminal Server mode is currently enabled for the Container.
- `app_user`: the user-based licensing scheme of the Terminal Server mode is currently enabled for the Container.

If a parameter that can be used with a specifier is used without any specifier in the command line; the current usage of the corresponding resource is shown by default.

---

## vzquota

This command is used to configure and see Container disk quota statistics. `vzctl` uses `vzquota` internally to configure Container quotas and you usually do not have to use `vzquota` except for checking the current quota statistics. The syntax of `vzquota` command is as follows:

```
vzquota [--quiet | --verbose] [-b] setlimit CT_ID -B num [--drive name]
vzquota [--quiet | --verbose] [-b] stat CT_ID
vzquota [--quiet | --verbose] [-b] show CT_ID
vzquota --help
```

General options available to all `vzquota` commands are:

- `-v` Verbose mode. Causes `vzquota` to print debugging messages about its progress. You can give up to two `-v` switches to increase verbosity.
- `-q` Quiet mode. Causes all warning and diagnostic messages to be suppressed. Only fatal errors are displayed.
- `--help` Displays the `vzquota` usage information and exits.
- `-b` Sets the batch mode.

`vzquota` understands the following commands:

- `setlimit` Configures the quota limit for the running quota.
- `stat` Shows quota statistics for the running quota.
- `show` Shows quota usage from the quota file.

## vzquota setlimit

This command updates limits for the running Container quota. It requires at least one limit to be specified. The syntax of this command is:

```
vzquota setlimit CT_ID -B num [--drive name]
```

The following options can be used with the command:

- |  |  |
|--|--|
| <code>-B, --block-hardlimit num</code> | Required. Specifies the disk quota block hard limit, in 1 Kb blocks. This limit cannot be exceeded by the Container. |
| <code>--drive name</code>              | Optional. Specifies the disk drive inside the Container for which the disk quota limit is to be set.                 |

For its execution, `vzquota setlimit` requires only the ID of the Container and the disk quota limit to be specified. However, you can additionally use the `--drive` option to specify to what Container disk drive the defined quota limit is to be applied.

---

**Note:** The disk quota limit for all Container disk drives except for the `C:\` drive is set on the fly. To make the disk quota limit changes apply to the `C:\` drive, restart the Container where the drive is located.

---

## vzquota stat and vzquota show

These commands are used for querying the Container quota statistics. The syntax is as below:

```
vzquota show CT_ID  
vzquota stat CT_ID
```

You can use both the `vzquota stat` and `vzquota show` commands to view disk quota parameters.

## CHAPTER 3

# Licensing Utilities

This chapter provides the information on utilities for managing Parallels Virtuozzo Containers licenses.

## In This Chapter

vzlicload .....	34
vzlicview .....	35

---

## vzlicload

This utility is used for managing Parallels Virtuozzo Containers licenses. It has the following syntax:

```
vzlicload [options]
```

The utility accepts the following options:

<code>-p, --product-key</code>	Installs the Parallels Virtuozzo Containers license on the Hardware Node.
<code>-i, --stdin</code>	Makes <code>vzlicload</code> use standard input while installing the Parallels Virtuozzo Containers license on the Hardware Node.
<code>-f, --license-file</code> <code>&lt;file_path&gt;</code>	The full path to the license file containing the Parallels Virtuozzo Containers license to be installed on the Hardware Node.
<code>-u, --update</code>	Connects to the Parallels Key Authentication (KA) server and updates the license installed on the Hardware Node.
<code>-t, --transfer</code>	Used to transfer the Parallels Virtuozzo Containers license from one Hardware Node (Source Node) to another (Destination Node). You should use this option along with the <code>-p</code> option when running the <code>vzlicload</code> utility on the Destination Node.
<code>-r, --remove</code>	Removes the Parallels Virtuozzo Containers license with the specified serial number from the Hardware Node. You can find out the license serial number using the <code>vzlicview</code> utility (please see the <code>vzlicview</code> subsection (p. 35)).
<code>-A, --proxy-server</code>	The IP address or hostname of the proxy server, if you use any, to connect to the Internet. By default, the Internet Explorer proxy settings are used.
<code>-U, --username</code>	The user name to log in to the proxy server.
<code>-P, --password</code>	The password of the user specified after the <code>-U</code> option.
<code>-h, --help</code>	Displays the utility usage and exits.

---

## vzlicview

This utility displays the license contents along with license status information. It has the following syntax:

```
vzlicview [options]
```

The following options can be used with this utility:

- |  |  |
|--|--|
| <code>-f, --license-file &lt;file&gt;</code> | Displays the license information from the specified Parallels Virtuozzo Containers license file.   |
| <code>-p, --product-key</code>               | Displays the license information contained in the specified Parallels Virtuozzo Containers product key.  |
| <code>-i, --stdin</code>                     | Makes <code>vzlicview</code> use standard input as a license and display its information.  |
| <code>-v, --validate</code>                  | Displays the total number of Containers which may simultaneously run on the Hardware Node in accordance with the license currently installed on this Node. |
| <code>-h, --help</code>                      | Display the utility usage and exit.  |

The utility reports the following statuses for the license:

- |         |  |
|---------|--|
| ACTIVE  | The license is valid.  |
| INVALID | The license is invalid (for example, because of the Hardware Node ID mismatch) or corrupted.                       |
| EXPIRED | The license file matches the Hardware Node ID, but has expired.  |
| GRACED  | The license has expired and is currently on the grace period (i.e. it is active till the end of the grace period). |

## CHAPTER 4

# Migration Utilities

This chapter focuses on utilities for migrating Containers between Hardware Nodes or within one Hardware Node.

## In This Chapter

vzmigrate.....	37
vzp2v.....	39
vzmlocal.....	40

## vzmigrate

This utility is used for moving Containers to another system (i.e. from the Source Node to the Destination Node) with minimal downtime. It has the following syntax:

```
vzmigrate [-r yes|no] [-n] -A ServerNodeAddr \
-P ServerPassword {CT_list}
{CT List} = <source_CTID>[:<dest_CTID>] [... ]
vzmigrate --help
```

The following options can be used with the `vzmigrate` utility:

<code>-r, --remove-area yes no</code>	If “yes” is specified, then the private area will be deleted after the corresponding Container is successfully migrated to the Destination Node. If “no” is specified, the private area will not be deleted. In this case the Container configuration file will be renamed from <code>CT_ID.conf</code> to <code>CT_ID.conf_migrated</code> . By default, the Container private area is removed from the Source Node.
<code>-n, --nostart</code>	Do not attempt to start the Container 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.
<code>-A, --srv_addr</code>	The IP address or the hostname of the Destination Node, i.e. of the Node where the Container is to be migrated.
<code>-P, --srv_pswd</code>	The password of the <code>vzadmin</code> user. You should provide the password you use to log in as <code>vzadmin</code> to the Service Container on the Destination Node.
<code>-h, --help</code>	Displays the utility usage and exits.
<code>--quiet</code>	Disables logging to the screen and to the log file.
<code>--verbose</code>	Sets the log level to the maximum possible value for this <code>vzmigrate</code> session.

To migrate the Container, execute the `vzmigrate` command on the Source Node and pass the corresponding options to it. `{CT_list}` is a list of `<source_CTID>[:<dest_CTID>]` denotations separated by spaces. A different `dest_CTID` parameter is needed in case the Source Node (the one where you run the `vzmigrate` command) and the Destination Node have Containers with one and the same ID. You can specify multiple Containers at once for migration.

If the Container is running on the Source Node, `vzmigrate` copies its private files, registry settings, etc. to the Destination Node, stops the Container on the Source Node, copies the files changed after the first copying to the Destination Node again, and starts the Container on the Destination Node (if the `-n` option is not specified).

Since a Container consists of thousands of files, copying all of them to a new Hardware Node may take considerable time. However, `vzmigrate` stops the Container only after all the files have been copied to the new Hardware Node and resynchronizes only those files that have been changed while being copied. This brings down a typical Container downtime to the time required for restarting a Container.

For example, to migrate Container 101 residing on the Source Node to Container 111 on the Destination Node having the IP address of 123.154.154.123, you can execute the following command:

```
C:\...\Administrator>vzmigrate -A 123.154.154.123 -P qw3r5f 101:111
```

## vzp2v

vzp2v is used to migrate a physical server to a Container on your Hardware Node. It has the following syntax:

```
vzp2v CT_ID -A Physical_Server_IP_Address -U username -P password
      [-s] [-v] [-x disk_drive [...]] [-b Virtual_Network_ID]
vzp2v --help
```

The options that can be used with the vzp2v utility are listed in the table below:

Name	Description
-A, --src_addr <i>Physical_Server_IP_Addres s</i>	Mandatory. The IP address or hostname of the physical server you are going to migrate to a Container on your Node.
-U, --src_user <i>username</i>	Mandatory. The username to log in to the physical server. The specified user should have the Administrator rights to the server.
-P, --src_pswd <i>password</i>	Mandatory. The password of the user specified as the value of the -U option.
-s, --src_stop	Optional. Stop the physical server after its successful migration. If this option is omitted, the physical server continues running upon the migration completion.
-v, --ve_start	Optional. Start the Container on the Hardware Node after the physical server has been successfully migrated. If the option is not specified, the Container is not started after the physical server migration.
-x, --exclude <i>disk_drive</i>	Optional. The name of the disk drive that you do not wish to move to the Container. The formats that can be used for specifying the drive name are <i>X</i> , <i>X:</i> or <i>X:\</i> where <i>X</i> denotes the name of the drive that is not to be moved to the Container. You can specify several drives to be excluded from the migration process and separate them by spaces. If you omit this option, all disk drives available on the physical server will be migrated to the Container.
-b, --bridged <i>Virtual_Network_ID</i>	Optional. If one or more network adapters on the physical server use the DHCP protocol to receive their IP address, you should specify this option to denote the Virtual Network on the Hardware Node where these network adapters are to be connected. If you omit this option, you will need to manually configure the Container network adapter parameters for it to be visible on your network.  Detailed information on Virtual Networks is provided in the <i>Parallels Virtuozzo Containers 4.6 User's Guide</i> .
--help, /?	Display the utility usage.

To migrate a physical server, you should execute the vzp2v command on the Hardware Node and pass the corresponding options to it. Before starting the migration process, please make sure that there is no Container on your Node with the ID you are going to specify as *CT\_ID*; otherwise, the migration will fail.

For example, the following command

```
C:\...\Administrator>vzp2v 111 -A 123.154.154.123 -U Administrator \
-P qw3r5f -s -v -x E:
```

allows you to log in to the physical server having the IP address of 123.154.154.123 with the credentials of Administrator and qw3r5f and to migrate this physical server to Container 111 on your Hardware Node. The `-s` and `-v` options tell `vzp2v` to stop the physical server and to start the Container on the Hardware Node after the migration has successfully completed and the `-x` option excludes the `E:\` disk drive on the physical server from the migration process.

---

## vzmlocal

Moving/copying a Container within one and the same Hardware Node consists in changing/adding the Container ID and private area path. Thus, you can use the `vzmlocal` utility either to change the ID and/or the private area path of any existing Container or to clone a Container. The utility has the following syntax:

```
vzmlocal <CT List>
<CT List> = <source_CTID>:<dest_CTID>[:<dest_private>] [...]
vzmlocal -C <CT List>
<CT List> = <source_CTID>:<dest_CTID>[:<dest_private>] [...]
vzmlocal --help
```

The options are the following:

<code>-C, --copy</code>	Clones the source Container instead of moving it.
<code>-s, --fast-sid</code>	Allows you to speed up the process of cloning the Container.
<code>-d, --destroy-source</code>	Destroys the source Container after its cloning.
<code>-n, --disable-network</code>	Disables offline management for the source Container after its cloning.
<code>-l, --skiplock</code>	Allows you to clone locked Containers.
<code>--quiet</code>	Disables logging to screen and to the log file.
<code>--verbose</code>	Sets log level to maximum possible value for this <code>vzmlocal</code> session.
<code>--help</code>	Displays the utility usage and exits.

You need to specify the source Container ID (`<source_CTID>`) and the destination Container ID (`<dest_CTID>`). Specifying the destination Container private area path (`<dest_private>`) is optional; it allows you to override the default path of `X:\vz\private\<CT_ID>`.

## CHAPTER 5

# Backing-Up Utilities

This chapter describes utilities for backing up and restoring Container private areas and configuration files.

## In This Chapter

vzabackup.....	42
vzarestore .....	44
vzvssctl.....	45

---

## vzabackup

The `vzabackup` utility is used to create backups of separate Containers or whole Hardware Nodes. It can be run on one of the following:

- Source Node where the Container to be backed up is residing.
- Backup Node, a special Node intended for storing Container backups.
- Any other Parallels Virtuozzo Containers-based physical server in your network.

The only requirements that should be met to execute `vzabackup` is to have a server with Parallels Virtuozzo Containers and the Parallels Agent software installed on it and to provide the network connectivity for this server to be able to establish connections to the Source and Backup Nodes, if necessary. The created Container backups are then stored on the Backup Node which can also be presented by any Node with running the Parallels Virtuozzo Containers and Parallels Agent software.

The `vzabackup` utility has the following syntax:

```
vzabackup [BACKUP_OPTIONS] --node NODE1 [CT_OPTIONS] ...
```

The general backup options (`[BACKUP_OPTIONS]`) are the following:

- |                                    |  |
|------------------------------------|--|
| <code>-F, -I, --TFull</code>       | Make a full backup. By default, <code>vzabackup</code> creates a full backup of Containers and Hardware Nodes.   |
| <code>-i, --Tinc</code>            | Make an incremental backup or, if no full backups are available, a full backup. If this option is omitted, the full backup is created.   |
| <code>--Tdiff</code>               | Make a differential backup or, if no full backups are available, a full backup. If this option is omitted, the full backup is created.   |
| <code>-D backup_description</code> | The description of the backup archive.   |
| <code>-o, --rm-old</code>          | Create a new backup and then remove the oldest backup of the specified Node/Container.   |
| <code>--rm-tag backup_ID</code>    | Create a backup and then remove the backup with the specified ID. You can learn what ID is assigned to what Container backup using the <code>-l</code> and <code>-f</code> options of the <code>vzarestore</code> utility.   |
| <code>-C(Letter)</code>            | Indicates the level of compression for the resulting Container backup archive. In the current version of Parallels Virtuozzo Containers, you can set this level to one of the following: <ul style="list-style-type: none"> <li>▪ <code>n</code>: create the Container backup without any compression. Using this level of compression may speed up the backing up time; however, it may significantly increase the size of the resulting backup file.</li> <li>▪ <code>g</code>: compress the resulting backup with the normal level of compression. This is the default level of compression used to back up all Nodes/Containers.</li> <li>▪ <code>b</code>: compress the resulting backup with the maximum level of compression. In this case the backup file size is the smallest; however, it may take much time to create the backups.</li> </ul> |

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 (e.g. all executable files with the .exe extension or image files with the .jpg, .jpeg, and .gif extensions).
- J If several Source Nodes are specified, back up these Nodes and/or their Containers simultaneously. If the option is omitted, the Nodes are backed up sequentially one after another.
  - force Force the process of backing up the Hardware Node(s)/Container(s). Can be used when backing up several HNs/Containers to continue the backup process even if some errors occur when creating a backup of a certain Node/Container.
  - storage *BACKUP\_NODE* The IP address and the credentials of the Backup Node. Should be specified in the following form: *USER[:PASSW]@IP\_ADDRESS*. If you do not indicate the password to log in to the Backup Node, you will be asked to do so during the *vzabackup* execution.  
  
If this option is omitted, *vzabackup* puts the created backup(s) to the backup folder on the Source Node (by default, this folder is *X:\vz\Backups*).
  - NodeX* The IP address and the credentials of the Source Node. Should be specified in the following form: *USER[:PASSW]@IP\_ADDRESS*. If you do not indicate the credentials to log in to the Source Node, you will be asked to do so during the *vzabackup* execution.  
  
You can omit this option when backing up particular Containers on the local Source Node (provided you are logged in to the Node as Administrator); in all other cases specifying the Source Node IP address and credentials is mandatory.
  - q, --no-progress Disables logging to the screen during the *vzabackup* operation.

The Container options (*[CT\_OPTIONS]*) define the list of Containers to be backed up:

- e *CT1...* The Containers to back up on the Source Node. If this option is omitted, all Containers on the given Node will be backed up. Containers can be specified using both their IDs (e.g. 101 or 102) and their names (e.g. *comp1* or *comp2*).
- x *CT1...* The Containers that need not be backed up (Containers to exclude). If this option is omitted, all Containers on the given Source Node will be backed up. Containers can be specified using both their IDs (e.g. 101 or 102) and their names (e.g. *comp1* or *comp2*).
- include-files *files\_list* Only the specified files and folders will be included in the Container backup.

---

**Note:** If you create a Container backup using this option, you will be able to restore only separate files from the resulting Container backup, but not the Container as a whole.

---

- exclude-files *files\_list* The path to the files and folders inside the Container to be excluded from the backup.

## vzrestore

The `vzrestore` utility is used to manage Container backups: restore a Container or certain Container files/folders from the Container backup archive, list the backups existing on the Backup Node, remove backups, etc. `vzrestore` can be run on any Hardware Node provided this Node has the Parallels Agent software installed. The utility has the following syntax:

```
vzrestore [CT | -e <CT1, ...> [-x <CT1, ...>] ] [RESTORE_OPTIONS]
          [BACKUP_NODE]
vzrestore -r,--remove <BACKUP_ID ...>
vzrestore -l,--list [LIST_OPTIONS] [BACKUP_NODE]
vzrestore --browse BACKUP_ID [BROWSE_OPTIONS] [BACKUP_NODE]
vzrestore --print-ct-config BACKUP_ID [BACKUP_NODE]
vzrestore --help
```

The restore options are the following:

<code>-e CT1, ...</code>	The Containers to be restored on the Destination Node. Any Container can be specified using both its IDs (e.g. 101 or 102) and its names (e.g. comp1 or comp2).
<code>-x CT1, ...</code>	The Containers that need not be restored (the Containers to exclude). Any Container can be specified using both its IDs (e.g. 101 or 102) and its names (e.g. comp1 or comp2).
<code>-b BACKUP_ID</code>	The ID assigned to the Container backup. This ID can be used to manage the backup archive (e.g. to delete the Container backup from the Backup Node or restore the files from the Container backup with the specified ID). If this option is omitted, the latest Container backup is used.
<code>--force</code>	Do not stop on errors during the <code>vzrestore</code> execution. Can be used when you are restoring more than one Container. This option allows you to continue the restoring process if some errors occur when restoring a certain Container.
<code>--skip-ct-config</code>	Do not restore the Container configuration file. Can be used only if you are restoring a single Container.

---

**Note:** The Container configuration file is not changed when restoring separate Container files.

---

<code>--files PATH_TO_FILES</code>	The full path to the files/folders inside the Container to be restored. This options is incompatible wit the <code>-e</code> and <code>-x</code> options.
<code>--skip-locked</code>	Do not stop on errors even if some of the files to be restored are in the 'locked' state.
<code>-B</code>	Handle the values after the <code>-e</code> and <code>-x</code> options as backup IDs.
<code>--storage BACKUP_SERVER</code>	The IP address and the credentials of the Backup Node. Can be specified in the following form: <code>USER[:PASSW]@IP_ADDRESS</code> . If this options is omitted, <code>vzrestore</code> looks for the Container backup on the local Node.

The options which can be used along with the `--list` option of `vzrestore`:

<code>f, --full</code>	Display detailed information on the backed up Containers.
<code>--latest</code>	Display the latest Container backup.

- 
- e *CT1*, ... Display the information on the backups for the specified Containers only.
  - B Handle values after the -e option as backup IDs.

Other options which can be used with `vzrestore`:

- d The path to the folder inside the Container backup archive whose contents is to be shown. This option can be used with the `vzrestore --browse BACKUP_ID` command only.
- print-ct-config *BACKUP\_ID* Displays the parameters set in the Container configuration file at the moment of creating the backup with the specified backup ID.
- r, --remove *BACKUP\_ID* Removes the Container backup with the specified backup ID.

---

## vzvssctl

The `vzvssctl` utility is used to suspend and resume the Container activity (e.g. during the backup operation). This utility has the following syntax:

```
vzvssctl -s [CT_ID] [...]  
vzvssctl -r [CT_ID] [...]  
vzvssctl --help
```

`vzvssctl` can be used with the following options:

- s Suspends the Container activity.
- r Resumes the Container activity.
- help, /? Displays the utility usage and exits.

## CHAPTER 6

# Template Management Utilities

This chapter focuses on utilities for creating and managing templates.

## In This Chapter

vzpkgls .....	46
vzpkgdeploy .....	47
vzpkgadd .....	47
vzpkgrm .....	48

---

## vzpkgls

This utility lists the OS and application templates installed on the Hardware Node or already applied to a particular Container. It has the following syntax:

```
vzpkgls [options] [<CT_ID> ...]
```

If you specify one or more Container IDs for this command, it lists the templates applied to these Containers. Without the `<CT_ID>` argument, the utility lists all templates available for Containers on the Hardware Node. Other options available to the `vzpkgls` command are listed below:

<code>-o, --os</code>	Lists all OS templates installed on the Hardware Node.
<code>-a, --application</code>	Lists all application templates installed on the Hardware Node.
<code>-s, --separate</code>	By default, <code>vzpkgls</code> outputs the template and all its updates on a single line separated by a space. This option changes the output and prints a separate line for each available version of the template.
<code>--quiet</code>	Disables logging to the display and to the log file.
<code>--verbose</code>	Sets the log level to the maximum possible value for this <code>vzpkgls</code> session.
<code>-h, --help</code>	Displays the usage information and exits.

---

## vzpkgdeploy

This utility is used for installing/removing OS and application templates on the Hardware Node. It has the following syntax:

```
vzpkgdeploy [-q|-v] --install|--uninstall PACKAGE...
vzpkgdeploy --help
```

The options available to this command are:

<code>-i, --install</code> <code>PACKAGE...</code>	Installs the template on the Hardware Node.
<code>-u, --uninstall</code> <code>PACKAGE...</code>	Removes the template from the Hardware Node.
<code>-h, --help</code>	Displays the usage information and exit.
<code>--quiet</code>	Disables logging to the display and to the log file.
<code>--verbose</code>	Sets the log level to the maximum possible value for this <code>vzpkgdeploy</code> session.

---

## vzpkgadd

This utility is used to add an application template to a Container. It has the following syntax:

```
vzpkgadd [options] <CT_ID> PACKAGE[/VERSION]...
```

This command will add a template (`<template>`) to the Container with the ID of `<CT_ID>`. The `<version>` parameter specifies the template version to use if there are available upgrades; by default, the latest available version is used.

You may specify a number of templates for adding to a Container.

Options available to this command are:

<code>-h, --help</code>	Display the usage info and exit.
<code>-f, --force</code>	Force template installation.
<code>--quiet</code>	Disables logging to the display and to the log file.
<code>--verbose</code>	Sets the log level to the maximum possible value for this <code>vzpkgadd</code> session.

A Container has to be running in order to apply a template to it.

---

## vzpkgrm

This utility is used to uninstall an application template from a Container. It has the following syntax:

```
vzpkgrm [options] <CT_ID> PACKAGE[/VERSION]...
```

This command will remove a template from the Container with the ID of <CT\_ID>. You may specify a number of templates for uninstalling.

You can use the following options with this command:

-h, --help	Display the usage info and exit.
-f, --force	Force the template uninstallation from the specified Container.
--quiet	Disables logging to the display and to the log file.
--verbose	Sets the log level to the maximum possible value for this vzpkgrm session.

A Container has to be running in order to uninstall a template from it.

## CHAPTER 7

# Supplementary Utilities

This chapter concentrates on utilities for performing different tasks in the Hardware Node and Container context.

## In This Chapter

vzkeygen .....	49
vzcache.....	50
vzlscache .....	51
vzuncache.....	52
vznetcfg.....	53
vznetstat .....	54
vzdevctl .....	55
vzcpucfg.....	57
vzquery.....	58
vzwinupdatecmd .....	59

---

## vzkeygen

This utility is used to generate the Hardware Node ID to uniquely identify the Node. You should send your Node ID together with a license request to the Parallels sales department to obtain a license for running Parallels Virtuozzo Containers. It has the following syntax:

```
vzkeygen [-n, --company-name NAME] [-o, --output-file FILENAME]
vzkeygen [-h, --help]
```

When using the utility without any options, the output is directed to `stdout`. By providing the `-o` option, the Hardware Node ID will be stored in the following locations:

- In the specified file if you indicate the full path to this path (e.g., `C:\MyFile.txt`).
- In the specified file in the `C:\Documents and Settings\Administrator` folder if you indicate the file name only (e.g., `MyFile.txt`).

---

## vzcache

There may be situations when one and the same application or application update is installed not as a template, but separately inside several Containers. A good example of this is the Adobe Acrobat Reader application that can be installed inside a number of Containers thus having a vast amount of identical files throughout the Containers. This fact tells dramatically on the Container quotas, which may be avoided by putting all the identical files to the Hardware Node template area and creating links instead of real files inside the affected Containers.

The problem like the one described above is solved by using the `vzcache` utility. The utility scans the specified Containers for common files and caches these files in the Hardware Node template area (in the `X:\vz\templates\__vzcache` folder), replacing the real files inside the Containers with links to the template area (`X:\vz\templates\__vzcache`). In case of a significant number of identical files, using `vzcache` results in a notable disk space gain.

The `vzcache` utility has the following syntax:

```
vzcache [options] CT_list
```

The following command line options can be used with `vzcache`:

Option	Description
<code>-h, --help</code>	Print usage information.
<code>--version</code>	Display the utility version.
<code>-v, --verbose</code>	Verbose mode. Causes <code>vzcache</code> to print debugging messages about its progress. Multiple <code>-v</code> options increase verbosity. The maximal number of allowed options is 2.
<code>-q, --quiet</code>	Quiet mode. Causes all warning and diagnostic messages to be suppressed. Only fatal errors are displayed.
<code>-n, --cache-name NAME</code>	The name of the folder in <code>X:\vz\templates\__vzcache</code> where you want to store the cached files. If you omit this option, <code>vzcache</code> creates a subfolder with a random name.  If used with the <code>-a</code> option, tells <code>vzcache</code> to look for similar files in the specified cache folders.
<code>-a, --append NAME</code>	Look for similar files inside the specified Containers and compare them with the files in the indicated caches and templates. <code>-a</code> requires one of the following options: <ul style="list-style-type: none"> <li>▪ <code>-n</code>. In this case <code>vzcache</code> compares the files inside the specified Containers with those in the specified caches and replaces them with links, if necessary.</li> <li>▪ <code>-T</code>. In this case <code>vzcache</code> compares the files inside the specified Containers with those in the specified templates and replaces them with links, if necessary.</li> <li>▪ <code>-H</code>. In this case <code>vzcache</code> compares the files inside the specified Containers with those in the system partition on the Node and replaces them with links, if necessary.</li> <li>▪ <code>-A</code>. In this case <code>vzcache</code> compares the files inside</li> </ul>

	the specified Containers with those in all available caches and templates. It does not, however, look for similar files in the system partition on the Node.
<code>-s, --size-limit N</code>	Do not process files smaller than <i>N</i> bytes. By default, only empty files are not processed.
<code>-m, --min-links M</code>	Cache files found at least <i>M</i> times. The default value is 2.
<code>-T, --template-name NAME</code>	Look for similar files in the specified OS and application templates. Must be used with the <code>-a</code> option.  You can indicate several templates at a time and separate them by commas (e.g. <code>Template1,Template2</code> ).
<code>-H, --host</code>	Look for similar files in the system partition on the Hardware Node. Must be used with <code>-a</code> and one of the following options: <ul style="list-style-type: none"> <li>▪ <code>-A</code></li> <li>▪ <code>-T OS_Name</code> (where <i>OS_Name</i> denotes the name of the OS template installed on the Node)</li> </ul>
<code>-A, --all</code>	Look for similar files in all caches and templates and in the system partition on the Hardware Node. Must be used with the <code>-a</code> option.
<code>-f, --force</code>	Force the <code>vzcache</code> execution.
<code>-t, --test</code>	Do not perform any caching. Just collect and display the statistics on identical files.

Here are some examples on using the `vzcache` utility:

- To create a cache named `cache1` for Containers 153 and 154 caching only those files which are greater than 1024 bytes by using the `vzcache` utility, you should issue the following command:

```
vzcache -s 1024 -n cache1 153 154
```

- To append Container 155 to the existing `cache1` cache:

```
vzcache -a -n cache1 155
```

---

## vzlscache

The `vzlscache` utility displays a list of cache directories created by the `vzcache` utility (see above). It can be used to show either all the cache directories created on the given Hardware Node, or those used by the specified Container(s). The syntax of the command is the following:

```
vzlscache [options] [CT_ID ...]
```

The following options may be given:

Option	Description
<code>-h, --help</code>	Print the utility usage information.
<code>-q, --quiet</code>	Quiet mode. Causes all warning and diagnostic messages to be suppressed. Only fatal errors are displayed.

---

## vzuncache

Whereas the `vzcache` utility helps effectively gain disk space both in the Hardware Node and within Containers, there may be situations when it is necessary to detach a Container from its cache and copy the cached files back to the Container private area. A typical example of this is migrating a Container to another Hardware Node. The migration is not possible if there are links in the Container private area pointing to the `C:\vz\templates\__vzcache` folder on the Hardware Node.

The `vzuncache` utility is used to copy the regular files from the specified cache directory on the Hardware Node (located in the `C:\vz\templates\__vzcache` folder) back to the private area of the specified Container, replacing the corresponding links or stubs inside the Container with the real files and thus detaching the Container from its cache. `vzuncache` has the following syntax:

```
vzuncache [options] CT_ID [cache-directory-name ...]
```

The following command line options can be used with the `vzuncache` utility:

Option	Description
<code>-h, --help</code>	Print usage information.
<code>--version</code>	Display the utility version.
<code>-q, --quiet</code>	Quiet mode. Causes all warning and diagnostic messages to be suppressed. Only fatal errors are displayed.
<code>-v, --verbose</code>	Verbose mode. Causes <code>vzuncache</code> to print debugging messages about its progress. Multiple <code>-v</code> options increase verbosity.
<code>-a, --all</code>	Detach the specified Container from all the caches. The list of cache directories is not needed if this option is specified.
<code>-t, --test</code>	Do not copy files from the cache. Just collect and display the relevant statistics.
<code>-f, --force</code>	Detach the specified Container from its cache even in case the cache does not exist.

## vznetcfg

The `vznetcfg` utility is used to manage Virtual Networks and network classes on the Hardware Node. It has the following syntax:

```
vznetcfg command
vznetcfg --help
```

Where *command* can be one of the following:

Name	Description
<code>net new <i>NET_ID</i> <i>XX-XX-XX-XX-XX-XX</i> <i>XX[:VLAN_ID]</i></code>	Creates a new Virtual Network with the name of <i>NET_ID</i> and associates it with either: <ul style="list-style-type: none"> <li>a physical adapter on the Hardware Node having the MAC address of <i>XX-XX-XX-XX-XX-XX</i> or</li> <li>a VLAN (Virtual Local Area Network) adapter bound to the physical adapter with the MAC address of <i>XX-XX-XX-XX-XX-XX</i>.</li> </ul>
<code>net change <i>NET_ID</i> <i>XX-XX-XX-XX-XX-XX</i> <i>XX[:VLAN_ID]</i></code>	Associates the specified Virtual Network with the physical or VLAN adapter on the Hardware Node.
<code>net del <i>NET_ID</i></code>	Deletes the specified Virtual Network from the Hardware Node.
<code>net list</code>	Displays the Virtual Networks currently existing on the Hardware Node.
<code>class add <i>CLASS_ID</i> <i>IP_RANGE</i></code>	Creates a new network class with the ID of <i>CLASS_ID</i> and adds the IP addresses range specified as the value of <i>IP_RANGE</i> to the newly created class. An IP addresses range consists of a starting IP address and a subnet mask defining the number of IP addresses to be included in the network class. For example, specifying <code>10.0.0.0/255.0.0.0</code> or <code>10.0.0.0/8</code> as the value of <i>IP_RANGE</i> will have all the IP addresses in the range from <code>10.0.0.0</code> to <code>10.255.255.255</code> included in the given network class. You can indicate several IP addresses ranges for the network class and separate them by spaces.
<code>class remove <i>CLASS_ID</i> <i>IP_RANGE</i></code>	Removes the IP addresses range of <i>IP_RANGE</i> from the specified network class. You can specify several IP addresses ranges and separate them by spaces.  If <i>IP_RANGE</i> is omitted, the utility will remove the specified network with all IP addresses ranges included in it.
<code>class rate <i>CLASS_ID</i> <i>RATE</i></code>	Sets the maximal network bandwidth, in kilobits per second, any Container on the Hardware Node is guaranteed to receive for outgoing traffic within the specified network class.
<code>class list</code>	List the network classes currently existing on the Hardware Node.

---

## vznetstat

This utility outputs traffic usage statistics for Containers. It has the following syntax:

```
vznetstat [-v <CT_ID>] [-c <net_class_id>] [-a] [-r]
```

The utility accepts the following options:

- v <CT\_ID>     Display statistics for a Container with the ID of <CT\_ID>.
- c <class>     Show statistics for the <class> class only.
- a             Display statistics for all classes.
- r K|M|G       Display the network statistics, which is shown in bytes by default, in the following measurement units:
  - K: display the network statistics in kilobytes;
  - M: display the network statistics in megabytes;
  - G: display the network statistics in gigabytes.
- help        Display the utility usage information.

When executed without any options, the `vznetstat` utility displays the network statistics, in bytes, for all defined network classes of all running Containers on the Node. However, you can use any of the aforementioned options to customize the utility output.

## vzdevctl

This utility is used to forward the hardware devices (SCSI, iSCSI, USB flash drives, etc.) existing on the Hardware Node to Containers. It has the following syntax:

```
vzdevctl [--quiet|--verbose] add CT_ID --deviceid "name"
          [--alias name] [--exclusive] [--connect] [--onboot]
vzdevctl [--quiet|--verbose] remove CT_ID
          --deviceid "name" |--alias name
vzdevctl [--quiet|--verbose] connect CT_ID
          --deviceid "name" |--alias name
vzdevctl [--quiet|--verbose] dconnect CT_ID
          --deviceid "name" |--alias name
vzdevctl [--quiet|--verbose] set CT_ID --deviceid "name"
          [--alias name] [--onboot yes|no]
vzdevctl [--quiet|--verbose] devtree
          [--deviceid "name" |--all |--forwarded]
vzdevctl [--quiet|--verbose] status CT_ID
          [--deviceid "name" |--alias name |--all]
vzdevctl [--quiet|--verbose] linkadd CT_ID [--symlink "name"]
          --object "name"
vzdevctl [--quiet|--verbose] linkdel CT_ID
          --symlink "name" |--object "name"
vzdevctl [--quiet|--verbose] linkquery CT_ID
vzdevctl --version
vzdevctl --help
```

The commands description is given in the table below:

Name	Description
add	Forwards the hardware device to the specified Container.
remove	Removes the hardware device from the specified Container. If the device is in the active state, it is first disconnected from the Container.
connect	Connects the hardware device to the Container. Before connecting the device to the Container, you must first forward it to this Container.
dconnect	Disconnects the hardware device from the Container.
set	Sets device info. (Forwarded/shared state can be changed after disconnect-connect cycle only)
devtree	Shows all hardware devices available on the Hardware Node.
status	Shows the hardware device status for the specified Container.
linkadd	Adds a symbolic link to the Hardware Node object.
linkdel	Deletes a symbolic link from the specified Container.
linkquery	Displays the symbolic links for the specified Container.

The following options can be used with `vzdevctl`:

Name	Description
--deviceid	The ID to be assigned to the hardware device. It should be set in the following form: <i>Enumerator\Hardware_ID\Instance_ID</i> where:

- *Enumerator* denotes a system component responsible for discovering hardware devices on your Hardware Node (usually, the Plug-an-Play manager).
- *Hardware\_ID* is a vendor-defined identification string used by the Setup program to match a hardware device on your Hardware Node to an INF file.
- *Instance\_ID* is a hardware device identification string that distinguishes the device from other devices of the same type on your Hardware Node.

Detailed information on device identification strings is provided at <http://msdn.microsoft.com/en-US/library/ff541224.aspx>.

<code>--alias</code>	A short name (alias) to be assigned to the hardware device.
<code>--exclusive</code>	Sets the hardware device forwarding mode to 'exclusive' instead of 'shared' which is set by default. In this case the device cannot be forwarded to any other Container on the Hardware Node. This command must be used for any SCSI or USB-flash device which you are going to forward to a Container.
<code>--connect</code>	Specifies whether the hardware device being forwarded to the Container is also to be connected to the Container.
<code>--onboot</code> (yes no)	Sets whether the hardware device is to be automatically connected to the Container on its startup (yes is specified) or should be manually joined after the Container is up and running (no is specified).
<code>--force</code>	Forces the device disconnection for busy devices. This option is strongly unrecommended.
<code>--forwarded</code>	Displays only those hardware devices on the Hardware Node that are currently forwarded to some Containers.
<code>--all</code>	Can be indicated in one of the following cases: <ul style="list-style-type: none"> <li>▪ when used with the <code>devtree</code> command, displays all SCSI devices available on the Hardware Node (connected, not connected, denied);</li> <li>▪ when used with the <code>status</code> command, displays all Container devices including the ones that are currently not connected to the Container.</li> </ul>
<code>--symlink</code>	The full name to be assigned to the symbolic link.
<code>--object</code>	The full name of the object for which you are creating a symlink. If used without the <code>--symlink</code> option, the name specified after this option is also used to denote the Container symbolic link.
<code>--version</code>	Shows the Parallels Virtuozzo Containers version.
<code>--quiet</code>	Disables logging to the log file and screen. Can be used with any commands.
<code>--verbose</code>	Sets the logging level to the maximum value. Can be used with any commands.
<code>--help, /?</code>	Gets the utility usage information and exits.

---

## vzcpucfg

This utility is used to manage CPU pools on the Hardware Node. Using `vzcpucfg`, you can do the following:

- create a new CPU pool
- remove an existing CPU pool
- list the CPU pools currently existing on the physical server
- configure the number of CPUs in a CPU pool
- assign CPU pools to Containers

The utility has the following syntax:

```
vzcpucfg pool set POOL_ID CPU_RANGE
vzcpucfg pool del POOL_ID
vzcpucfg pool list
```

The following command line options can be used with the `vzcpucfg` utility:

Name	Description
<code>set</code>	Creates a new CPU pool or configures the number of CPUs in an existing CPU pool.
<code>del</code>	Removes a CPU pools from the physical server.
<code>list</code>	Lists the CPU pools existing on the physical server.
<i>POOL_ID</i>	The ID to be assigned to the CPU pool or the ID of the CPU pool to be removed from the physical server.
<i>CPU_RANGE</i>	The CPU range to be included in the CPU pool. A range may contain one or several CPUs (for example, 1 or 1-3). You can also specify several ranges at once and separate them by commas (for example, 0-2,3-6 , 8).

---

## vzquery

The `vzquery` utility is used to determine the Container ID using one of the following parameters:

- process ID
- session ID

The utility has the following syntax:

```
vzquery s2v <session_ID>
vzquery v2s <CT_ID>
vzquery p2v <process_ID>
```

The commands and options that can be used with `vzquery` are explained in the following table:

Name	Description
<code>s2v</code>	Displays the ID of the Container to which the Terminal Services session with the specified ID is opened.
<code>v2s</code>	Lists the IDs of all Terminal Services sessions opened to the specified Container.
<code>p2v</code>	Displays the ID of the Container where the process with the specified ID is running.
<code>session_ID</code>	The ID assigned to the Terminal Services session.
<code>CT_ID</code>	The ID assigned to the Container.
<code>process_ID</code>	The ID assigned to the process.

---

## vzwinupdatecmd

This utility is used to manage Windows Server updates inside Containers. It has the following syntax:

```
vzwinupdatecmd /listctupd <CT_ID1> <CT_ID2> ... [/all]
vzwinupdatecmd /rmupd <CT_ID1> <CT_ID2> ... [/all] /updates <KB1> <KB2> ...
vzwinupdatecmd /addupd <CT_ID1> <CT_ID2> ... [/all] /updates <KB1> <KB2> ...
vzwinupdatecmd /disableupdates <CT_ID1> <CT_ID2> ... [/all]
vzwinupdatecmd /enableupdates <CT_ID1> <CT_ID2> ... [/all]
```

The options that can be used with `vzwinupdatecmd` are explained in the following table:

Option Name	Description
<code>/listctupd</code>	List the Windows updates currently installed inside the Container.
<code>/addupd</code>	Installs a new Windows update inside the Container.
<code>/rmupd</code>	Removes an installed Windows update from the Container.
<code>/disableupdates</code>	Disables the automatic installation of Windows updates inside the Container.
<code>/enableupdates</code>	Enables the automatic installation of Windows updates inside the Container.
<code>/all</code>	Applies the corresponding operation to all Containers residing on the physical server.
<code>/updates</code>	Introduces the list of Windows updates to be installed inside or removed from the Container.
<code>CT_ID1</code>	Specifies the Container in respect to which the operation is to be performed.
<code>KB1</code>	The update to be installed inside or removed from the Container. The names of Windows updates should be specified in the same form they are used by Microsoft (e.g., KB893756 or KB896358).

---

# 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.

*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*.

*MAC address* stands for Media Access Control address, a hardware address that uniquely identifies each Node in a network. The MAC layer interfaces directly with the network media. Consequently, each different type of network media requires a different MAC layer.

*OS template* (or *Operating System template*) is used to create new *Containers* with a pre-installed operating system. See also *Template*.

*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.

*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* must have its own license installed.

*Parallels Infrastructure Manager* is an obsolete designation of *Parallels Virtual Automation*.

*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 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.

*Private area* is a part of the file system where *Container* files that are not shared with other *Containers* are stored.

*TCP (TCP/IP)* stands for Transmission Control Protocol/Internet Protocol. This suite of communications protocols is used to connect hosts on the Internet.

*Template* 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*.

*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 Power Panels* (or *VZPP*) is an obsolete designation of *Parallels Power Panel*.

*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

## A

About Parallels Virtuozzo Containers - 6  
About This Guide - 7  
Administrator  
    Container - 25  
Applications - 50

## B

Backing-Up Utilities - 41  
Backup  
    full - 42  
    incremental - 42

## C

Container  
    destroying - 17  
    listing - 28  
    migrating - 37  
    mounting - 17  
    private area - 17, 37  
    restoring - 25, 44  
    starting/stopping - 17

## D

Documentation Conventions - 8

## F

Feedback - 10

## G

General Utilities - 13  
Getting Help - 9  
Glossary - 60

## H

Host OS - 60  
Hostname  
    Hardware Node - 37

## I

Internet Explorer - 34, 60  
IP Address

Hardware Node - 42

## L

Licensing Utilities - 33

## M

Migration  
    Container to Container - 37  
    physical server to Container - 39  
Migration Utilities - 36  
Mozilla - 60

## N

Node  
    Destination - 37  
    Source - 37, 42

## O

Offline Management - 40  
Organization of This Guide - 7

## P

Parallels Agent - 42  
Parallels Power Panel - 60  
Parallels Virtual Automation - 60  
Parallels Virtuozzo Containers License - 60  
    checking - 35  
    installing - 34  
    overview - 60  
Parallels Virtuozzo Containers Utilities  
    Overview - 11  
Preface - 5

## R

Registry - 17  
Resources  
    disk space - 31

## S

Service Container - 60  
Supplementary Utilities - 49

## T

Template

- application - 46, 47, 48
- area - 50
- installing - 47
- listing - 46
- removing - 48

Template Management Utilities - 46

## U

### Utilities

- backup management - 42, 44, 45
- Container management - 17, 23, 24, 25, 28
- license management - 34, 35
- migration management - 37, 39, 40
- resources management - 31, 32
- supplementary - 49, 50, 51, 52, 53, 54
- template management - 46, 47, 48

## V

- vzabackup - 42
- vzarestore - 44
- vzcache - 50
- vzcpucfg - 57
- vzctl - 14
- vzctl addrole, vzctl delrole, vzctl enumrole - 27
- vzctl create - 16
- vzctl defrag - 26
- vzctl delete and vzctl destroy - 17
- vzctl exec, vzctl exec2, and vzctl enter - 23
- vzctl mount and vzctl umount - 17
- vzctl mounttext and vzctl umounttext - 24
- vzctl partadd and vzctl partdel - 24
- vzctl reinstall - 25
- vzctl set - 18
- vzctl shrink - 25
- vzctl start, vzctl stop, vzctl restart, and vzctl status - 17
- vzdevctl - 55
- vzkeygen - 49
- vzlicload - 34
- vzlicview - 35
- vzlist - 28
- vzlist Output Parameters and Their Modifiers - 29
- vzlscache - 51
- vzmigrate - 37
- vzmlocal - 40
- vznetcfg - 53
- vznetstat - 54
- vzp2v - 39
- vzpkgadd - 47
- vzpkgdeploy - 47
- vzpkgls - 46
- vzpkgrm - 48
- vzquery - 58

- vzquota - 31
- vzquota setlimit - 32
- vzquota stat and vzquota show - 32
- vzuncache - 52
- vzvssctl - 45
- vzwinupdatecmd - 59

## X

XML - 60