
Parallels

Parallels Virtuozzo Containers for Windows

Reference Guide

Version 4.0



Copyright © 1999-2008 Parallels Software International Inc.

ISBN: N/A
Parallels Holdings Ltd.
13755 Sunrise Valley Drive
Suite 600
Herndon, VA 20171
USA
Tel: +1 (703) 815 5670
Fax: +1 (703) 815 5675

© 1999-2008 Parallels Holdings Ltd. All rights reserved.
Distribution of this work or derivative of this work in any form is prohibited unless prior written permission is obtained from the copyright holder.

Contents

Preface	5
About Parallels Virtuozzo Containers	6
About This Guide	7
Organization of This Guide	7
Documentation Conventions	8
Getting Help	9
Feedback	10
Configuring Parallels Virtuozzo Containers	11
Configuring Containers.....	11
Adding Custom Action Scripts	14
Parallels Virtuozzo Command Line Interface	16
vzctl	17
vzctl create.....	19
vzctl delete and vzctl destroy.....	20
vzctl mount and vzctl umount	21
vzctl start, vzctl stop, vzctl restart, and vzctl status	21
vzctl set.....	22
vzctl exec, vzctl exec2, and vzctl enter.....	27
vzctl mounttext and vzctl umounttext.....	28
vzctl partadd and vzctl partdel.....	29
vzctl reinstall	30
vzctl shrink	31
vzlist.....	31
vzlist Output Parameters and Their Specifiers	32
vzquota.....	33
vzquota setlimit	34
vzquota stat and vzquota show	34
Licensing Utilities.....	35
vzlicload	35
vzlicview	35
Migration Utilities	36
vzmigrate	37
vzp2v	38
vzmlocal	39
Backing-Up Utilities	40
vzabackup.....	41
vzarestore.....	43
vzvssctl	44
Template Management Utilities.....	45
vzpkgls.....	45
vzpkgdeploy	45
vzpkgadd	46
vzpkgrm.....	46
Supplementary Tools	47
vzsveinstall	47
vzkeygen.....	47

vzcache48
vzlscache49
vzuncache49
vznetcfg50
vznetstat.....51

Glossary 52

Index 54

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. Virtuozzo Containers 4.0 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 Virtuozzo 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 Virtuozzo-based system structure:

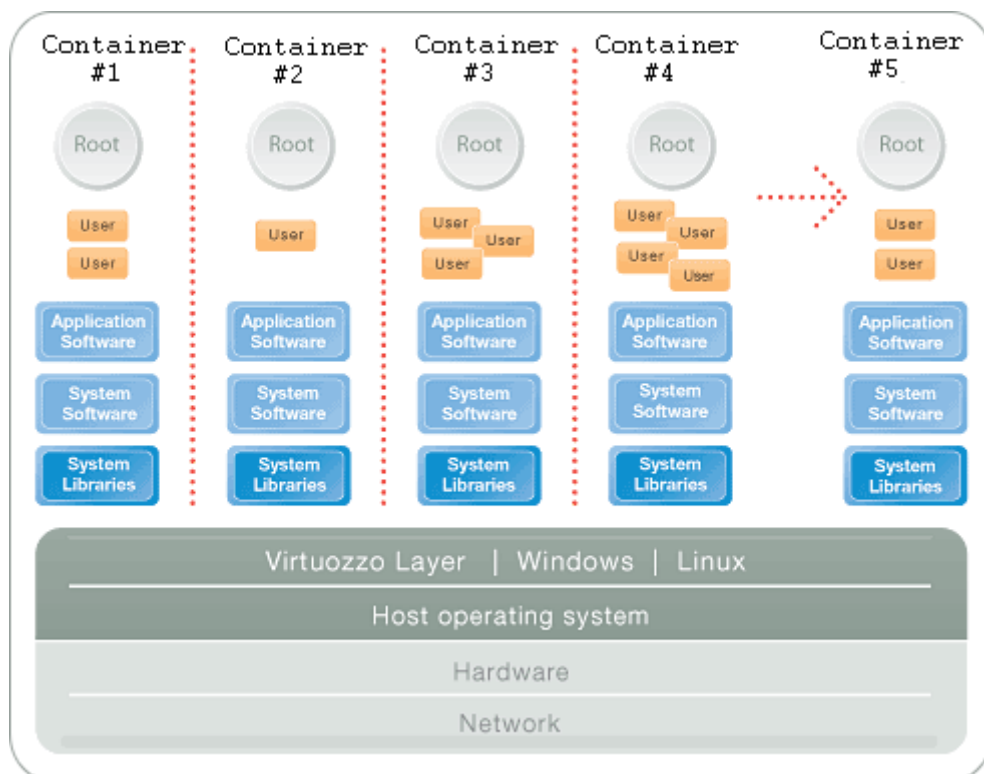


Figure 1: Virtuozzo Containers OS Virtualization

The Parallels Virtuozzo 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 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 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 Virtuozzo Containers 4.0 to meet your requirements and to perform various tasks by using the corresponding Virtuozzo 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 Virtuozzo 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, *Configuring Parallels Virtuozzo Containers*, explains how to configure Parallels Virtuozzo configuration files and to use custom action scripts to make your Containers function more effectively.

Chapter 3, *Parallels Virtuozzo Command Line Interface*, describes all available Hardware Node command-line utilities together with the options and switches that can be passed to them.

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 shipped with Virtuozzo Containers 4.0 which can help you use the product more effectively. These resources include:

- **Manuals:**
 - **Parallels Virtuozzo Containers Evaluation Guide.** This guide is destined to introduce you to the main features of Virtuozzo Containers 4.0 and to its underlying technology, to help you set up an environment for evaluating the Parallels Virtuozzo major features, and to suggest the relevant procedures for this evaluation.
 - **Getting Started With Parallels Virtuozzo Containers for Windows.** This guide provides basic information on how to install Parallels Virtuozzo Containers 4.0 on your server, create new Containers, and perform main operations on them. As distinct from the given guide, it does not contain detailed description of all the operations needed to install and set Parallels Virtuozzo to work (e.g. planning the structure of your Virtuozzo network or performing the Virtuozzo Containers unattended installation).
 - **Parallels Virtuozzo Containers for Windows Installation Guide.** This guide provides exhaustive information on the process of installing, configuring, and deploying your Virtuozzo system. As distinct from the given guide, it contains a more detailed description of all the operations needed to install and set Virtuozzo Containers 4.0 to work including planning the structure of your Virtuozzo network, performing the Virtuozzo Containers unattended installation, etc. Besides, it does not include the description of any Container-related operations.
 - **Parallels Virtuozzo Containers for Windows User's Guide.** This guide provides comprehensive information on Virtuozzo Containers 4.0 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 Virtuozzo system.
 - **Parallels Virtuozzo for Windows Templates Management Guide.** This guide is meant to provide complete information on Virtuozzo templates - an exclusive Parallels Virtuozzo 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.).
- **Help systems:**
 - **Parallels Management Console Help.** This help system provides detailed information on Parallels Management Console - a graphical user interface tool for managing Virtuozzo Hardware Nodes and their Containers.
 - **Parallels Infrastructure Manager Online Help.** This help system shows you how to work with Parallels Infrastructure Manager - a tool providing you with the ability to manage Virtuozzo 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 thru a common Web browser on any platform.

Feedback

If you spot a typo in this guide, or if you have thought of a way to make this guide better, we would love to hear from you!

The Parallels documentation forum is the ideal place for your comments and suggestions. It is regularly monitored by the members of the Parallels technical documentation department, so it is likely that you will receive a reply to your post before long.

Note that new users will be asked to fill in a short registration form before being able to post. Registering will allow you to participate not only in the documentation forum discussions, but in all the other Parallels forums as well.

CHAPTER 2

Configuring Parallels Virtuozzo Containers

This chapter provides information on the Parallels Virtuozzo configuration parameters. Besides, it explains how you can use your own custom action scripts to configure your Containers.

In This Chapter

Configuring Containers	11
Adding Custom Action Scripts	14

Configuring Containers

Each Container has its own configuration file which is stored in the `X:\vz\Conf\` folder and has a name of `CT_ID.conf`. The parameters in the `X:\vz\Conf\<<CT_ID>.conf` file are presented in the form of the `PARAMETER="value"` pattern. The main parameters defining the Container configuration can be subdivided into the following categories: miscellaneous, resource management, and networking parameters.

Note: During the Virtuozzo Containers installation, the `Conf` subfolder is automatically created in the folder you specified for storing all Container configuration files. By default, the `C:\vz\` folder is used. However, you might have defined another path for this folder.

Miscellaneous parameters:

Parameter	Description	Default value
VzPrivate	The path to the Container private area. You specify the location of the Container private area by passing the <code>--private path</code> argument to the <code>vzctl create</code> command. This parameter is omitted if the Container private area is located in the <code>C:\vz\private\</code> directory.	<code>C:\vz\private\</code>
GUID	The pseudo-random number used to uniquely identify the Container. GUIDs (Globally Unique Identifiers) are written using a four-byte word, 3 two-byte words, and a six-byte word, such as: { 3F2504E0-4F89-11D3-9A0C-0305E82C3301 }	-

Autoboot	Specifies whether the Container should be started automatically on system startup. Virtuozzo automatically starts all Containers that have this value set to "1".	1
OfflineMgmt	Specifies whether the Container can be managed by the Container administrator using the offline services indicated in the <code>OfflineServices</code> parameter in this file provided the <code>OfflineMgmt</code> parameter in the <code>C:\vz\Conf\0.conf</code> file is set to "1".	1
OfflineServices	Overrides the <code>OfflineServices</code> parameter from the <code>C:\vz\Conf\0.conf</code> file. The services enumerated in this parameter can be used by the Container administrator for managing Containers, if the <code>OfflineMgmt</code> parameter in the <code>C:\vz\Conf\0.conf</code> file is set to "1".	-
TEMPLATE	This is the name of the OS template the Container is based upon.	w2k3
TEMPLATES	Provides a list of application templates. You are not recommended to modify this parameter since it is used to track the history of the added templates. This parameter is omitted if no templates have been added to the Container.	-
WindowsDir	The path to the folder where the Windows Server 2003 system files for the given Container are located. In the current version of Virtuozzo, you cannot change the default path set during the Container creation.	C:\WINDOWS
DISKTYPE	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"> ▪ <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. ▪ <code>sparse</code>: in this case the size of a virtual hard disk is also fixed and disk space is allocated during the disk creation. However, an <code>.efd</code> file corresponding to a virtual hard disk can be regarded as a standard NTFS file supporting the 'sparse file' functionality. When the 'sparse file' functionality is enabled, the Host OS does not allocate hard drive space to an <code>.efd</code> file except in regions where it contains non-zero data. This allows you to make the disk space usage inside your Containers more efficient. ▪ <code>compact</code>: 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 as large as the maximum size specified during the hard disk creation. 	plain

Resource management parameters control the amount of resources a Container can consume. All resources are described in the **Managing Resources** chapter of the **Parallels Virtuozzo Containers User's Guide** in detail; here is only a list of parameters that can be found in the `C:\vz\Conf\<<CT_ID>.conf` file:

Parameter	Description
DiskQuota	The total size of disk space that can be consumed by the Container, in 1-Kb blocks.
CpuUnits	The 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.
ProcessCountLimit	The maximal number of processes the Container may create. Upon hitting this limit, the Container will not be able to start a new process or thread.
SessionCountLimit	The number of terminal sessions. This parameter is used to limit the number of simultaneous terminal sessions, i.e. the number of users that are allowed to connect to the Container at the same time thru MS TSC, RDP, etc.
CommittedMemoryLimit	The size of private (or potentially private) memory that can be allocated by the Container, in bytes. The memory that is always shared among different applications is not included in this resource parameter.
NetAllowBroadcasts	Setting this parameter to "1" enables the network browsing for the Container. The network browsing, if enabled, allows the Container administrator to view all the network servers and file sharing resources available on the Microsoft Windows network. By default, the network browsing is disabled. You can enable this feature for all Containers on your Node at once by setting this parameter to "1" in the <code>C:\vz\Conf\0.conf</code> file.
NetQoSPacketScheduler	Setting this parameter to "1" enables the Windows QoS packet scheduler for the Container. The QoS packet scheduler is used to regulate how much traffic an application inside your Container is allowed, thus enforcing the QoS parameters originally set for a particular application. By default, the QoS packet scheduler is disabled.
<p>Note: In case you have enabled the QoS packet scheduler for a running Container, you should restart this Container for the changes to take effect.</p>	
CpuLimit	The positive number indicating the CPU time, in percent, the corresponding 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.

Network-related parameters allow you to set a hostname, an IP address, and a network mask for the given Container:

Parameter	Description
Hostname	Specifies the hostname of the Container.
IP	Sets an IP address for the Container. Several IP addresses per Container are allowed; in this case they should be divided by the semicolon and a space.
IPMASK	The network mask of the Container. It should be the same for all the Containers located on a specific physical network segment.

VeMac The MAC address to be assigned to the Container. A new MAC address should be specified in the form of "XX-XX-XX-XX-XX-XX" (e.g. VeMac="00-AA-BB-CC-DD-EE").

Note: You can change MAC addresses for stopped Containers only.

Adding Custom Action Scripts

ContainerVirtuozzo allows you to write and execute your own scripts on the Hardware Node to do additional actions when certain Container-related operations are performed. There are two kinds of scripts you can specify for execution on the Node or in the Container context:

- **Preoperation scripts.** These scripts are executed before some action is performed. They should be specified in the form of the `vz-pre<action_name>.<ext>` pattern, where `<action_name>` denotes the name of the operation to be performed (e.g. `start`), and `<ext>` is the extension of the file containing the script. Currently, the files with the `.exe`, `.cmd`, `.bat`, `.vbs` extensions are supported.
- **Postoperation scripts.** These scripts are executed after some action has been successfully completed. They should be specified in the form of the `vz-post<action_name>.<ext>` pattern where `<action_name>` denotes the name of the operation to be performed (e.g. `stop`), and `<ext>` is the extension of the file containing the script. Currently, the files with the `.exe`, `.cmd`, `.bat`, `.vbs` extensions are supported.

The names of all possible scripts that may be specified for executing before or after performing an operation are listed in the table below:

Script Name	Description
<code>vz-precreate.<ext></code>	This script is to be run before the Container creation.
<code>vz-postcreate.<ext></code>	This script is to be run after the Container creation.
<code>vz-premount.<ext></code>	This script is to be run before mounting the Container private area.
<code>vz-postmount.<ext></code>	This script is to be run after mounting the Container private area.
<code>vz-preunmount.<ext></code>	This script is to be run before unmounting the Container private area.
<code>vz-postunmount.<ext></code>	This script is to be run after unmounting the Container private area.
<code>vz-prestart.<ext></code>	This script is to be run before the Container start.
<code>vz-poststart.<ext></code>	This script is to be run after the Container start.
<code>vz-prestop.<ext></code>	This script is to be run before stopping the Container.
<code>vz-poststop.<ext></code>	This script is to be run after stopping the Container.
<code>vz-preclone.<ext></code>	This script is to be run before copying the Container.
<code>vz-postclone.<ext></code>	This script is to be run after copying the Container.
<code>vz-prechangevpsid.<ext></code>	This script is to be run before changing the Container ID.

`vz-postchangevid.<ext>` This script is to be run after changing the Container ID.

Note: All the scripts are executed in the Host OS context.

You may specify scripts that will be applied to all Containers residing on the Hardware Node. To this effect, you should place your script to the `C:\vz\Scripts\`. In its turn, locating a script to the `C:\vz\private\CT_ID\scripts\` directory allows you to set this script for executing in the given Container only (per-Container script). In the latter case, you may need to create the `C:\vz\private\CT_ID\scripts\` directory manually since it is not created by creating a Container. When executed, any script takes only one parameter - the ID of the corresponding Container.

Note: During the Virtuozzo Containers installation, the `Scripts` subdirectory is automatically created in the directory you specified for storing all Container configuration files. By default, the `C:\vz\` directory is used. However, you might have defined another path for the directory.

Let us assume you have created Container 101 with the private area located in the `C:\vz\Private\101\` directory. And now you wish to write a script that will be executed each time after the Container start and display the number of events currently recorded in the Container System event log. To this effect, you should open Notepad or another text editor and write the following script code:

```
strComputer = "."
Set objWMIService = GetObject("winmgmts:" _
    & "{impersonationLevel=impersonate}!\\" & strComputer & "\root\cimv2")
Set colLogFiles = objWMIService.ExecQuery _
    ("Select * from Win32_NTEventLogFile where LogFileName='System'")
For Each objLogFile in colLogFiles
    Wscript.Echo objLogFile.NumberOfRecords
Next
```

After that, you should save the file with the `vz-poststart.vbs` name, create the `scripts` subdirectory in the `C:\vz\Private\101\` directory, and place your script to this subdirectory. Make sure that the full path of the script is `C:\vz\Private\101\scripts\vz-poststart.vbs`. From this point on, the `vz-poststart.vbs` script will be executed every time you start Container 101.

Note: By default, Virtuozzo is shipped with the `vz-poststart.cmd` script located in the `C:\vz\Scripts\` directory. This script is needed for all Containers residing on the Hardware Node to perform certain operations necessary for Container error-free functioning. However, specifying your own per-Container scripts (i.e. scripts in the `C:\vz\private\CT_ID\scripts\` directory) prevents the `vz-poststart.cmd` script from running. Therefore, you should make sure that each of your per-Container scripts calls the `C:\vz\Scripts\vz-poststart.cmd` script.

CHAPTER 3

Parallels Virtuozzo Command Line Interface

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

All the Parallels Virtuozzo utilities are supposed to be run on the Hardware Node by using the `cmd.exe` command interpreter, which can be opened by selecting **Start --> Run** and executing the `cmd` command. There are a number of regulations which should be observed while working in the command prompt window. One of such rules would be to enclose in quote marks any parameter consisting of two or more words (e.g. "C:\Program Files"). Please consult the **Microsoft Command-line reference** (<http://www.microsoft.com/resources/documentation/windows/xp/all/proddocs/en-us/ntcmds.msp>) to learn the rules of working in the command prompt window.

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

Licensing utilities allow loading new license files and viewing the state of the license:

<code>vzlicview</code>	Utility to show the Virtuozzo license status and parameters.
<code>vzlicload</code>	Virtuozzo license loader.

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 the local cloning or moving of the Containers.
<code>vzp2v</code>	Utility for migrating physical servers to Containers on the Hardware Node.

Container backup utilities allow you to back up and restore the Container private areas, configuration files, and quota information:

<code>vzabackup</code>	Utility to back up Containers and to store them either on a local Node or on a special Backup Node.
<code>vzarestore</code>	Utility to restore Containers backed up by using the <code>vzabackup</code> utility.
<code>vzvssctl</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 get a list of templates available on the Hardware Node and in Containers.
<code>vzpkgdeploy</code>	Utility to install/remove OS and application templates on/from the Hardware Node.
<code>vzpkgadd</code>	Utility to add a new application template to a Container.
<code>vzpkgrm</code>	Utility to remove a template from a Container.

Supplementary utilities perform a number of tasks and are used by other Virtuozzo utilities:

<code>vzsveinstall</code>	Utility to create the Service Container on the Hardware Node.
<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 directories 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.

In This Chapter

<code>vzctl</code>	17
<code>vzlist</code>	31
<code>vzquota</code>	33
Licensing Utilities.....	35
Migration Utilities.....	36
Backing-Up Utilities.....	40
Template Management Utilities.....	45
Supplementary Tools.....	47

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

Where `command` can be one of the following:

<code>create</code>	Used to create a Container.
<code>delete</code>	Used to remove a Container.
<code>destroy</code>	Like <code>vzctl delete</code> , this command is also used to remove a Container from the Hardware Node.
<code>mount</code>	Used to mount the Container private area.
<code>umount</code>	Used to unmount 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>	Used to set Container parameters, including resource control settings, the Container private area location, its hostname, IP addresses, etc.
<code>enter</code>	Provides a way for Hardware Node administrators to “enter” a Container without knowing the Container root password. Use this command with caution and never run it on untrusted Containers.
<code>exec</code> , <code>exec2</code>	These two commands allow you to run arbitrary commands inside a Container without logging in to the corresponding Container. The difference between the command is their returned status.
<code>mountext</code>	Used to mount folders and drives of the Hardware Node to a Container.
<code>umountext</code>	Used to unmount external folders and drives of the Hardware Node from a Container.
<code>partadd</code>	Used to create a new Container drive by mounting it to a loopback file on the Hardware Node.
<code>partdel</code>	Used to unmount the Container drive mounted by means of the <code>vzctl partadd</code> command from a file on the Hardware Node.
<code>interface</code>	Used to manage network adapters on the Hardware Node and inside Containers.
<code>reinstall</code>	Use to restore the original state of the Container system and application files if the Container was broken for some reason or other.
<code>shrink</code>	Used to shrink Container virtual disk drives of the <code>compact</code> type.

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

<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>vzctl</code> session.

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

<code>--version</code>	Displays the Virtuozzo version currently installed on the Hardware Node.
<code>--help</code>	Displays the usage information on <code>vzctl</code> .

vzctl create

This command is used to create a new Container on your Hardware Node. It has the following syntax:

```
vzctl create CT_ID --pkgset name [--ipadd ip_address]
                    [--hostname hostname] [--config filename]
                    [--private path] [--disktype plain|sparse|compact]
```

With this command, you can create regular Containers. A Container ID is required for this command and shall be unique within a Hardware Node.

Note: Container IDs from 1 to 100 are reserved for internal Virtuozzo needs. Do not use IDs from 1 to 100 for your Containers.

Command arguments are as follows:

Name	Description
<code>--pkgset name</code>	Mandatory. Denotes the OS template to be used for the Container creation.
<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 be assigned to the Container.
<code>--config filename</code>	Optional. If this option is given, <code>vzctl</code> copies the values from the sample Container configuration file located in the <code>X:\Program Files\SWSOft\Virtuozzo\Configs</code> folder on the Hardware Node and having the name in the form of <code>ve-<name>.conf-sample</code> . The sample configuration files usually have a number of resource control limits for the Container and some application templates to be added to the Container immediately upon its creation. If you skip this option, you will have to set resource control parameters for the Container by using the <code>vzctl set</code> command before you are able to start the Container. The application templates will have to be also added manually.
<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. For example, you may specify the <code>--private F:\MyPrivateArea</code> argument to locate your Container private area in the <code>MyPrivateArea</code> folder on the <code>F:\</code> volume in case the <code>F:\</code> volume exists on the Hardware Node. While defining a path for this folder, you should take care of the following: <ul style="list-style-type: none"> ▪ This folder cannot be a mount point, i.e. you cannot mount external disk partitions to this folder. ▪ This folder cannot be a network share, i.e. it cannot be located on a server network drive. ▪ The hard disk partition where this folder will be located should have no less than 10 Gb of free disk space.

`--disktype`
`plain|sparse|compact`

Optional. The type of the Container virtual disk drive. It can be one of the following:

- `plain`: in this case the size of a virtual hard disk is fixed, i.e. all disk space is allocated during the virtual disk creation.
- `sparse`: in this case the size of a virtual hard disk is also fixed and disk space is allocated during the disk creation. However, an `.efd` file corresponding to a virtual hard disk can be regarded as a standard NTFS file supporting the 'sparse file' functionality. When the 'sparse file' functionality is enabled, the Host OS does not allocate hard drive space to an `.efd` file except in regions where it contains non-zero data. This allows you to make the disk space usage inside your Containers more efficient.
- `compact`: 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 as large as the maximum size specified during the hard disk creation.

When creating a new Container, you should specify a unique ID for it. There are no restrictions besides uniqueness from the `vzctl` standpoint. However, it is advisable to assign different ID ranges to Hardware Nodes in multi-node environments. For example, you can use IDs from 101 to 2000 on the first node, IDs from 2001 to 4000 on the second one and so on. This will help you in tracking down the node where Container was created and will eliminate possibility of Container IDs conflicts when migrating Containers between Nodes.

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 dismounted.

Note: The `vzctl delete` command introduced in Virtuozzo 4.0 for the first time is meant to replace the `vzctl destroy` command in future. However, `vzctl destroy` is fully supported in the current version of Virtuozzo Containers; so, it depends entirely on you which command to use for removing Containers from your Hardware Node.

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 (for example, `C:\vz\root\<CT_ID>` on the Hardware Node) 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 as 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, for `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. It will set up all network interfaces, initialize the Container quota, if needed, etc.

`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 the corresponding Container.

The `vzctl status` command shows the current Container state. It outputs the following information: what IP address is assigned to the Container and whether the Container is running as in the example below:

```
C:\Documents and Settings\Administrator> vzctl status 101
CtId 101 exist unmounted down
```

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. Practically all Container settings can be changed dynamically without the necessity of the Container reboot (except for the `--onboot` and `--psched` options).

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”, Virtuozzo will automatically start this Container on next system startup.
<code>--userpasswd user:password</code>	This setting creates a new user with the specified password inside the Container, or changes the password of an already existing user. Omitting the user name (e.g. <code>:2wse3rd</code>) will change the password for the built-in Administrator account inside the Container, even though it was renamed for some reason or other.
<code>--offline_management yes no</code>	This setting enables/disables the direct managing of the Container thru a common Internet browser by means of Parallels Power Panel. To enable/disable the offline management feature for the Hardware Node and, therefore, for all Containers residing on it, you should set this parameter for the Container with ID 0.
<code>--devadd device</code>	This setting allows you to grant access to any drive on the Hardware Node from inside a Container. Drive names must be specified in the <code>X:</code> form (e.g. <code>C:</code> , <code>D:</code>).
<code>--devdel device</code>	This setting shuts down access to a drive on the Hardware Node granted by means of the <code>--devadd</code> setting. Drive name must be specified in the <code>X:</code> form (e.g. <code>C:</code> , <code>D:</code>).
<code>--showveid yes no</code>	If set to <code>yes</code> , the Container 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. If set to <code>no</code> , the string is not shown on the Container desktop. By default, the Container ID string is displayed on the Container desktop. Please keep in mind that you should log off from the Container and log in to it anew for the changes to take effect.
<code>--showhostname yes no</code>	If set to <code>yes</code> , the Container 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. If set to <code>no</code> , the string is not shown on the Container desktop. By default, the hostname string is hidden from the Container desktop. Please keep in mind that you should log off from the Container and log in to it anew for the changes to take effect.

<code>--name</code>	An arbitrary name assigned to the Container. This name can be used, along with the Container ID, to refer to the Container while performing certain Container-related operations on the Hardware Node.
<code>--description</code>	This option allows you to set the description for the Container.
<hr/>	
	Note: You are allowed to use only symbols in the 'A - z' and '0-9' ranges in your descriptions.
<hr/>	
<code>--tsmode admin app</code>	Specifies the Terminal Services mode for the Container: <ul style="list-style-type: none"> ▪ if set to <code>admin</code>, enables the Remote Desktop for Administration mode for the Container; ▪ if set to <code>app</code>, enables the Terminal Server mode for the Container. By default, any Container is set to work in the Terminal Server mode during its creation.
<code>--tslicservers name</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>--scsi yes no</code>	Enables/disables SCSI-aware drivers to load inside the Container. If you wish to enable/disable the loading of SCSI-aware drivers for all Containers on the Hardware Node at once, specify 0 as <code><CT_ID></code> .
<code>--failover_cluster yes no</code>	Enables/disables the support of the Microsoft Cluster Server (MSCS) software inside the specified Container. If you wish to enable/disable the possibility of using the MSCS clustering software for all Containers on the Hardware Node at once, specify 0 as <code><CT_ID></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>	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 percent</code>	This is a positive integer number that determines the minimal guaranteed share of the CPU time, in per cent, the given Container is guaranteed to receive.
<code>--cpulimit percent</code>	This is a positive number indicating the CPU time, in percent, the corresponding 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>--diskspace <i>number</i></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 <i>drive_name</i></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"> ▪ <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. ▪ <code>compact</code>: 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 as much as the maximum size specified during the hard disk creation.
<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 startup.
<code>--rate <i>class:Kbits</i></code>	If traffic shaping is turned on, then 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. that a Container can use:

<code>--hostname <i>name</i></code>	Sets the hostname of the Container.
<code>--ipadd <i>addr</i></code>	Adds an IP address to a 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. If used with the <code>--netif</code> option, adds an IP address to the Container virtual network adapter specified.
<code>--ipadd <i>addr/netmaskbits</i></code>	Along with adding an IP address, it sets a subnet mask that determines to what subnet the given IP address will belong.
<code>--ipadd <i>addr:netmask</i></code>	If used with the <code>--netif</code> option, sets a subnet mask for the specified Container virtual network.

<code>--ipdel <i>addr</i> all</code>	<p>Allows you to revoke an IP address from the Container. Specifying <code>all</code> after the option, removes all IP addresses assigned to the Container.</p> <p>If used with the <code>--netif</code> option, deletes an IP address from the specified Container virtual network adapter.</p>
<code>--nameserver <i>addr</i></code>	<p>The DNS server IP address for the Container. More than one server may be specified in the space-separated format.</p> <p>If used with the <code>--netif</code> option, sets the DNS server for the specified Container virtual network adapter.</p>
<code>--searchdomain <i>domain</i></code>	<p>The DNS search domain for the Container. You may specify several domains and separate them by spaces.</p>
<code>--psched on off</code>	<p>Enabling/disabling the Quality of Service packet scheduler for the Container; by default, it is disabled. The QoS packet scheduler is used to regulate how much traffic an application inside your Container is allowed, thus enforcing the QoS parameters originally set for a particular application. After enabling the Quality of Service packet scheduler, any Container can be managed as a normal stand-alone Windows 2003 server in respect of setting and working with all Windows Quality of Service components.</p> <hr/> <p>Note: In case you have enabled the QoS packet scheduler for a running Container, you should restart this 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">▪ Use the <code>on</code> command switch to allow the Container to communicate only with the servers from the same subnet(s).▪ Use the <code>off</code> command switch to enable the Container to access/be accessed by the servers located both on the same subnet and on other ones.

`--preferred_adapter`
Adapter_ID

In case there are several network adapter cards 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, therefore, to handle the whole network traffic of the Container.

In Virtuozzo-based systems, network adapters are identified by the Media Access Control (MAC) addresses assigned to them. So, if you wish to set a particular adapter to be used for this or that Container on the Node, you should specify its MAC address as the value of this option. For example, the following command

```
# vzctl set 101 --preferred_adapter 00-0A-CR-32-F1-FD
```

sets the network adapter card with the MAC address of 00-0A-CR-32-F1-FD to be used for handling all the network traffic for Container 101.

While working with network adapters in Virtuozzo-based systems, please keep in mind the following:

- After the Virtuozzo Containers installation, all active adapters on the Hardware Node are used to control network traffic for all Containers residing on the Node.
- You can specify several network adapters for any Container on the Node and separate them by spaces. In this case the corresponding Container will be able to access all networks to which the specified adapters are connected.
- If you use the option without specifying any value (i.e. execute the `vzctl set CT_ID --preferred_adapter ""` 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.

`--netif_add` *name*

Creates a new virtual network adapter for the Container with the name specified. You can create up to 16 network adapters for each Container on the Hardware Node.

`--netif_del` *name/GUID*

Removes the virtual network adapter with the specified name or GUID (globally unique identifier) from the Container.

`--nettype` *bridged|routed*

Sets the operating mode (either bridged or host-routed) for the default Container virtual network adapter. If used with the `--netif` option, sets the operating mode for the specified Container adapter.

`--netif` *name*

Specifies the name of the Container virtual network adapter whose settings are to be configured. This option can be used along with one of the following options: `--ipadd`, `--ipdel`, `--nameserver`, `--winserver`, `--mac`, `--gateway`, `--vpn`, `--nat`, `--cluster_ip`, `--dhcp`.

<code>--mac XX-XX-XX-XX-XX-XX</code>	The MAC address to be assigned to the default Container virtual network adapter. If used with the <code>--netif</code> option, sets the MAC address for the specified Container virtual network adapter.
<code>--gateway <i>addr</i></code>	Specifies the IP address of the device to be used for routing the traffic from the default Container virtual network adapter to external networks. If used with the <code>--netif</code> option, sets the gateway for the specified Container virtual network adapter.
<code>--winsserver <i>addr</i></code>	The IP address of the WINS server to be used by the default Container virtual network adapter. If used with the <code>--netif</code> option, sets the WINS server for the specified Container virtual network adapter.
<code>--vpn on off</code>	Enables/disables the Virtual Private Network (VPN) support by the default Container virtual network adapter. By default, the VPN support is disabled. If used with the <code>--netif</code> option, enables/disables the VPN support for the specified Container virtual network adapter.
<code>--nat on off</code>	Enables/disables the Network Address Translation (NAT) functionality for the default Container virtual network adapter. If used with the <code>--netif</code> option, enables/disables the VPN support for the specified Container virtual network adapter. By default, NAT is disabled for any newly created Container network adapter.
<code>--cluster_ip <i>IP_addr</i></code>	Adds the virtual (Network Load Balancing cluster) IP address to the default Container virtual network adapter. If used with the <code>--netif</code> option, adds the NLB cluster IP address for the specified Container virtual network adapter.

vzctl exec, vzctl exec2, and vzctl enter

These commands are used to run arbitrary commands inside a Container being authenticated as the administrator on the Hardware Node. 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 mounttext and vzctl umounttext

You may wish to mount a drive on the Hardware Node to a drive or folder inside your Container. To this effect, you can use the `vzctl mounttext` command. It has the following syntax:

```
vzctl mounttext CT_ID --srcdir path --dstdir path
```

The command should be used with the following arguments:

Name	Description
<code>CT_ID</code>	The ID of the Container where the external drive from the Hardware Node is to be mounted.
<code>--srcdir path</code>	The drive inside your Container where the external drive from the Node will be mounted. You may use any existing folder or drive or create a new folder or drive (e.g. using the <code>vzctl partadd</code> command) to mount the desired drive.
<code>--dstdir path</code>	The full path to the drive on the Node that is to be mounted inside your Container.

Before mounting drives from the Hardware Node to any of your Containers, you should first make them accessible by executing the `vzctl set --devadd` command (see page 22). The session below demonstrates how you can make your CD-ROM drive on the Hardware Node available inside Container 101:

```
vzctl set 101 --devadd D:
Command output
vzctl mounttext 101 --srcdir C:\CDROM --dstdir D:
Command output
```

In the example above, we granted Container 101 access to the D drive on the Hardware Node (assuming this is used as the Node CD-ROM drive) and mounted this drive to the `C:\CDROM` folder inside Container 101.

You might as well mount the Node CD-ROM drive to a separate drive letter inside the Container:

```
vzctl set 101 --devadd D:
Command output
vzctl mounttext 101 --srcdir E: --dstdir D:
Command output
```

Note: The permissions of the drive mounted to a Container correspond to those of this drive on the Hardware Node, i.e. you are able to perform the same operations on the drive inside the Container as you are allowed on the Node.

At any time, you can use the `vzctl umounttext` command to unmount an external drive from a Container. So, if you used the first example for mounting the CD-ROM drive, you can issue the following command to unmount it from Container 101:

```
vzctl umounttext 101 --srcdir C:\CDROM
```

As can be seen from our example, you should pass the `CT_ID` and `--srcdir path` arguments to the command to identify the Container where the mounted drive exists and the drive itself that will be unmounted from this Container.

vzctl partadd and vzctl partdel

The `vzctl partadd` command is used to create a new drive inside your Container by mounting it to a loopback file on the Hardware Node. After the command execution, the mounted drive becomes visible

- On the Hardware Node:
 - As a subfolder in the `X:\vz\root\CT_ID\` folder and
 - As a file in the `.efd` format in the `X:\vz\private\CT_ID` folder.
- From inside the Container: log in to the Container via RDP and launch Windows Explorer. Your new drive will be displayed among other local drives available inside your Container.

You can then use the newly created Container drive in the same way as you would use it on a stand-alone server: format it, create new files and folders within the drive, etc.

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 of the Container drive to be created inside the Container. Upon the command completion, this name will be shown as the name of the folder in the <code>X:\vz\root\CT_ID\</code> folder on the Node (e.g. <code>C:\vz\root\101\F</code>). If you log in to the Container via RDP, you can see this name by launching Windows Explorer where it will be listed among your other local drives. The formats that can be used for specifying the drive name are <code>X</code> , <code>X:</code> or <code>X:\</code> where <code>X</code> denotes the name of the drive you wish to create inside the Container.
<code>--size drive_size</code>	Mandatory. The size of the Container drive to be created inside the Container, in kilobytes. If you use this option without specifying any value, the Container drive size will be set to 100 Mb.
<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 located in the <code>X:\vz\private\CT_ID</code> folder on the Node (e.g. <code>C:\vz\private\101\My_File.efd</code>).

Let us assume that you wish to create a new drive having the name of `F:\` and the size of 50 Mb inside Container 101. Besides, you want the drive to be displayed as the `MyVirtualDisk.efd` file inside the `C:\vz\private\101\` folder on the Hardware Node. To this effect, you should issue the following command on your Node:

```
vzctl partadd 101 --drive F:\ --size 51200 --file MyVirtualDisk
```

After the command execution, the newly created drive becomes visible as the `C:\vz\root\101\F` folder and the `C:\vz\private\101\MyVirtualDisk.efd` file on your Hardware Node. You can also log in to Container 101 via RDP and check (e.g. by launching Windows Explorer) that a new drive with the name of Local Disk (`F:`) is now available inside the Container. This drive has the size of 50 Mb and can be used as any other drive on a stand-alone server.

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 may also use the `--delete` option to remove the file from the `X:\vz\private\CT_ID` folder on the Node which, otherwise, remains intact.

For example, to unmount the `F:\` drive that was mounted to Container 101 in the example above and delete the `MyVirtualDisk` file from the `C:\vz\private\101` folder on the Node, you should execute the following command:

```
vzctl partdel 101 --drive F:\ --delete
```

vzctl reinstall

The `vzctl reinstall` command is used to restore the original state of the Container system and application files in case the Container administrator has inadvertently tampered with them and thereby broken something. 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 <code>Administrator</code> credentials from the users' database and creates a clean database as for any new installation.
	Note: During the Container reinstallation, the user login name is always reset to <code>Administrator</code> 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.

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 virtual disk drives of the `compact` type inside your Containers. It has the following syntax:

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

When executed, the command removes unused space in the `compact` disk drive with the name of `CT_drive_name` inside a Container with the ID of `CT_ID`, thus, reducing the amount of space the virtual disk occupies on the Hardware Node. While using this command, please 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 Container virtual disks of the `plain` and `sparse` types.

vzlist

The `vzlist` utility is used to list the Containers existing on the given Hardware Node together with additional information about these Containers. The output and sorting of this information can be customized as needed. The utility has the following syntax:

```
vzlist [-a] [-S] [-o specifier[,specifier...]] [-H] [CT_ID [CT_ID...]]
      [-s specifier | -specifier] [-h [*]pattern[*]]
      [--quite] [--verbose]
vlist -L
vlist --help
```

Here follows the description of available options:

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 parameter[.specifier]</code>	This option is used to 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, they should be separated with commas, as is shown in the synopsis above.
<code>-H, --no-header</code>	Do not display column headers.
<code>-L, --list</code>	List all the parameters available.
<code>--help</code>	Display the usage information and exit.

<p><code>-s, --sort</code> <code>[-]parameter[.specifier]</code></p> <p><code>-h, --hostname</code> <code>hostname_pattern</code></p> <p><code>-n, --name</code></p> <p><code>-N, --name_filter</code> <code>name_pattern</code></p> <p><code>-d, --description</code> <code>desc_pattern</code></p> <p><code>--quiet</code></p> <p><code>--verbose</code></p>	<p>Sort the Containers in the list by the specified parameter. If "-" is given before the name of the parameter, the sorting order is reversed.</p> <p>Display only those Containers that correspond to the specified hostname pattern. The following wildcards can be used: *.</p> <p>If used without any parameters, displays information on all the Containers on the Node together with their names. If you indicate the Container ID after this option, displays information including the Container name on the specified Container only.</p> <p>Displays only the Container that corresponds to the specified name pattern.</p> <p>Displays only the Container whose description corresponds to the specified pattern.</p> <p>Disables logging to the display and to the log file.</p> <p>Sets the log level to the maximum possible value for this <code>vzlist</code> session.</p>
--	---

vzlist Output Parameters and Their Specifiers

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>numproc</code>	NPROC	The number of processes allowed. 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>	<code>CPUUNI</code>	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>	<code>CPUGUARANTEE</code>	This is a positive integer number that determines the minimal guaranteed share of the CPU time, in per cent, the given Container is guaranteed to receive.
<code>cpulimit</code>	<code>CPULIMIT</code>	This is a positive number indicating the CPU time, in percent, the corresponding Container is not allowed to exceed.
<code>vprvmem</code>	<code>VPRVMEM</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>templates</code>	<code>TEMPLATES</code>	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>disktype</code>	<code>DISKTYPE</code>	The type of the Container virtual disk drive. It can be one of the following: <ul style="list-style-type: none"> ▪ <code>plain</code>, i.e. the size a virtual hard disk is fixed and all disk space is allocated during the virtual disk creation. ▪ <code>compact</code>: i.e. only the maximal size of a Container 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 as large as the maximum size specified during the hard disk creation.

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 disk quota statistics for Containers. `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 usage information and exits.

`-b` Sets the batch mode.

`vzquota` understands the following commands:

`setlimit` Allows you to change the quota limits 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:

`-B, --block-hardlimit num` Required. Specifies the disk quota block hard limit, in 1 Kb blocks. This limit cannot be exceeded by the Container.

`--drive name` 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. For example, the following command sets the disk quota limit of 100 Mb for the `F:\` drive inside Container 101:

```
C:\...\Administrator>vzquota setlimit 101 -B 102400 --drive F:\
Command 'setlimit' is successfully finished
```

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, you should 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. Typical outputs of the commands are below:

```
C:\Documents and Settings\Administrator>vzquota stat 101
disk resource usage hardlimit path
C: 1k-blocks 32768 122880 C:\vz\private\101\root.efd
C:\Documents and Settings\Administrator>vzquota show 101
disk resource usage hardlimit path
C: 1k-blocks 32768 122880 C:\vz\Private\101\root.efd
```

Both outputs show that only one disk drive - `C:\` - exists inside Container 101. It is located in the `C:\vz\private\101` folder on the Hardware Node. Container 101 currently uses 32 Mb of disk space available on the `C:\` drive and the hard limit for this drive is set to 120 Mb.

Licensing Utilities

vzlicload

This utility is used for managing Virtuozzo Server licenses installed on your Hardware Node. It has the following syntax:

```
vzlicload [options]
```

The utility accepts the following options:

<code>-p, --product-key</code>	Installs the Virtuozzo license on the Hardware Node.
<code>-i, --stdin</code>	Makes <code>vzlicload</code> use standard input while installing the Virtuozzo license on the Hardware Node.
<code>-f, --license-file <file_path></code>	The full path to the license file containing the Virtuozzo 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 Virtuozzo 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 Virtuozzo 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 (on page 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 <file></code>	Displays the license information from the specified Virtuozzo license file.
<code>-p, --product-key</code>	Displays the license information contained in the specified Virtuozzo 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 can simultaneously run on the Hardware Node in accordance with the license currently installed on this Node.

The utility reports the following statuses for the license:

ACTIVE	The license file is valid and has been successfully loaded into the Virtuozzo management service.
VALID	The license file contains a valid license for this Hardware Node; however, no license is loaded into the Virtuozzo management service.
INVALID	The license file 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 and, therefore, could not be loaded into the Virtuozzo management service.
GRACED	The license file has been successfully loaded into the Virtuozzo management service; however, it has expired and is currently on the grace period (i.e. it is active till the end of the grace period).

Migration Utilities

The migration utilities described in the following subsections allow you to perform the following operations:

- Migrate one or more Container from one Hardware Node to another (e.g. in the case of the Hardware Node upgrade);
- Migrate a physical server running the Windows Server 2003 operating system to a Container on your Hardware Node;
- Move or copy a Container within one and the same Hardware Node, i.e. change the Container ID and/or the Container private area path.

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, you should 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
<code>-A, --src_addr Physical_Server_IP_Addresses</code>	Mandatory. The IP address or hostname of the physical server you are going to migrate to a Container on your Node.
<code>-U, --src_user username</code>	Mandatory. The username to log in to the physical server. The specified user should have the Administrator rights to the server.
<code>-P, --src_pswd password</code>	Mandatory. The password of the user specified as the value of the <code>-U</code> option.
<code>-s, --src_stop</code>	Optional. Stop the physical server after its successful migration. If this option is omitted, the physical server continues running upon the migration completion.
<code>-v, --ve_start</code>	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 disk_drive` 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 `X`, `X:` or `X:\` where `X` 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, Virtual_Network_ID --bridged` 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 **Managing Virtuozzo Network** chapter of **Parallels Virtuozzo Containers User's Guide**.
- `--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, private area, and root paths. Thus, you may use the `vzmlocal` utility either to change the ID and/or the private area path and/or the root path of any existing Container(s) or to clone a Container, i.e. to create a complete copy of an existing Container with different ID and path. It 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:

- `-C, --copy` Clones the source Container instead of moving it.
- `-s, --fast-sid` 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 should 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>`. For example, you can issue the following command to move Container 101 to Container 201, i.e. to change the Container ID from 101 to 201, and to locate the Container private area in the `C:\MyCT` folder on the Hardware Node:

```
C:\...\Administrator>vzlist -a
CTID  NPROC  STATUS  IP_ADDR      HOSTNAME
   1     22  running  192.168.45.63  CT1
  101     -  stopped  10.198.198.101 CT101
C:\...\Administrator>vzmlocal 101:201:C:\MyCT
Command 'vzmlocal' is successfully finished
```

Note: In the current version of Virtuozzo Containers, you can change IDs of stopped Containers only.

Backing-Up Utilities

Any Container is defined by its private area and configuration parameters stored in the registry. Backing up these components allows you to restore the whole Container at any time in case the Container gets broken.

The following subsections provide the complete reference on the Virtuozzo backup and restore utilities:

- `vzabackup`;
- `vzarestore`.

vzabackup

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

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

The only requirements that should be met to execute `vzabackup` is to have a server with Virtuozzo Containers 4.0 for Windows 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 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>	Force performing a full backup.
<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.

- `-C(Letter)` Indicates the level of compression for the resulting Container backup archive. In the current version of Virtuozzo, you can set this level to one of the following:
- `n`: 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.
 - `g`: compress the resulting backup with the normal level of compression. This is the default level of compression used to back up all Nodes/Containers.
 - `b`: 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.
- 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 `vzbackup` execution.
- If this option is omitted, `vzbackup` 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 `vzbackup` 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 `vzbackup` 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`).

<code>-x CT1...</code>	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).
<code>--include-files files_list</code>	Only the specified files and folders will be included in the Container backup.
<hr/>	
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.	
<hr/>	
<code>--exclude-files files_list</code>	The path to the files and folders inside the Container to be excluded from the backup.

vzarestore

The `vzarestore` 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. `vzarestore` can be run on any Hardware Node provided this Node has the Parallels Agent software installed. The utility has the following syntax:

```
vzarestore [CT | -e <CT1, ...> [-x <CT1, ...>] ] [RESTORE_OPTIONS]
           [BACKUP_NODE]
vzarestore -r,--remove <BACKUP_ID ...>
vzarestore -l,--list [LIST_OPTIONS] [BACKUP_NODE]
vzarestore --browse BACKUP_ID [BROWSE_OPTIONS] [BACKUP_NODE]
vzarestore --print-ve-config BACKUP_ID [BACKUP_NODE]
vzarestore --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>vzarestore</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-ve-config</code>	Do not restore the Container configuration file. Can be used only if you are restoring a single Container.
<hr/>	
Note: The Container configuration file is not changed when restoring separate Container files.	
<hr/>	
<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</code> <code>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</code> , <code>--full</code>	Display detailed information on the backed up Containers.
<code>--latest</code>	Display the latest Container backup.
<code>-e CT1, ...</code>	Display the information on the backups for the specified Containers only.
<code>-B</code>	Handle values after the <code>-e</code> option as backup IDs.

Other options which can be used with `vzrestore`:

<code>-d</code>	The path to the folder inside the Container backup archive whose contents is to be shown. This option can be used with the <code>vzrestore --browse BACKUP_ID</code> command only.
<code>--print-ve-config</code> <code>BACKUP_ID</code>	Displays the parameters set in the Container configuration file at the moment of creating the backup with the specified backup ID.
<code>-r</code> , <code>BACKUP_ID</code> <code>--remove</code>	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:

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

Template Management Utilities

Template (or package set) is a set of original application files, repackaged for mounting over Virtuozzo File System. There are two types of templates in Virtuozzo: *OS templates* are used to create new Containers. A typical example of an OS template is the template containing the Windows Server 2003 distribution kit. An OS template cannot be installed into a Container, because any Container is already based on some OS template. *Application templates* can be installed into a Container after its creation; more than one application template may be installed in the same Container. Examples of application templates are the msde template, openssh template, or template with any other application (possibly more than one in single template) not included into the OS template for some reason.

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 [options] --install | --uninstall PACKAGE...
vzpkgdeploy --help
```

The options available to this command are:

<code>-i, PACKAGE...</code>	--install	Installs the template on the Hardware Node.
<code>-u, PACKAGE...</code>	--uninstall	Removes the template from the Hardware Node.

-s	Adds Windows Server 2003 components necessary for running Virtuozzo to the Host OS.
-h, --help	Displays the usage information and exit.
--quiet	Disables logging to the display and to the log file.
--verbose	Sets the log level to the maximum possible value for this vzpkgdeploy 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:

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

Supplementary Tools

vzsveinstall

This utility is used for the Service Container installation. The Service Container is needed for running the Parallels Agent software, which is used by Parallels Management Console and Parallels Infrastructure Manager. The syntax of this command is as follows:

```
vzsveinstall -s SCT_IP_Address [-t OS_Template] [-p] [-w] [-f]
```

This utility creates and starts the Service Container. This utility accepts the following options:

-h, --help	Displays the utility help.
-s < <i>SCT_IP</i> >	Required. The IP address to be assigned to the Service Container.
-t < <i>OS template</i> >	Optional. OS template to use for the Service Container. By default, w2k3 is used.
-p	Optional. Reads the vzadmin password from the standard input. By default, it reads from the console.
-w	Optional. Allows to install the Parallels Infrastructure Manager application to the Service Container.
-u	Updates the Service Container.

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 Virtuozzo license request to the Parallels sales department to obtain a license for running Virtuozzo Containers 4.0. 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 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 each and every Container. 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 `vzcache` 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 or the so-called stubs, to the template area. In case of a significant number of identical files, using this utility results in a notable disk space gain. `vzcache` has the following syntax:

```
vzcache [options] CT_ID-list
```

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

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>-a, --append</code>	Use an existing cache for comparing its files with the regular files inside the specified Containers. Should be used together with the <code>-n</code> option.
<code>-n, --cache-name NAME</code>	The name of the cache subdirectory in the <code>X:\vz\templates__vzcache</code> directory.
<code>-t, --test</code>	Do not perform any caching. Just collect and display the statistics on identical files.
<code>-s, --size-limit N</code>	Do not process files smaller than N bytes. By default, only empty files are not processed.
<code>-m, --min-links M</code>	Cache files found at least M times. The default value is 2.
<code>-T, --template-name NAME</code>	The name(s) of the OS or Application template(s) to restore the links to. Should be used together with the <code>-a</code> option. If you indicate two or more templates at a time, do not insert a gap after commas.
<code>-H, --host</code>	Restore the links to the host system files of the OS template. Should be used together with the <code>-a</code> option.
<code>-f, --force</code>	Force <code>vzcache</code> operations for a single Container and single regular files.

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

To create a cache named `myfirstcache` 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 myfirstcache 153 154
```

To append Container 155 to the existing `myfirstcache` cache:

```
vzcache -a -n myfirstcache 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[: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"> a physical adapter on the Hardware Node having the MAC address of <i>XX-XX-XX-XX-XX-XX</i> or 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>.
<code>net change <i>NET_ID</i> <i>XX-XX-XX-XX-XX-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 CLASS_ID [IP_RANGE]</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 CLASS_ID RATE</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:

<code>-v <CT_ID></code>	Display statistics for a Container with the ID of <i><CT_ID></i> .
<code>-c <class></code>	Show statistics for the <i><class></i> class only.
<code>-a</code>	Display statistics for all classes.
<code>-r K M G</code>	Display the network statistics, which is shown in bytes by default, in the following measurement units: <ul style="list-style-type: none"> ▪ K: display the network statistics in kilobytes; ▪ M: display the network statistics in megabytes; ▪ G: display the network statistics in gigabytes.
<code>--help</code>	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.

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* and *Container 1* are used to designate the *Hardware Node* and the *Service Container*, respectively.

Container 0 is used to designate a *Hardware Node* where the *Virtuozzo Containers* software is installed.

Container 1 is used to designate the *Service Container*.

Hardware Node (or *Node*) is a server where the *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 preinstalled operating system. See also *Template*.

Parallels Infrastructure Manager (or *Infrastructure Manager*) is a tool designed for managing *Hardware Nodes* and all *Containers* residing on them with the help of a standard Web browser on any platform.

Parallels Management Console (or *Management Console*) is a *Virtuozzo Containers* management and monitoring tool with graphical user interface. It is used to control individual *Hardware Nodes* and their *Containers*. *Management Console* is cross-platform and runs on both Microsoft Windows and Linux workstations.

Parallels Power Panel is a means for administering personal *Containers* with the help of a standard Web browser (Internet Explorer, Mozilla, etc.) on any platform.

Parallels Virtuozzo Containers (or *Virtuozzo Containers*) is a complete server automation and virtualization solution allowing you to create multiple isolated *Containers* on a single physical server to share hardware, licenses, and management effort with maximum efficiency.

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

Service Container is a special *Container* automatically created on the *Hardware Node* during the *Virtuozzo Containers* installation and needed to manage your *regular Containers* by means of *Parallels Infrastructure Manager*, *Parallels Power Panel*, and *Parallels Management Console*. Sometimes, the *Service Container* is marked as *Container 1*.

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 Infrastructure Manager*.

Virtuozzo File System (VZFS) is a virtual file system for mounting to *Container* private areas. *VZFS* symlinks are seen as real files inside *Containers*.

Virtuozzo Server license is a special license that you should load to the *Hardware Node* to be able to start using the *Virtuozzo Containers* software. Every *Hardware Node* shall have its own *Virtuozzo Server license*.

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
- Adding Custom Action Scripts • 14
- Administrator
 - Container • 11
 - Hardware Node • 17

B

- Backing-Up Utilities • 40

C

- Configuration File
 - overview • 11
- Configuring Containers • 11
- Configuring Parallels Virtuozzo Containers • 11
- Container
 - backing up • 40
 - creating • 19
 - destroying • 20
 - listing • 31, 32
 - migrating • 36, 37
 - mounting • 21
 - private area • 11, 14, 19, 20, 21, 37
 - restoring • 40
 - starting/stopping • 21

D

- DNS server • 22
- Documentation Conventions • 8

F

- Feedback • 10

G

- Getting Help • 9
- Glossary • 52

H

- Hostname

- Container • 11
- Hardware Node • 37
- setting • 11, 19, 22

I

- IP Address
 - changing • 22
 - deleting • 22
 - setting • 11, 22

L

- Licensing Utilities • 35
- Log
 - Event • 14

M

- Migration
 - Container to Container • 36, 37
 - physical server to Container • 38
- Migration Utilities • 36

N

- Network
 - mask • See Subnet Mask
 - settings • 22
- Node
 - Destination • 37
 - Source • 37

O

- Offline Management • 22
- Organization of This Guide • 7

P

- Parallels Virtuozzo Command Line Interface • 16
- Password
 - changing • 22
 - setting • 22
- Preface • 5

Q

- QoS Scheduler • 22

R

- RAM • See memory

Registry • 21

Resources

CPU • 11, 22

disk space • 11, 22, 33

memory • 11, 22

Windows Server 2003 • 22

S

Scripts • 14

Service Container • 47

Subnet • 22

Subnet Mask • 22

Supplementary Tools • 47

T

Template

application • 45, 46

area • 48

installing • 45, 46

listing • 45

removing • 46

Template Management Utilities • 45

U

Utilities

backup management • 40

Container management • 17, 19, 20, 21, 22,
27, 28, 31

license management • 35

migration management • 36, 37

resources management • 22, 32, 33, 34

supplementary • 47, 48, 49

template management • 45, 46

V

Virtuozzo Server License

checking • 35

installing • 35

vzabackup • 41

vzarestore • 43

vzcache • 48

vzctl • 17

vzctl create • 19

vzctl delete and vzctl destroy • 20

vzctl exec, vzctl exec2, and vzctl enter • 27

vzctl mount and vzctl umount • 21

vzctl mounttext and vzctl umounttext • 28

vzctl partadd and vzctl partdel • 29

vzctl reinstall • 30

vzctl set • 22

vzctl shrink • 31

vzctl start, vzctl stop, vzctl restart, and vzctl
status • 21

vzkeygen • 47

vzlicload • 35

vzlicview • 35

vzlist • 31

vzlist Output Parameters and Their Specifiers •
32

vzlscache • 49

vzmigrate • 37

vzmlocal • 39

vznetcfg • 50

vznetstat • 51

vzp2v • 38

vzpkgadd • 46

vzpkgdeploy • 45

vzpkgls • 45

vzpkgrm • 46

vzquota • 33

vzquota setlimit • 34

vzquota stat and vzquota show • 34

vzsveinstall • 47

vzuncache • 49

vzvssctl • 44