
Parallels

Parallels Virtuozzo Containers for Linux

Templates Management Guide

Version 4.0



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	6
About Parallels Virtuozzo Containers	7
About This Guide	8
Organization of This Guide	8
Documentation Conventions	9
Getting Help	10
Feedback	11
Virtuozzo Templates Overview	12
Managing EZ Templates	14
Understanding EZ Templates	15
EZ Templates Overview	15
EZ Templates Basics	16
EZ Template Directory Structure	17
Differences Between OS and Application EZ Templates	20
EZ Template Lifecycle	20
Creating EZ Template	21
Creating Metafile for EZ Template	22
Using vzmktempl to Create EZ Template	25
Setting Up Repositories and Proxy Servers for EZ Templates	26
Managing Default Repository	27
Creating Local Repository	28
Managing Repositories for Commercial Linux Distributions	32
Creating Proxy Server for EZ Templates	35
Preparing OS EZ Template for Container Creation	40
Installing Application EZ Templates on Hardware Node	41
Upload and Install New EZ Templates on Hardware Node in Management Console	42
Listing EZ Templates	44
Adding Application EZ Templates to Container	46
Keeping EZ Templates Up To Date	47
Updating EZ Templates on Hardware Node	48
Updating OS EZ Template Caches	54
Updating EZ Templates Packages Inside Container	56
Creating Historical Mirror for Backed Up Container	58
Copying EZ Templates to Another Hardware Node	60
Removing Application EZ Template From Container	62
Removing EZ Templates From Hardware Node	63
Converting Containers Based on Standard Templates	65
Managing Standard Templates	67
Standard Template Lifecycle	68
Listing Standard Templates	69
Creating Container on Basis of OS Standard Template	70
Choosing OS Template	70
Creating Container	72

Creating and Installing Application Standard Templates73
Creating and Installing Standard Template Updates.....75
Adding Standard Templates and Template Updates to Containers.....77
Copying Templates to Another Hardware Node.....79
Removing Standard Templates and Template Updates From Containers.....80

Index 81

Table of Figures

Figure 1: Virtuozzo Containers OS Virtualization	7
Figure 2: Virtuozzo Network With Caching Proxy Server	35
Figure 3: Management Console - Choosing Templates for Uploading	42
Figure 4: Management Console - Choosing Target Hardware Nodes	43
Figure 5: Management Console - Reviewing Templates Distribution Parameters	43
Figure 6: Management Console - Listing EZ Templates	44
Figure 7: Management Console - Adding Application EZ Templates to Container	46
Figure 8: Updating Virtuozzo Containers - Welcome Screen	48
Figure 9: Updating Virtuozzo Containers - Selecting Linux Distribution	49
Figure 10: Updating Virtuozzo Containers - Selecting EZ Templates	50
Figure 11: Updating Virtuozzo Containers - Viewing EZ Templates to Install	51
Figure 12: Management Console - Updating EZ Templates	53
Figure 13: Management Console - Caching OS Template	55
Figure 14: Management Console - Copying Templates	61
Figure 15: Removing EZ Templates From Hardware Node - Selecting Templates	64
Figure 16: Management Console - Listing Templates	69
Figure 17: Management Console - Listing Templates	71
Figure 18: Management Console - Copying Templates	79

CHAPTER 1

Preface

In This Chapter

About Parallels Virtuozzo Containers.....	7
About This Guide.....	8
Getting Help.....	10
Feedback.....	11

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:

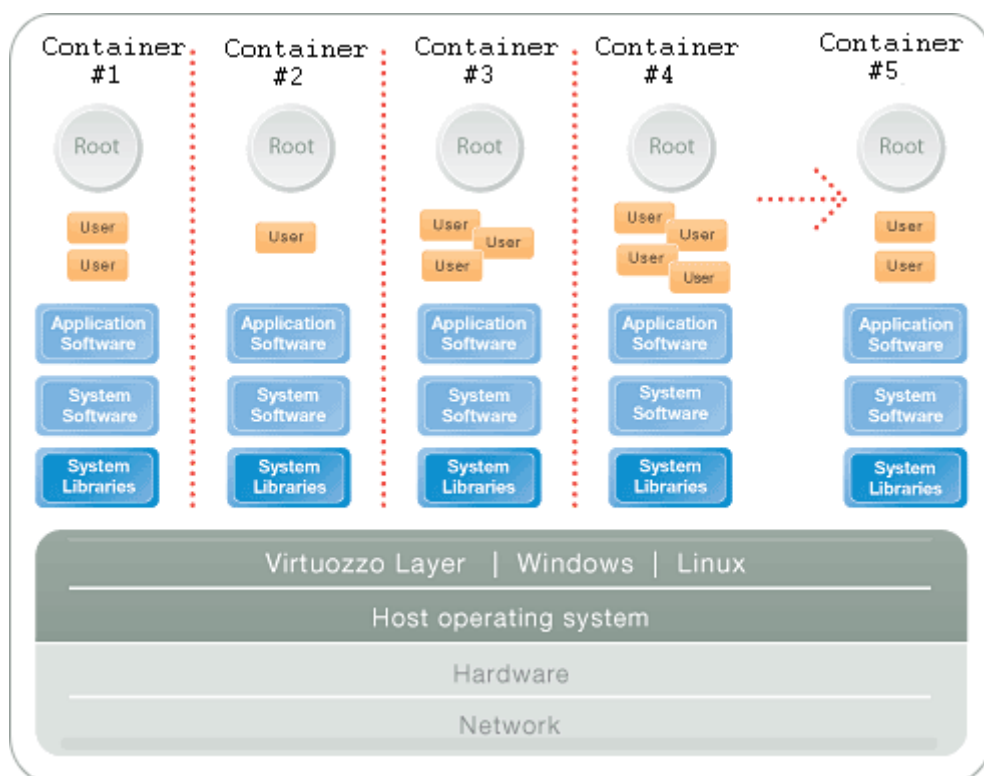


Figure 1: Virtuozzo Containers OS Virtualization

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 meant to provide complete information on Virtuozzo Templates - an exclusive Parallels Virtuozzo technology allowing you to efficiently deploy standard Linux applications inside your Containers and to greatly save your Hardware Node resources (physical memory, disk space, etc.). In particular, you will learn how to create your own application templates and manage them in a number of different ways.

The primary audience for this guide is anyone who is intended to deploy one or several applications inside their Containers and looking for ways to do it with the maximal level of efficiency. To complete all the operations described in this guide, no more than basic Linux administration habits is required.

Organization of This Guide

Chapter 2, Virtuozzo Templates Overview, provides general information on Virtuozzo Templates: what Virtuozzo Templates are, the advantages of their usage in Virtuozzo-based systems, the types of Virtuozzo Templates, etc.

Chapter 3, Managing EZ Templates, provides instructions on managing OS and application EZ templates. You will know how to create and install EZ templates on the Hardware Node, add them to and remove them from Containers, make OS template caches and update them, etc.

Chapter 4, Managing Standard Templates, shows you the way to handle Virtuozzo OS and application standard templates – create and install templates and template updates on the Hardware Node, add them to and remove from Containers, cache OS templates, etc.

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.

The table below presents the existing formatting conventions.

Formatting convention	Type of Information	Example
Triangular Bullet(▶)	Step-by-step procedures. You can follow the instructions below to complete a specific task.	<i>To create a Container:</i>
Special Bold	Items you must select, such as menu options, command buttons, or items in a list.	Go to the Resources tab.
	Titles of chapters, sections, and subsections.	Read the Basic Administration chapter.
<i>Italics</i>	Used to emphasize the importance of a point, to introduce a term or to designate a command line placeholder, which is to be replaced with a real name or value.	These are the so-called <i>EZ templates</i> . To destroy a Container, type <code>vzctl destroy ctid</code> .
Monospace	The names of commands, files, and directories.	Use <code>vzctl start</code> to start a Container.
Preformatted	On-screen computer output in your command-line sessions; source code in XML, C++, or other programming languages.	<pre>Saved parameters for Container 101</pre>
Monospace Bold	What you type, as contrasted with on-screen computer output.	<pre># rpm -V virtuozzo-release</pre>
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

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

Getting Help

In addition to this guide, there are a number of other resources shipped with 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 Virtuozzo major features, and to suggest the relevant procedures for this evaluation.
 - **Parallels Virtuozzo Containers for Linux 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 Linux 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 Parallels Virtuozzo system.
 - **Parallels Virtuozzo Containers for Linux 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 Linux applications inside your Containers and to greatly save the Hardware Node resources (physical memory, disk space, etc.).
 - **Parallels Virtuozzo Containers for Linux Reference Guide.** This guide is a complete reference on all Virtuozzo configuration files and Hardware Node command-line utilities.
- **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 through 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 (<http://forum.swsoft.com/forumdisplay.php?s=&forumid=239>) 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

Virtuozzo Templates Overview

A template in Virtuozzo Containers 4.0 is a set of application files and registry settings installed on the Host operating system in such a way as to be usable by any Container by mounting over Virtuozzo File System. The Virtuozzo Containers software provides tools for creating templates, installing and removing them on/from the Hardware Node, adding them to a Container, etc. Using templates lets you:

- Securely share the RAM among similar applications running in different Containers to save hundreds of megabytes of memory;
- Securely share the files comprising a template among different Containers to save gigabytes of disk space;
- Install applications and patches simultaneously in many Containers;
- Use different versions of an application on different Containers (for example, perform an upgrade only in certain Containers);
- etc.

There are two types of templates in Virtuozzo Containers 4.0. These are OS templates and application templates. An OS template is an operating system and the standard set of applications to be found right after the installation. The Virtuozzo Containers software uses OS templates to create new Containers with a preinstalled operating system. An application template is a set of repackaged software packages optionally accompanied with configuration scripts. Virtuozzo Containers 4.0 uses application templates to add extra software to the existing Containers. For example, you can create a Container on the basis of the `redhat` OS template and add the MySQL application to it with the help of the `mysql` template.

You may perform the following operations with templates:

- create your own application templates;
- upload new templates to the Hardware Node;
- list the templates currently installed on the Hardware Node;
- install templates on and remove them from the Hardware Node;
- add templates to any number of Containers;
- remove templates from the Hardware Node;
- remove templates that are not needed any more from Containers;
- migrate templates from the Source Node to the Destination Node;
- etc.

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

Virtuozzo Containers 4.0 templates can be shipped in one of the following forms:

- As EZ templates. Detailed information on what EZ templates are and how to manage them is provided in the [Managing EZ Templates](#) chapter (p. 14).
- As standard templates. For detailed information on what standard templates are and how to manage them is given in the [Managing Standard Templates](#) chapter (p. 67).

Important! Virtuozzo standard OS and application templates have evolved to new OS and application EZ templates. Although Virtuozzo Containers 4.0 still supports standard templates to provide the compatibility with the previous versions of Virtuozzo Containers, you are highly recommended to use EZ templates on the Hardware Node and inside your Containers. This recommendation becomes even more actual taking into account the fact that all new versions of Linux distributions and applications (e.g. RHEL 5 and all applications to be used with it) are shipped as EZ templates only.

CHAPTER 3

Managing EZ Templates

The given chapter familiarizes you with Virtuozzo EZ templates which have evolved on the basis of Virtuozzo standard templates and provide significant technology enhancements over their counterpart.

In This Chapter

Understanding EZ Templates.....	15
EZ Template Lifecycle.....	20
Creating EZ Template	21
Setting Up Repositories and Proxy Servers for EZ Templates	26
Preparing OS EZ Template for Container Creation	40
Installing Application EZ Templates on Hardware Node	41
Upload and Install New EZ Templates on Hardware Node in Management Console	42
Listing EZ Templates	44
Adding Application EZ Templates to Container.....	46
Keeping EZ Templates Up To Date.....	47
Creating Historical Mirror for Backed Up Container	58
Copying EZ Templates to Another Hardware Node	60
Removing Application EZ Template From Container	62
Removing EZ Templates From Hardware Node.....	63
Converting Containers Based on Standard Templates	65

Understanding EZ Templates

EZ Templates Overview

EZ templates are part and parcel of the Virtuozzo philosophy because they provide a way of sharing resources among lots of Containers, thus, enabling huge savings in terms of disk space and memory. For example, when you install and cache an OS template on the Hardware Node, Virtuozzo Containers 4.0 creates the `/vz/template/<name_of_the_OS>` directory containing all the OS files that can be shared among Containers. When a Container based on this template is created, it contains only symlinks to the OS template files. These symlinks occupy very little space on the hard disk. They are situated in the so-called *private area* of the Container. The corresponding directory is `/vz/private/<CT_ID>`. The private area of a Container contains not only symlinks to the necessary template files, but also the copy-on-write area of the Container (the area for storing the information about those changes that the Container makes to the template files; this information pertains only to the given directory) and all the private Container files. When the Container is started, this private area is mounted as Virtuozzo File System (VZFS) to the `/vz/root/<CT_ID>` directory. This directory is seen as the root directory from within the Container. And, which is the pivot of it all, thanks to the VZFS, the symlinks of the Container private area are seen as real files there!

Thus, the most important directories in the `/vz` partition are the following:

- `/vz/template` - contains OS and application files shared among Containers;
- `/vz/private` - contains VZFS symlinks to template files;
- `/vz/root` - contains Container mounted symlinks and all other Container files.

The relation of these directories may be represented as below:

`/vz/template` (real files) → `/vz/private` (symlinks) → `/vz/root` (symlinks seen as real files in `/` for the Container)

While you are able to perform all kinds of tasks within a Container including building RPM packages and installing them, Virtuozzo Containers 4.0 provides an easy and far more efficient way of installing the applications you need inside Containers. In the same way as you install an OS template on the Virtuozzo system in order to create any number of Containers on its basis and share its resources, you can install application templates in Virtuozzo Containers 4.0 in order to share application files among any number of Containers. You can then add these applications to any number of Containers with a single command.

EZ Templates Basics

All OS and application EZ templates are defined by the following features:

- EZ templates do not carry all the necessary package files inside themselves (as opposed to standard templates). They contain only information about what packages should be installed on the Hardware Node to make the templates fully operational and from what network repository these packages should be downloaded.

Note: For the sake of brevity, we will be saying throughout this guide that packages are included in EZ templates, which actually means that EZ templates contain the information on the corresponding packages without carrying the packages themselves.

- The dependencies of software packages included in an EZ template are automatically resolved during the packages installation on the Hardware Node. So, if the specified packages require other packages to be installed, these packages are also downloaded from the repository and installed on the Node. In case a package has requirements that conflict with existing software on the Node or any dependencies for the package being installed cannot be satisfied, the package installation process fails without making any changes to the system.
- The EZ templates technology allows you to use the original OS and application vendor's packages and to receive the updated packages from a central repository right after their release.

One of the basic concepts in the EZ template technology is the concept of 'repository' where software packages for the given EZ template are stored. A repository is a prepared directory or web site containing the packages and index files for Linux operating systems and/or any of their applications. An example of such a repository is the repository located at the <http://mirrors.usc.edu/pub/linux/distributions/fedora/> web site and storing software packages for the Fedora Core releases. Using repositories gives you the following advantages:

- Software packages included in the given EZ template do not contain versions, but only names (e.g. `bash`, `crontabs`). So, you always update any package included in the EZ template to its latest version available in the repository.
- As a result of the fact that a list of packages does not provide their versions, EZ templates do not have versions either (e.g. `redhat-e15-x86`). Thus, you install any EZ template on the Hardware Node only once and, after that, use the installed template to update the packages inside any Container where it is applied.
- You can create several OS EZ template sets for one and the same Linux operating system. Any OS EZ template you are provided with has the default packages set included in it and is called the base OS EZ template. However, you can make your own OS EZ template sets (the so-called non-base OS EZ template sets) which may differ from the corresponding base template:
 - In the number of packages included in these EZ template sets;
 - In the number and location of repositories to be used for these EZ template sets;
 - In the number and kind of scripts to be executed on different EZ template sets lifecycle stages, etc.

Non-base OS EZ template sets should have their own names and are created by appending a random identifier to the base OS EZ template name. For example, if you wish your Container to run Red Hat Enterprise Linux 5 and to function as a Linux-based server only, you can create the `redhat-e15-x86-server` OS EZ template set and include only those packages in it that are needed for performing main server tasks. So, you can specify packages to be used for setting up file and print sharing and exclude all the packages for graphical interfaces (GNOME and KDE).

Virtuozzo Containers 4.0 provides you with a `vzpkg` tool allowing you to automatically locate and obtain the correct packages for your EZ templates from one or several package repositories. The packages are downloaded from the repository and installed on the Hardware Node in one of the following cases:

- When creating a cache for an OS EZ template;
- When updating an existing OS EZ template cache (if there are new packages available in the repository);
- When adding an application EZ template or package to the first Container;
- When updating EZ templates or software packages inside a destination Container.

Note: Detailed information on how to manage software package repositories is provided in the [Setting Up Repository for EZ Templates](#) section (p. 26).

EZ Template Directory Structure

All EZ templates and the software packages included in them and installed on the Hardware Node are located in the so-called template area the path to which is set as the value of the `TEMPLATE` variable in the `/etc/vz/vz.conf` file. By default, the `/vz/template` directory is used. The template area includes two main subdirectories:

- The `cache` subdirectory where the tar archive of the potential private area of a Container based on the corresponding OS EZ template is stored. The tar archive is created during the OS EZ template caching. Please keep in mind that the OS EZ template should be obligatorily cached before you can start creating Containers on its basis.
- The template directory having the name of `<os_name>/<os_version>/<arch>` where:
 - `<os_name>` denotes the name of the Linux distribution for which the OS EZ template is created (e.g. `redhat`, `centos`, `fedora-core`);
 - `<os_version>` is the version of the Linux distribution specified as `<os_name>` (e.g. 7 or 8), and
 - `<arch>` denotes the microprocessor architecture where the OS EZ template is to be run (`x86`, `x86-64`, `ia64`).

For example, after installing the 32-bit version of the Fedora 8 EZ template, the `/vz/template/fedora-core/8/x86` directory on the Hardware Node is created.

In its turn, the `<arch>` directory contains the following subdirectories and areas:

- The template configuration subdirectory including:
 - the `config/os/default` directory where the appropriate configuration files for the base OS EZ template are stored.

- the `config/os/<setname>` directory where the appropriate configuration files for non-base OS EZ templates, if any, are stored.
- the `config/app/<app_name>/default` directory where the appropriate configuration files for the base application EZ template are stored. This directory is created if at least one application EZ template for the given OS EZ template is installed on the Hardware Node.
- the `config/app/<app_name>/<setname>` directory where the appropriate configuration files for non-base application EZ templates, if any, are stored.
- The packages area containing a number of software packages downloaded from the repository and installed on the Hardware Node. The installed files can be shared among Containers, i.e. when a Container based on the given OS EZ template is created or application EZ templates are added to any Container, it contains only symlinks to the template files in the packages area. The installed package has the following structure:

```
<name>-<epoch>:<version>-<release>.<arch>
```

where:

- `<name>` is the package name;
- `<epoch>` denotes the package epoch;
- `<version>` indicates the package version;
- `<release>` is the package release, and
- `<arch>` denotes the microprocessor architecture where the package is to be used.

Examples of the installed software packages are the `zlib-1.2.3-14.fc8.i386.rpm` or `glib-1.2.10-28.fc8.i386.rpm` packages that can be found in the packages area on the Hardware Node after installing and caching the Fedora 8 OS EZ template.

- One or several subdirectories containing the packages comprising the corresponding OS EZ template. The directories have the following names:
 - `baseN` for the base OS template;
 - `<setname>N` for the non-base OS template with the name of `<setname>`, if any;
 - `<appname>N` for the base application template, and
 - `<appname>-<setname>N` for the application template with the name of `<setname>`, if any.

`N` denotes the index number of the URL specified in the `repositories/mirrorlist` file (please see information on the `repositories/mirrorlist` file below).

As has been mentioned above, the configuration directory (i.e. `/<template_area>/<template_directory>/config`) contains a number of subdirectories storing 'EZ templates'-related configuration files. The contents of these subdirectories can vary depending on whether it is a base OS EZ template or a non-base one and on the EZ template type (OS or application template). The most important configuration files are listed below:

- Data files:
 - `packages`: contains a list of software packages names included in the corresponding EZ template.
 - `package_manager`: specifies the packaging system used to handle the EZ template.

- `repositories`: a list of repositories where the packages comprising the EZ template are stored.
- `mirrorlist`: one or several URLs to the file containing a list of repositories from where the packages comprising the EZ template are to be downloaded.
- `distribution`: the name of the Linux distribution for which the EZ template is created. This file should be absent for application EZ templates.
- `summary`: brief information on the EZ template.
- `description`: detailed information on the EZ template. As distinct from the `summary` file, it can contain additional data relevant for the EZ template.
- `environment`: a list of environment variables set in the form of `key=value`.
- **Scripts:**
 - `pre-cache`: this script is executed before installing the packages included in the EZ template on the Hardware Node.
 - `post-cache`: this script is executed after installing the packages included in the EZ template on the Hardware Node.
 - `pre-install`: this script is executed before adding the EZ template to or installing the package inside the Container.
 - `post-install`: this script is executed after adding the EZ template to or installing the package inside the Container.
 - `pre-upgrade`: this script is executed before updating the packages inside the Container.
 - `post-upgrade`: this script is executed updating the packages inside the Container.
 - `pre-remove`: this script is executed before removing the application EZ template/package from the Container.
 - `post-remove`: this script is executed after removing the application EZ template/package from the Container.
- **Document files**: one or several files with arbitrary names containing the information on the EZ template (e.g. `README`).

Note: Detailed information on the files contained in the OS template configuration directory is provided in the *EZ Template Management Utilities* chapter of the *Parallels Virtuozzo Containers Reference Guide*.

While working with EZ template configuration files, please keep in mind the following:

- The `packages` file should be specified for all EZ templates.
- The `packages`, `package_manager`, and `repositories/mirrorlist` files should be specified for all base OS EZ templates.
- The `package_manager` and `distribution` files should be specified for all base OS EZ templates and absent for non-base OS EZ templates and all application EZ templates.
- The information from the `repositories/mirrorlist` file(s) created for non-base OS and all application EZ templates is added to that in the `repositories/mirrorlist` file(s) for the base OS EZ template.

Differences Between OS and Application EZ Templates

Actually, there are four major differences between OS EZ templates and application templates:

- OS templates are used to create new Containers, whereas application templates provide additional software for already created Containers.
- You may define a list of compatible templates and a list of required templates for application templates.
- OS templates and their updates are cacheable, whereas application templates and their updates are not.

The last point needs further explanation. The fact is that creating a huge number of symlinks to the OS template when creating a Container (i.e. its private area) may take a very considerable amount of time. To reduce the time needed for creating a new Container, you should use the `vzpkg create cache` command allowing you to make a tarball of the potential private area of a Container based on the corresponding template. This tarball is also located in the `/vz/template/cache` directory. When a Container is being created, the tarball is simply deployed into the Container private area.

EZ Template Lifecycle

An EZ template has the following development stages:

- 1 Any EZ template should be first installed on the Hardware Node. The `vzpkg install template` command enables you to install OS and application EZ templates on the Node.
- 2 The EZ template should be cached by using the `vzpkg create cache` command. This step is required for OS EZ templates only. As a result of the `vzpkg create cache` execution, the necessary packages included in the OS EZ template are downloaded from the network repository, installed in the `/vz/template/<os_name>/<os_version>` directory on the Hardware Node, and a gzipped tarball for the OS EZ template is created and put to the `/vz/template/cache` directory.
- 3 Either a new Container is created on the basis of the cached OS EZ template with the `vzctl create` command or an application EZ template is added to any number of Containers by using the `vzpkg install` command. In the latter case, the necessary application files are first downloaded from the network repository and installed on the Hardware Node and then the VZFS links to the installed files are created and added to the Container private area.
- 4 An obsolete EZ template applied to the corresponding Container can be updated by using the `vzpkg update` command.
- 5 Any EZ template excluding OS EZ templates can be removed from the Container with the `vzpkg remove` command.
- 6 An EZ template that is not used by any Container may be completely removed from the Hardware Node with the `vzpkg remove template` command.

All these steps may be performed either through the Virtuozzo command line utilities indicated above or with the help of Parallels Management Console.

Creating EZ Template

Virtuozzo Containers 4.0 is shipped with a certain number of OS (`fedora-core-8-x86`, `redhat-el5-x86`, etc.) and application (`proftpd-fedora-core-8-x86`, `php-redhat-el5-x86`, etc.) EZ templates, which is usually sufficient to deploy the main Linux distributions and their applications inside your Containers. However, you may wish to create your own OS and application EZ templates and use them to base your Containers on or run different applications inside your Containers, respectively. To create an EZ template, you should:

- Install the `vztt-build` package shipped with Virtuozzo Containers 4.0. This package contains the `vzmktmpl` utility and a number of template metafile samples and scripts needed to create EZ templates. The `vztt-build` package is located in the `/virtuozzo/RPMS` directory of your Virtuozzo Containers distribution and is not installed by default during the Virtuozzo Containers installation. So, you should manually install this package on your Hardware Node with the `rpm -i` command before proceeding with the EZ template creation. For example:

```
# rpm -ihv /virtuozzo/RPMS/vztt-build-4.0.0-84.swsoft.i386.rpm
Preparing...      ##### [100%]
 1:vztt-build     ##### [100%]
```

- Make a special metafile that will be used as the basis for the EZ template creation.
- Run the `vzmktmpl` utility and pass the corresponding options to it, if needed.

The following subsections concentrate on the last two tasks in detail.

Creating Metafile for EZ Template

On the first step, you should build a metafile - a special file serving as the basis for your new OS or application EZ template and used by the `vzmktmp1` utility during the template creation. A metafile is a text file having the `.metafile` extension and containing a list of parameters for your EZ template.

Let us assume that you wish to create an OS EZ template for the Ubuntu 6.06 Linux distribution which is to be run under x86-64-bit processors. To create a metafile for the Ubuntu EZ template, you should perform the following operations:

- 1 Create a new metafile with an arbitrary name (e.g. `ubuntu-64.metafile`) and open it for editing. The easiest way to do it is to make a copy of the appropriate metafile sample located in the `/usr/share/vztt/samples` directory on the Node and configure it to meet your demands. The `/usr/share/vztt/samples` directory contains the metafile samples of virtually all major Linux distributions. For example, you can use the provided `/usr/share/vztt/samples/ubuntu-6.06-x86_64/metafile` file as the basis for creating the `ubuntu-64.metafile` metafile:

```
# cp /usr/share/vztt/samples/ubuntu-6.06-x86_64/metafile /root/ubuntu/ubuntu-64.metafile
# vi /root/ubuntu/ubuntu-64.metafile
```

Note: When creating an EZ template metafile, please pay close attention to its correct configuration. EZ templates made out of incorrect metafiles may cause the Containers you will create on the basis of these templates to malfunction. For example, you may have one or more unwanted services running inside your Containers (such as `mingetty` and `klogd`), or the 'passwordless' root user access to your Container can be enabled. Therefore, we highly recommend that you use the appropriate EZ OS template metafile samples shipped with Virtuozzo Containers 4.0 and configure them in accordance with your demands.

- 2 Provide the following information in the file:
 - Specify the name of the Linux distribution for which you are creating the OS EZ template as the value of the `%osname` parameter. For example:

```
%osname
ubuntu
```

You can specify any name you like as the value of the `%osname` parameter. This name will then be assigned to the template directory on the Node where the base OS EZ template will be installed (e.g. `/vz/template/ubuntu`).

Note: If you are creating an application EZ template, you should make sure that the value of the `%osname` parameter corresponds to the name of the main template directory on the Hardware Node where the OS EZ template of the Linux distribution, under which your application EZ template is to be run, is installed. More detailed information on how EZ template directories are organized is provided in the [EZ Template Directory Structure](#) subsection (p. 17).

- Specify the version of the Linux distribution for which you are creating the OS EZ template as the value of the `%osver` parameter:

```
%osver
6.06
```

You can specify any name you like as the value of the `%osver` parameter. This name will be then assigned to the subdirectory on the Node which will denote the version of your Linux distribution (e.g. `/vz/template/ubuntu/6.06`).

Note: If you are creating an application EZ template, you should make sure that the value of the `%osver` parameter corresponds to the name of the subdirectory located in the main template directory on the Hardware Node and denoting the version of the Linux distribution specified as the value of the `%osname` parameter.

- Provide the information on the microprocessor architecture where the OS EZ template is to be run as the value of the `%osarch` parameter:

```
%osarch
x86_64
```

You can set the value of the `%osarch` parameter to one of the following:

- * `x86`: this value should be specified if your EZ template is to be used on 32-bit platforms;
- * `x86_64`: this value should be specified if your EZ template is to be used on x86-64-bit platforms (e.g. on servers with the AMD Opteron and Intel Pentium D processors installed);
- * `ia64`: this value should be specified if your EZ template is to be used on IA-64-bit platforms (i.e. on servers with the Itanium 2 processor installed).

As our template is intended for use on x86-64-bit platforms, the value of the `%osarch` parameter should be set to `x86_64`.

- Indicate what packages are to be included in your OS EZ template. The names of the packages should correspond to the names of real packages (with or without indicating the package version: e.g. `wget` or `wget=1.9.1`) that are stored in the repository to be used for managing the Ubuntu 6.06 OS EZ template. These packages will be downloaded from the package repository and installed on the Hardware Node when caching the OS EZ template. The packages for Ubuntu 6.06 should be indicated as the value of the following parameters:
 - a** `%packages_0`: provide a list of packages to be used for creating a minimal Ubuntu `chroot` environment. These packages should correspond to those installed on a standalone server on the first stage of the Ubuntu distribution installation. The packages will be installed on the Hardware Node one by one in the specified order during the OS EZ template caching. If you wish several packages to be simultaneously installed on the Node, you should specify the package names on a single line and separate them by spaces.
 - b** `%packages_1`: specify a list of 'base' packages for the Ubuntu 6.06 distribution. These packages are needed to install the packages listed as the value of the `%packages` parameter.
 - c** `%packages`: list the packages that are not specified as the values of `%packages_0` or `%packages_1` and that you to include in the Ubuntu OS EZ template.

Note: 1. The `%packages_0` and `%packages_1` parameters should be specified only if you are creating Debian/Ubuntu-based EZ templates, which is caused by the fact that the Debian/Ubuntu distribution installation is carried out in three stages.

2. Please make sure that you have a clear understanding of what packages are to be included in `%packages_0` and `%packages_1`; otherwise, consult the corresponding Ubuntu documentation.

- Specify the package manager to be used for handling the OS EZ template.

```
%package_manager
dpkgx64
```

Depending on the Linux distribution for which you are creating the template or under which the template will be used, you should set the following values for the `package_manager` parameter:

32-bit Linux distributions:

- * `rpm44x86`: Red Hat Enterprise Linux 5, Fedora Core 4, 5, and 6 and Fedora 7 and 8;
- * `rpm43x86`: Red Hat Enterprise Linux 3 and 4 (with the 2.6 kernel and NPTL support), CentOS 4 and 5;
- * `rpm41x86`: SUSE Linux Enterprise Server 10 and SUSE Linux 10.x where *x* denotes the minor number of the SUSE Linux 10 release (e.g. 10.1 or 10.2);
- * `rpm41s9x86`: SUSE Linux Enterprise Server 9;
- * `dpkg`: Debian and Ubuntu;

64-bit Linux distributions for x86-64 processors:

- * `rpm44x64`: Red Hat Enterprise Linux 5, Fedora Core 4, 5, and 6 and Fedora 7, 8;
- * `rpm43x64`: Red Hat Enterprise Linux 3 and 4 (with the 2.6 kernel and NPTL support) and CentOS 4 and 5;
- * `rpm41x64`: SUSE Linux Enterprise Server 10 and SUSE Linux 10.x where *x* denotes the minor number of the SUSE Linux 10 release (e.g. 10.1 or 10.2);
- * `rpm41s9x64`: SUSE Linux Enterprise Server 9;
- * `dpkgx64`: Debian and Ubuntu;

64-bit Linux distributions for ia64 processors:

- * `rpm44i64`: Red Hat Enterprise Linux 5;
- * `rpm43i64`: Red Hat Enterprise Linux 3 and 4 (with the 2.6 kernel and NPTL support) and CentOS 4 and 5;
- * `rpm41s9i64`: SUSE Linux Enterprise Server 9;
- * `rpm41i64`: SUSE Linux Enterprise Server 10;
- * `dpkgi64`: Debian and Ubuntu.

- Define a list of repositories where the packages comprising the EZ template are stored:

```
%repositories
http://archive.ubuntu.com/ubuntu breezy main restricted universe
multiverse
http://archive.ubuntu.com/ubuntu breezy-updates main restricted
universe multiverse
http://archive.ubuntu.com/ubuntu breezy-security main restricted
universe multiverse
```

All the aforementioned parameters should be necessarily set in any metafile to be used for the OS EZ template creation. You can also specify a number of supplementary parameters in your metafile (e.g. you can define the `version` and `release` parameters). Detailed information on the parameters that should be present in a metafile intended for creating an OS EZ template and all the additional parameters is provided in the `vzpkg.metafile` manual pages; examples of metafiles for EZ templates can also be found in the `/usr/share/vztt/samples` directory on the Hardware Node.

- 3 Save the file.

Using `vzmktmpl` to Create EZ Template

After you have successfully built a metafile for your OS EZ template, you can use the `vzmktmpl` utility to create the EZ template. When executed, the utility takes the parameters specified in the metafile and creates a new EZ template. Let us take the metafile prepared in the previous subsection and create the `ubuntu` EZ template on its basis. Assuming that the `ubuntu-64.metafile` file is located in the `/root/ubuntu` directory on the Hardware Node, this can be done as follows:

```
# vzmktmpl /root/ubuntu/ubuntu-64.metafile
Executing(%prep): /bin/sh -e /var/tmp/rpm-tmp.65461
+ umask 022
+ cd /usr/src/redhat/BUILD
+ cd /usr/src/redhat/BUILD
+ rm -rf ubuntu-6.06-x86-ez
...
```

You can also specify a number of additional options during the `vzmktmpl` execution. For example, you can use the `--post-install` option to indicate the path to the script which is to be executed after caching the `ubuntu` OS EZ template. For detailed information on all options that can be passed to `vzmktmpl`, please turn to the utility manual pages; script examples can also be found in the `/usr/share/vztt/samples` directory on the Hardware Node.

Upon the command completion, the created EZ template is put to your current working directory:

```
# ls /root
ubuntu-6.06-x86_64-ez-1.0-1.noarch.rpm
...
```

As can be seen from the example above, a new EZ template with the name of `ubuntu-6.06-x86_64-ez-1.0-1.noarch.rpm` has been successfully created and placed to the `/root` directory on the Node. To start using your new OS EZ template (i.e. start creating new Containers on its basis), you should first install it on the Hardware Node using the `vzpkg install template` command and then cache it with the `vzpkg create cache` command.

Note: Non-base OS EZ templates inherit a number of properties from their base OS EZ templates. So, before installing a non-base OS EZ template, please make sure that the corresponding base OS EZ template is installed on the Hardware Node.

Setting Up Repositories and Proxy Servers for EZ Templates

If you are going to use OS and application EZ templates inside your Containers, you should first have one or several repositories with software packages prepared for your EZ templates. Package repositories are required for the EZ templates functioning due to the fact that these templates do not carry all the necessary package files inside themselves. They contain only information about what packages are included in the corresponding EZ template and from what repository they should be downloaded. In Virtuozzo Containers 4.0, you can make use of the following package repositories:

- Default repositories. When you install an EZ template on the Hardware Node, it is preconfigured to use official vendor's file sources (e.g. RPMs from the Fedora web site) and a number of packages provided by Parallels and needed for the correct EZ template functioning. So, you can start using the default repositories right after the EZ template installation on the Node. Please note that the default repositories are provided for non-commercial versions of Linux distributions only.
- Local repositories. You can build your own local repositories, which allows you:
 - To greatly save on network bandwidth when deploying package updates to several Hardware Nodes in your network.
 - To organize your own package repository if public repositories provided by Linux OS and application vendors are not compatible with a `vzpkg` tool used to manage Virtuozzo EZ templates.

Note: The `vzpkg` tool supports all the repositories that can be used by the `yum` utility (version 2.4.0 and higher) and the `apt` utility.

- Special repositories used to store software packages for commercial Linux distributions (e.g. Red Hat Enterprise Linux 5 or Suse Linux Enterprise Server 10).

Along with setting up the aforementioned package repositories, you can also create special caching proxy servers and use them to efficiently manage your OS and application EZ templates:

- The `vzpkgproxy` utility allows you to automatically set up any computer on your network to serve as a caching proxy server for EZ templates of non-commercial Linux distributions (e.g. Fedora 7, 8 or CentOS 4 or 5).
- The `vzrhnp` utility allows you to create special RHN Proxy Servers to effectively manage the software packages included in the RHEL 4 and 5 OS EZ templates.

Managing Default Repository

When you install an OS EZ template on the Hardware Node, it is preconfigured to use one or several package repositories storing Linux OS vendor's file sources. The path to the repositories for the corresponding OS EZ template and all its application EZ templates is automatically set during the EZ template installation in the `mirrorlist` or `repositories` files located in the `/vz/template/<os_name>/<os_version>/<arch>/config/os/default/` directory on the Node. Thus, you do not have to perform any additional operations to start using an installed OS EZ template (i.e. cache it and create Containers on its basis). The only requirement that your system should meet is to have an Internet connection to access the servers where the specified repositories are located.

Let us assume that you wish to use the Fedora 8 OS EZ template to base your Containers on. To this effect, you should install the `fedora-core-8-x86-tmpl-4.0.0-1.swoft.noarch.rpm` EZ template on the Hardware Node:

```
# vzpkg install template \\  
  fedora-core-8-x86-tmpl-4.0-1.swoft.noarch.rpm  
Preparing... ##### [100%]  
 1:fedora-core-8-x86-tmpl ##### [100%]
```

After the OS EZ template has been successfully installed on the Node, you can see the path to the default repositories storing the Fedora OS packages in the `/vz/template/fedora-core/8/x86/config/os/default/mirrorlist` file. For example:

```
# cat /vz/template/fedora-core/8/x86/config/os/default/mirrorlist  
$SW_SERVER/download/mirrors/fedora-core-8  
$SW_SERVER/download/mirrors/updates-released-fc8  
$FC_SERVER/download/mirrors/fedora-core-8  
$FC_SERVER/download/mirrors/updates-released-fc8
```

where `$SW_SERVER` and `$FC_SERVER` denote the variables whose values are specified in the `/etc/vztt/vztt.conf` file on the Hardware Node. In our case these values will be the IP addresses or hostnames of the Parallels and Fedora web servers, respectively. The Fedora server stores all RPM packages for the Fedora 8 release, whereas the Parallels server keeps a number of software packages needed for the correct `fedora-core-8-x86` EZ template operation. The priority according to which software packages are downloaded from the specified servers is determined by the repositories order in the `mirrorlist` file. So, in the example above RPM packages from the Parallels server will be downloaded first and then - all the packages from the Fedora server.

Note: If you are going to use a commercial version of the Linux distribution (e.g. Red Hat Enterprise Server 4 or 5), the path to the package repository is not set by default and you have to create a special repository which will store software packages and their updates for the corresponding OS EZ template. Detailed information on how to manage repositories for commercial Linux distributions is provided in the [Managing Repositories for Commercial Linux Distributions](#) subsection (p. 32).

Software packages will be downloaded and installed on the Hardware Node from the repositories specified in the `mirrorlist` or `repositories` files in one of the following cases:

- When you create the cache of the OS EZ template;
- When updating an existing OS EZ template cache (if there are new packages available in the repository);

- When you add the application EZ template or package to the Container for the first time;
- When you update the EZ templates or software packages inside your Container.

You can easily add your own repositories (e.g. storing unofficial software packages) to be used by your EZ templates. To this effect, you should only create the `repositories` file in the `/vz/template/<os_name>/<os_version>/<arch>/config/os/default` directory on the Hardware Node, if it is not present, and specify the path to the needed repository. For example, to add the extra repository located at `http://mirrors.dotsrc.org/jpackage/` and keeping Java packages for Fedora 8, you should perform the following operations:

- 1 Create the `/vz/template/fedora-core/8/x86/config/os/default/repositories` file on the Hardware Node, if it is not yet present.
- 2 Add the following string to the file:

```
http://mirrors.dotsrc.org/jpackage
```
- 3 Save the file.

Creating Local Repository

You can also set up a local repository where the packages included in EZ templates will be downloaded and stored. Organizing your own local repository results in less bandwidth consumption and rapid software updates inside your Containers. You may also wish to build a local repository if OS vendors or third-party software developers do not provide repositories for their versions of Linux distributions compatible with the `vzpkg` tool; so, you have to manually find and install new applications or updates inside your Containers.

Note: The `vzpkg` tool supports all the repositories that can be used by the `yum` utility (version 2.4.0 and higher) and the `apt` utility. For detailed information on these utilities, please see their man pages.

The process of setting up your local repository includes the following main stages:

- Obtaining software packages comprising the given Linux distribution. This can be done by:
 - Simply copying the needed packages from your distribution disks or the OS vendor's web site.
 - Copying the needed packages from the `up2date` cache. You can use this way only if the OS EZ template for which you are going to create a local repository corresponds to the OS installed on your Hardware Node and you ran the `up2date` utility at least once to update your Host OS. Please consult the documentation for `up2date` to learn about its cache functionality.
- Creating the metadata repository from a set of the copied software packages with the `createrepo` utility. This step can be omitted if you are going to create a repository which will be a mirror of a public repository.
- Making your repository accessible for Containers users. You can let Container users access your repository in one of the following ways:
 - By using the `http` protocol. In this case the repository should represent a web site containing software packages for the EZ template.
 - By using the `ftp` protocol. In this case the repository should represent an FTP site containing software packages for the EZ template.

- By using the `file` protocol. In this case the repository should represent a directory path (e.g. on your local Hardware Node) containing software packages for the EZ template.

While the first two protocols allow you to remotely (i.e. from Hardware Nodes located in other networks) access the created repository, the last way can be used within your local Hardware Node only.

Let us assume that you wish to build a local package repository for the Fedora 8 EZ template where the RPM packages for Fedora 8 will be downloaded and stored. The repository will be used by Hardware Nodes from both your local network and other networks, and it will deliver packages download for users through the `http` protocol. In our example below, we presume the following:

- The package repository will be located inside Container 101. You can use any Virtuozzo OS template (e.g. `redhat-el5-x86`) to base the Container on.

Note: We recommend that you always place your local repositories inside separate Containers not to compromise the Hardware Node security. In particular, it is of significant importance if you are going to provide access to your repositories through the `http` and `ftp` protocols.

- Container 101 is started. It has the IP address of `123.145.145.123` assigned to it and can be accessed from other networks.
- The `apache` web server is running inside Container 101 and the default document root for `apache` is `/var/www/html`, i.e. the `apache` web server stores its sites in the `/vz/root/101/var/www/html` directory on the Hardware Node.
- The `apache` user and group inside Container 101 are `apache`.

To create a local repository for Fedora 8, you should perform the following operations:

- 1 Install the `fedora-core-8-x86` OS EZ template shipped with Virtuozzo Containers 4.0:

```
# vzpkg list
redhat-el5-x86
# vzpkg install template \
  fedora-core-8-x86-tmpl-4.0.0-1.swsoft.noarch.rpm
Preparing... ##### [100%]
 1:fedora-core-8-x86-tmpl ##### [100%]
# vzpkg list
fedora-core-8-x86
redhat-el5-x86
```

- 2 Change to the `/vz/root/101/var/www/html` directory and create two subdirectories within it:

Note: You can also log in to Container 101 and perform the operations described in Steps 2-8 from inside the Container. However, in this case your working directory inside Container 101 should be `/var/www/html`; you will need to install the `createrepo` package inside the Container and grant the Container access to the CD-ROM drive on the Hardware Node.

- The subdirectory where the base RPM packages for Fedora 8 will be stored:

```
# mkdir -p download/fedora/core/8/i386/os/Fedora/RPMS
```

- The subdirectory where the updated versions of RPM packages for Fedora 8 will be stored:

```
# mkdir -p download/fedora/core/updates/8/i386
```

- 3** Copy all the packages comprising the Fedora 8 distribution (e.g. from your Fedora 8 distribution disks) to the `download/fedora/core/8/i386/os/Fedora/RPMS` directory on the Hardware Node. For example, you can run the following commands for each of the Fedora 8 CDs to get the source files for the repository:

```
# mount /media/cdrom
# cp /media/cdrom/Fedora/RPMS/*.rpm download/fedora/core/8/\
i386/os/Fedora/RPMS
```

- 4** Get a copy of updates for Fedora 8 which can be found on the Fedora Download Server (at <http://rhold.fedoraproject.org/Download/updates.html/>) or your friendly neighbourhood mirror, and put it to the `download/fedora/core/updates/8/i386` directory on the Hardware Node. You can simply grab the updated RPMs with your favorite download tool and place them in the updates folder.

- 5** Install the `createrepo` package on the Hardware Node:

```
# rpm -Uhv createrepo-0.4.3-1.2.el4.rf.noarch.rpm
Preparing... ##### [100%]
1:createrepo ##### [100%]
```

- 6** Change to the `/vz/root/101/var/www/html` directory and create the following metadata repositories:

- For the Fedora 8 base RPM packages:

```
# createrepo download/fedora/core/8/i386/os
```

- For the updated versions of the Fedora 8 RPM packages:

```
# createrepo download/fedora/core/updates/8/i386
```

Creating the package metadata repository may take some time depending on the speed of your processor and hard disk drive.

- 7** Create a directory for storing mirror site lists. In our case, we will keep them in the `/vz/root/101/var/www/html/download/mirrors` directory:

```
# mkdir -p download/mirrors
```

- 8** Create the mirror list files and set the path to your local repository. For example, you can do it in the following way:

- For the Fedora 8 base RPM packages:

```
# echo 'http://123.145.145.123/download/fedora/core/8/i386/os/' >\
> download/mirrors/fedora-core-8
```

- For the updated versions of the Fedora 8 RPM packages:

```
# echo 'http://123.145.145.123/download/fedora/core/updates/8/i386' >\
download/mirrors/updates-released-fc8
```

The aforementioned commands create the `fedora-core-8` and `updates-released-fc8` files in the `/vz/root/101/var/www/html/download/mirrors` directory on the Hardware Node and add the `http://123.145.145.123/download/fedora/core/8/i386/os/Fedora/RPMS` and `http://123.145.145.123/download/fedora/core/updates/8/i386` strings to them, respectively.

- 9** Open the `/etc/vztt/vztt.conf` file on the Hardware Node for editing (e.g. by using `vi`) and change the value of the `FC_SERVER` variable as follows:

```
FC_SERVER=http://123.145.145.123
```

10 Grant the `apache` user and the `apache` group access to the created repositories inside Container 101 by executing the following command on the Hardware Node:

```
# vzctl exec 101 chown -R apache.apache /var/www/html/download
```

So, our local repository is created. From now on, the `vzpkg` tool will obtain RPM packages for the Fedora 8 EZ template and their updates from your local repositories inside Container 101. You can connect to these repositories through the `http` protocol from both remote and local Hardware Nodes. However, you can speed up the process of managing RPM files in your repository (e.g. update EZ templates and RPM packages) for those Containers that reside on your local Hardware Node (i.e. the Node where the repositories are stored). This can be done by specifying the `file` protocol to be used instead of the `http` one to connect to your created repositories:

1 Open the `/vz/template/fedora-core/8/x86/config/os/default/mirrorlist` file on the Hardware Node and comment the strings containing `FC_SERVER`:

```
#$FC_SERVER/download/mirrors/fedora-core-8
#$FC_SERVER/download/mirrors/updates-released-fc8
```

2 Execute the following commands to create the repository files:

- To create the `/vz/template/fedora-core/8/x86/config/os/default/repositories` file on the Node and to make it point to the Fedora 8 base RPM packages from your local repository:

```
# echo 'file:///vz/root/101/var/www/html/download/fedora/core/8/i386/os/Fedora/RPMS' > /vz/template/fedora-core/8/x86/config/os/default/repositories
```

- To create the `/vz/template/fedora-core/8/x86/config/os/default/repositories` file on the Node and to make it point to the updated versions of the Fedora 8 RPM packages from your local repository:

```
# echo file:///vz/root/101/var/www/html/download/fedora/core/updates/8/i386" >> /vz/template/fedora-core/8/x86/config/os/default/repositories
```

Managing Repositories for Commercial Linux Distributions

If you are going to run a commercial version of the Linux distribution (e.g. RHEL 5 or SLES 10) inside your Containers, you should create a special repository which will store the software packages for the corresponding distribution and enable you to update the existing packages inside your Containers.

In the example below, we will create the repository which will store the RPM packages included in the Red Hat Enterprise Linux 4 distribution. Besides, we will consider the situation explaining to you how to keep your repository up-to-date by getting the updated packages from the Red Hat Enterprise Linux 4 web site. In our example, we presume the following:

- The Hardware Node where the repository will be located is running Red Hat Enterprise Linux 4 (RHEL 4).
- The package repository will be stored inside Container 111.
- Container 111 can be accessed from other networks.
- The apache web server is running inside Container 111 and the default document root for apache is `/var/www/html`, i.e. the apache web server stores its sites in the `/vz/root/111/var/www/html` directory on the Hardware Node.
- The apache user and group inside Container 111 are `apache`.
- The `http` protocol will be used to access the RHEL 4 packages repository.

To create a repository for RHEL 4, you should perform the following operations:

- 1 Install the `redhat-as4-x86` OS EZ template on the Hardware Node, if it is not yet installed:

```
# vzpkg list
fedora-core-8-x86
fedora-core-5-x86
# vzpkg install template redhat-as4-x86-tmpl-4.0.0-1.swsoft.noarch.rpm
Preparing... ##### [100%]
 1:redhat-as4-x86-tmpl ##### [100%]
# vzpkg list
fedora-core-8-x86
fedora-core-5-x86
redhat-as4-x86
```

- 2 Create the Container where the repository storing the RHEL 4 packages will be located and assign an IP address and hostname to it. Let us use the `fedora-core-8-x86` OS EZ template to base your Container on. For example, to create Container 111 having the IP address of `144.134.134.144` and the hostname of `my_repo` for housing the repository, you can execute the following commands:

```
# vzpkg list
fedora-core-8-x86
fedora-core-5-x86
redhat-as4-x86
# ls /vz/template/cache
fedora-core-8-x86.tar.gz
fedora-core-5-x86.tar.gz
# vzctl create 111 --ostemplate fedora-core-8-x86 \
--ipadd 144.134.134.144 --hostname my_repo
Creating Container private area (fedora-core-8-x86)
Container is mounted
Postcreate action done
Container is unmounted
Container private area was created
```

```
Delete port redirection
Adding port redirection to Container(1): 4643 8443
```

3 Make sure that Container 111 is running and the `httpd` service is started inside the Container:

```
# vzlist -a
CTID      NPROC STATUS  IP_ADDR      HOSTNAME
  1         42  running 10.163.163.1 localhost
 111        -   stopped 144.134.134.144 my_repo
...
# vzctl start 111
Starting Container ...
Container is mounted
...
Container start in progress...
# vzctl exec 111 service httpd status
httpd is running...
```

Container 111 should be running to be able to perform the commands listed below.

4 Inside Container 111, create a directory where the RPM packages for Red Hat Enterprise Linux 4 will be stored:

```
# mkdir -p /vz/root/111/var/www/html/download/redhat/as4/i386/os/ \
RedHat/RPMS
```

5 Copy the RPM packages from the RHEL 4 distribution disks to the `/vz/root/111/var/www/html/download/redhat/as4/i386/os/RedHat/RPMS` directory by executing the following command for each of the RHEL 4 CDs:

```
# cp /media/cdrom/RedHat/RPMS*.rpm /vz/root/111/var/www/html/ \
download/redhat/as4/i386/os/RedHat/RPMS
```

6 Install the `createrepo` package on the Hardware Node:

```
# rpm -Uhv createrepo-0.4.3-1.2.el4.rf.noarch.rpm
Preparing... ##### [100%]
1:createrepo ##### [100%]
```

7 Create the metadata repository for the RHEL 4 packages with the `createrepo` utility:

```
# createrepo /vz/root/111/var/www/html/download/redhat/as4/ \
i386/os/RedHat/RPMS
```

Creating the RPM metadata repository may take some time depending on the speed of your processors and hard disk drive.

8 Create a directory for keeping mirror site lists. In our case, mirror site lists will be stored in the `/vz/root/111/var/www/html/download/mirrors` directory on the Hardware Node:

```
# mkdir -p /vz/root/111/var/www/html/download/mirrors
```

9 Create the `/vz/root/111/var/www/html/download/mirrors/redhat-as4` mirror list file and make it point to the repository inside Container 111 where RPM packages for RHEL 4 are stored:

```
# echo "http://144.134.134.144/download/redhat/as4/i386/os/RedHat/ \
RPMS/" > /vz/root/111/var/www/html/download/mirrors/redhat-as4
```

This command makes the `/vz/root/111/var/www/html/download/mirrors/redhat-as4` file on the Node and adds the `http://144.134.134.144/download/redhat/as4/i386/os/RedHat/RPMS` string to the file.

So, we have just created a repository for your RHEL 4 OS EZ template. Now you can cache the `redhat-as4-x86` EZ template and start creating Containers on its basis. However, if you wish to receive package updates from the RHEL 4 web site, you should additionally perform the following steps:

- 1 On the Hardware Node, run the `up2date` utility and register your `up2date` account with RHEL 4. Please consult the `up2date` documentation to complete this task.

Note: If your Hardware Node has another Linux OS installed on it (e.g. Fedora 8), you should create a special Container which is to run Red Hat Enterprise Linux 4 and register the `up2date` account from inside this Container.

- 2 Inside Container 111, create a directory where the updated versions of the RHEL 4 packages will be stored:

```
# mkdir -p /vz/root/111/var/www/html/download/redhat/updates/as4/i386
```

- 3 Create the `/vz/root/111/var/www/html/download/mirrors/updates-released-as4` mirror list file and make it point to the repository inside Container 111 where the updated versions of the RHEL 4 packages are stored:

```
# echo "http://144.134.134.144/download/redhat/updates/as4/i386/" > \
  /vz/root/111/var/www/html/download/mirrors/updates-released-as4
```

This command makes the `/vz/root/111/var/www/html/download/mirrors/updates-released-as4` file on the Node and adds the `http://144.134.134.144/download/redhat/updates/as4/i386` string to the file.

- 4 On the Hardware Node, create an empty RPM database. For example:

```
# mkdir -p /var/repo/redhat-as4
# rpm --initdb --dbpath /var/repo/redhat-as4
# rpm --dbpath /var/repo/redhat-as4 --import /usr/share/ \
  rhn/RPM-GPG-KEY
```

- 5 Install the RPMs from the official RHEL 4 disks in the created database:

```
# find /vz/root/111/var/www/html/download/redhat/as4/i386/os/ \
  RedHat/RPMS -name '*.rpm' | xargs rpm -ihv --justdb \
  --dbpath /var/repo/redhat-as4 --ignoresize --force --nodeps
```

Installing all RPM packages for the RHEL 4 distribution may take a rather long run; please wait until the installation process completes. After that, you can start using the `up2date` utility to update the created repository.

For example, the following session updates the RPM packages in your local repository inside Container 111:

- Obtain a list of RPM packages

```
# up2date -l --tmpdir=/tmp/up2date --dbpath /var/repo/redhat-as4/ | \
  awk 'BEGIN { stage = 0; } \
  stage == 0 && /^--*$/ { stage = 1; next; } \
  stage == 1 && /^$/ { stage = 2; next } \
  stage == 1 { print $1; }' \
  > /tmp/pkgs-list
```

- Download them:

```
# cat /tmp/pkgs-list | xargs up2date -d --tmpdir=/tmp/up2date \
--dbpath /var/repo/redhat-as4/
# rpm -ivh --justdb --dbpath /var/repo/redhat-as4/ --ignoresize \
--force --nodeps /tmp/up2date/*.rpm
# mv /tmp/up2date/*.rpm /vz/root/111/var/www/html/download/redhat/ \
updates/as4/i386/
# createrepo /vz/root/111/var/www/html/download/redhat/updates \
/as4/i386
```

You can also make a script to automatically perform the aforementioned operations and set this script to be run as a cron job.

Creating Proxy Server for EZ Templates

Along with setting up the package repositories described in the previous subsections, you can create special caching proxy servers allowing to efficiently manage your OS and application EZ templates.

Setting Up Proxy Server for EZ Templates

Virtuozzo Containers 4.0 allows you to set up special caching proxy servers and use them to efficiently manage your OS and application EZ templates. The following picture demonstrates an example of the Virtuozzo network containing two Hardware Nodes and a separate proxy server:

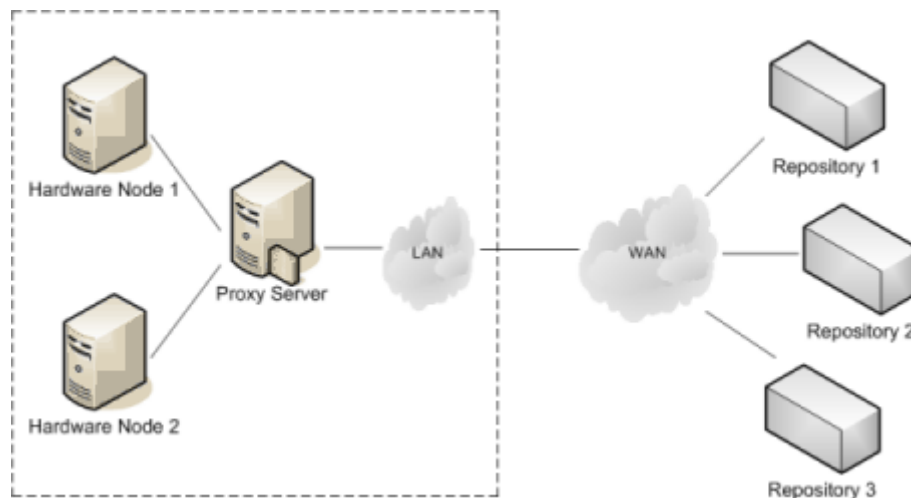


Figure 2: Virtuozzo Network With Caching Proxy Server

The caching proxy server sits between the Hardware Nodes (*Hardware Node 1* and *Hardware Node 2*) keeping a number of EZ templates and the repositories (*Repository 1*, *Repository 2*, and *Repository 3*) storing the packages for these EZ templates. When either Hardware Node requests certain packages from any repository for the first time, these packages are first downloaded to the proxy server where they are cached and then returned to the corresponding Node. From this moment on, if a Hardware Node requests a package already available in the cache on the proxy server and there are no updates for this package in the official repositories, the package will be immediately downloaded to the Node without requesting the repositories on the Internet for this package. At the same time, if one or several updates are available for the requested package in the official repositories, the procedure of handling these updates is identical to that of the main packages, i.e. the found updates are downloaded to the proxy server, cached there, and then retransmitted to the corresponding Hardware Node.

Besides, a special `Virtuozzo` script is automatically launched on the proxy server at stated intervals. During its execution, the script creates a local repository on the basis of the cached packages (please note that the script does not remove the cached packages; so, they remain intact until you delete them manually). The packages in the local repository are organized in the same way as in the original repository. So, you can make the `vzpkg` tool use the packages from the created local repository on the proxy server, which may be useful, for example, for migrating a Container that is based on an OS EZ template containing one or more outdated packages. In this case the Container migration may fail if the following conditions are simultaneously met:

- the Destination Node does not have any of the outdated software packages included in the OS EZ template the Container to be migrated is based on and
- the missing packages are not available in the official repositories set to handle the OS EZ template; so, they cannot be downloaded and installed on the Destination Node. The absence of certain software packages in the official repositories may come out of the fact that these packages have become obsolete in the course of time and have been replaced with newer versions.

You can easily avoid the aforementioned problem by setting up a caching proxy server that will store both the outdated and newer versions of the necessary packages and using this proxy server for handling your OS and application EZ templates.

In general, setting up a proxy server for managing EZ templates has the following advantages:

- Your Internet bandwidth consumption is greatly reduced since all packages are downloaded to the proxy server only once and can then be used by any Hardware Node on your network.
- You can more rapidly apply software updates to your Containers since the proxy server where the downloaded packages are stored resides in the local network.
- You can always have the software packages included in your OS EZ templates at hand and do not have to worry whether they have been changed in or removed from their original repositories.
- The local repository created by the `Virtuozzo` script on the proxy server enables you to effectively manage your EZ templates even if some packages are not any more available in the corresponding official repository or the server keeping the original repository went down for some reason or other.

`Virtuozzo Containers 4.0` provides you with the `vzpkgproxy` utility allowing to automatically set up any computer on your network to serve as a caching proxy server for your EZ templates. During its installation, the utility performs all the tasks necessary to install, configure, and put into operation your proxy server. `vzpkgproxy` is located in the `/virtuozzo/RPMS` directory of your `Virtuozzo Containers` distribution and can be installed with the `rpm -i` command on any servers (including `Virtuozzo Containers`) meeting the following requirements:

- the Apache `httpd` server, version 2.0.53 and higher, should be installed on the server;
- the `createrepo` package, version 0.4.2 and higher, should be installed on the server.

Please keep in mind that you may also need to install a number of additional packages to satisfy the `vzpkgproxy` dependencies (e.g. the `libstdc++.so.5` package).

Note: If you are going to have one or more Containers running Red Hat Enterprise Server 4 or 5, you should use the `vzrhnp` utility to create special RHN Proxy Servers enabling you to effectively manage the packages included in the RHEL 4 and 5 OS EZ templates. Detailed information on this utility is provided in the next subsection.

By default, the proxy server is configured to cache the packages from all external hosts. However, you can modify the `CACHE_DISABLE` parameter in the `/etc/vzpkgproxy/vzpkgproxy.conf` file on the proxy server to explicitly specify the hosts to be excluded from the caching process. You can also configure the behavior of the proxy server by performing the following operations:

- Modifying the port number on which the `httpd` daemon running on your proxy server is listening by setting the desired port number in the `/etc/httpd/conf.d/vzproxy.conf` file on the proxy server; the default port number is 8080.
- Editing the `REPO_DIR` parameter in the `vzpkgproxy.conf` file to change the path to the directory where the local repository created on the basis of the cached packages will be stored. By default, this directory has the path of `/var/www/html/download`.
- Making the `vzpkg` tool use the packages from the local repository on the proxy server while handling your EZ templates. To this effect, you should edit the `/etc/vztt/vztt.conf` file on the Hardware Node and specify:
 - the URL of the proxy server, the port number where the `httpd` daemon is listening, and the path to the directory where the EZ templates local repository is located as the value of the `VZTT_PROXY` parameter. For example, if your proxy server has the `127.123.123.127` IP address assigned, the `httpd` daemon running on the proxy server is listening on port 8080, and the local repository is stored in `/var/www/html/download`, you should set the `VZTT_PROXY` parameter to `http://127.123.123.127:8080/download`.
 - the URL of the proxy server and the port number where the `httpd` daemon is listening as the value of the `HTTP_PROXY` parameter. For example, you should set this value for the aforementioned proxy server to `http://127.123.123.127:8080`. Please keep in mind that you also need to set the `HTTP_PROXY_PASSWORD` and `HTTP_PROXY_USER` parameters in the `/etc/vztt/vztt.conf` file if the access to your proxy server is password-protected.

Setting Up RHN Proxy Server for RHEL OS EZ Templates

If some of your Containers are to run the Red Hat Enterprise Linux 4 (RHEL 4) or 5 (RHEL 5) distribution, you may wish to create a special caching proxy server - *RHN (Red Hat Network) Proxy Server* - allowing for faster RHEL packages downloads, easier distribution, and lower bandwidth requirements. RHN Proxy Servers can be created using the `vzrhnproxy` utility shipped with Virtuozzo Containers 4.0. This utility can be installed on any computer (including Virtuozzo Containers) running the RHEL 4 and RHEL 5 Linux distributions with the `rpm -i` command.

Notes: 1. You may need to install a number of additional packages to satisfy the `vzrhnproxy` dependencies.

2. You can also try to deploy an RHN Proxy Server on systems running other RHEL-based distributions (e.g. CentOS 5 or Fedora 8); however, `vzrhnproxy` has not been extensively tested with them.

Let us assume that you wish to create an RHN Proxy Server on the server with the IP address of `192.168.10.10` that will serve all Containers running the 32-bit version of RHEL 5 and residing on the Hardware Node with the hostname of `mycomputer1` and the IP address of `192.168.0.125`. To this effect, you should perform the following operations:

- 1 Log in to the server where you are planning to create the RHN Proxy Server (further referred to as Proxy Server) and make sure the `vzrhnproxy` utility is installed on this server.
- 2 Specify a valid user name and password you use to log in to Red Hat Network (RHN) as the values of the `REDHAT_LOGIN` and `REDHAT_PASSWORD` parameters, respectively, in the `/etc/vz/pkgproxy/rhn.conf` file on the Proxy Server. These credentials will be used by `vzrhnproxy` on the next step to register your system profile with RHN. For example:

```
# vi /etc/vz/pkgproxy/rhn.conf
REDHAT_LOGIN="user1"
REDHAT_PASSWORD="2WSX00KM"
...
```

- 3 Execute the following command on the Proxy Server:

```
# vzrhnproxy register i386 5Server mycomputer1 192.168.0.125
registering for i386-5Server-mycomputer1
...
```

where `i386` and `5Server` denote the system architecture and the operating system you wish to register with RHN (in our case, we are registering the 32-bit version of the Red Hat Enterprise Linux 5 server).

During the command execution, `vzrhnproxy` will:

- connect to Red Hat Network (available at <http://rhn.redhat.com>) with the credentials specified in the `rhn.conf` file on the previous step;
- create a profile and register it with RHN for the system running the 32-bit version of RHEL 5;
- download the headers of the packages comprising the 32-bit RHEL 5 distribution to the Proxy Server;
- create a pseudo-repository containing the repodata generated on the basis of the downloaded headers;

- grant the server with the IP address of 192.168.0.125 (i.e. our Hardware Node) access to the Proxy Server.

4 On the Hardware Node:

- open the `/etc/vztt/vztt.conf` file for editing (e.g. using `vi`) and change the value of the `RH_SERVER` parameter as follows:

```
RH_SERVER=http://192.168.10.10/rhn
```

- save the file.

From this moment on:

- If the Hardware Node with the IP address of 192.168.0.125 requests certain packages included in the RHEL 5 distribution for the first time (e.g. while caching the RHEL 5 OS EZ template), this request will be sent to the Proxy Server which, in its turn, will connect to Red Hat Network and retrieve the requested packages. These packages will then be downloaded to the Proxy Server where they are cached and finally returned to the Hardware Node.
- If the Hardware Node requests a package already available in the cache on the Proxy Server, the package will be immediately downloaded from the cache to the Node.

You can make the Proxy Server serve the requests for RHEL 5 packages from more than one Node. To this effect, you should specify the IP addresses of the corresponding Hardware Nodes during the `vzrhnproxy register` command execution and properly edit the `/etc/vztt/vztt.conf` files on each of these Nodes (please see **Step 3** and **4** above). Please keep in mind that, while executing the `vzrhnproxy register` command, you should specify the hostname of one Hardware Node only; this can be the hostname of any Node to be handled by the Proxy Server.

You can also create and register several system profiles with Red Hat Network. For example, if you have one or more Hardware Nodes hosting Containers with the x86-64-bit version of RHEL 5, you may wish to use the Proxy Server for handling the packages included in this RHEL 5 version as well. To this effect, you should perform once more **Steps 1-4** described above and use the following command on **Step 3** to register a new system profile with RHN:

```
# vzrhnproxy register x86_64 5Server mycomputer2 192.168.22.22
registering for x86_64-5Server-mycomputer2
...
```

where 192.168.22.22 is the IP address of the Hardware Node hosting 64-bit Containers.

To list all system profiles registered with RHN, you can execute the following command on the Proxy Server:

```
# vzrhnproxy list
i386-5Server-mycomputer1
x86_64-5Server-mycomputer2
x86_64-5Server-mycomputer3
```

As you can see, three system profiles are currently registered with RHN: two for servers running the x86-64-bit version of RHEL 5 and one for the server running the 32-bit version of RHEL 5. For each of these profiles, the corresponding pseudo-repository containing the RHEL 5 package repodata exists on the Proxy Server.

After a lapse of time, the repodata (and, consequently, the cache on the Proxy Server) may become obsolete. In this case you can use the `vzrhnproxy update` command to update the repodata in pseudo-repositories on the Proxy Server. For example, the following command will update the repodata in the pseudo-repository corresponding to the `i386-5Server-mycomputer1` profile:

```
# vzrhnproxy update i386-5Server-mycomputer2
```

Preparing OS EZ Template for Container Creation

OS EZ templates are used to create Containers on their basis. To prepare an OS EZ template for the Container creation, you should:

- 1 Install the OS EZ template on the Hardware Node and
- 2 Cache the installed OS EZ template.

To install a new OS EZ template on the Hardware Node, you should use the `vzpkg install template` command. For example, to install the Red Hat Enterprise Linux 5 OS EZ template on the Node running the 32-bit version of Virtuozzo Containers:

```
# vzpkg install template redhat-el5-x86-ez-4.0.0-4.swsoft.noarch.rpm
Preparing... ##### [100%]
 1:redhat-el5-x86 ##### [100%]
# vzpkg list
redhat-el5-x86
```

As you see, the `redhat-el5-x86` EZ template is now installed on the Hardware Node. The corresponding path is `/vz/template/redhat/el5`. However, before the `redhat-el5-x86` EZ template can be used as a basis for the Container creation, it should first be cached. This can be done by using the `vzpkg create cache` command:

Note: Before you can start caching your OS EZ templates, you may need to set up a package repository for them. So, you have to build a special repository for all commercial versions of the Linux distributions (e.g. Red Hat Linux Enterprise 4 or 5). Detailed information on how to manage package repositories is provided in the [Setting Up Repository for EZ Templates](#) section (p. 26).

```
# vzpkg create cache redhat-el5-x86
...
Complete!
Packing cache file redhat-el5-x86.tar.gz ...
Cache file redhat-el5-x86.tar.gz [14M] created.
```

The created tar archive is put to the `/vz/template` directory on the Hardware Node:

```
# ls /vz/template/cache
redhat-el5-x86.tar.gz
```

After the `redhat-el5-x86` EZ template has been successfully cached, you can start creating Container on its basis. Detailed information on how to create Containers on the basis of OS EZ templates is provided in the [Creating New Container](#) section of the [Parallels Virtuozzo Containers User's Guide](#).

Installing Application EZ Templates on Hardware Node

The same way as you use an OS EZ template on the Virtuozzo system in order to create any number of Containers on its basis and share its resources, you can use application EZ templates in Virtuozzo Containers 4.0 in order to share package files among any number of Containers. You can then add these applications to any number of Containers.

To install a new application EZ template on the Hardware Node, you should use the `vzpkg install template` command. For example, to install the `mysql` EZ template intended to be run on the Red Hat Enterprise Linux 5 distribution, you can issue the following command:

Note: If you are running one of the rpm-based Linux distributions (e.g. Red Hat Enterprise Linux 5 or Fedora 8), you can also use the `rpm -Uhv` command to install application EZ templates on your Hardware Node.

```
# vzpkg install template mysql-redhat-el5-x86-ez-3.0.0-4.swsoft.rpm
Preparing...                               ##### [100%]
 1:mysql-redhat-el5-x86                     ##### [100%]
```

The `mysql` EZ template will be installed in the `/vz/template/redhat/el5/x86/config/app/mysql` directory on the Hardware Node. To make sure that this EZ template has been successfully installed, you can use the `vzpkg list` command:

```
# vzpkg list
redhat-el5-x86                2007-05-21 02:22:45
redhat-el5-x86 mysql
```

As you see, the `mysql` EZ template is now available on the Node and can be added to any number of Containers.

Upload and Install New EZ Templates on Hardware Node in Management Console

In case you have one or more new EZ templates that you would like to upload and install on your Hardware Node(s), you should first have these templates accessible from the workstation where Parallels Management Console is installed (e.g. insert the CD-ROM with the template(s) into the computer CD-ROM drive or copy the templates to the computer hard disk) and then launch the Upload and Install New Virtuozzo Templates wizard.

To invoke the wizard, expand the **Templates** item under any Hardware Node registered in Parallels Management Console, right-click the **Templates** item, and select **Upload and Install New Virtuozzo Templates** on the context menu. The **Choose Virtuozzo Templates to Distribute** window opens:

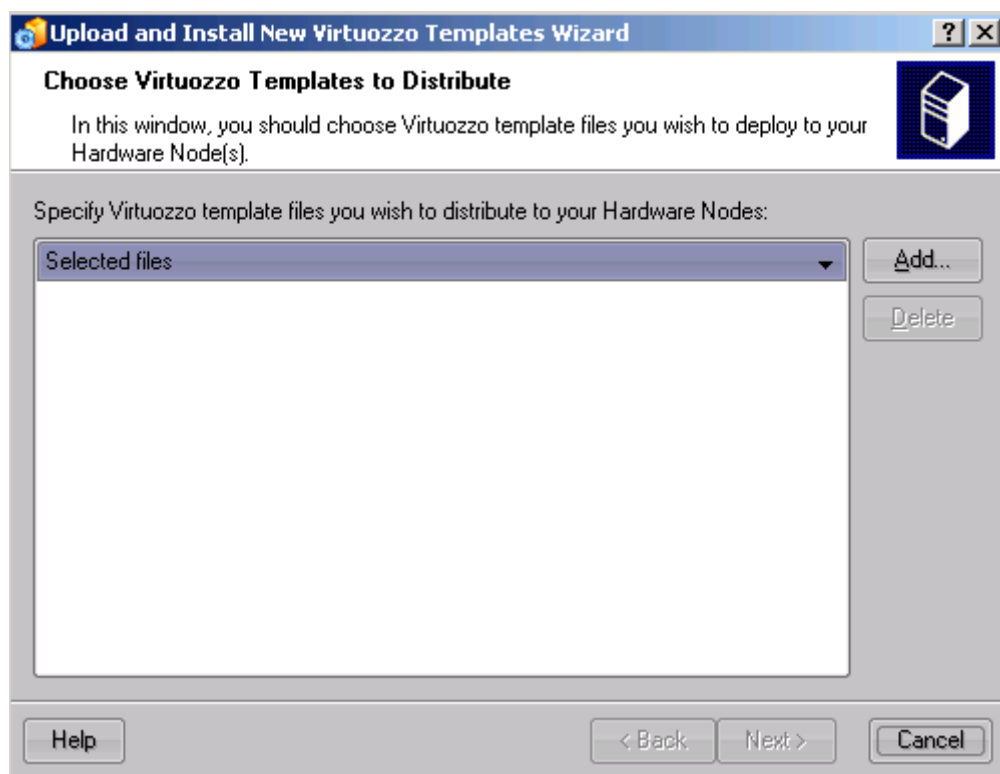


Figure 3: Management Console - Choosing Templates for Uploading

In this window you will be asked to choose EZ templates you wish to deploy to your Hardware Node(s). To this effect, click the **Add** button and provide the path to the Virtuozzo Template(s) you would like to install on the Node; then click **Next**.

The next screen will prompt you to define target Hardware Nodes where you wish to add new EZ templates.

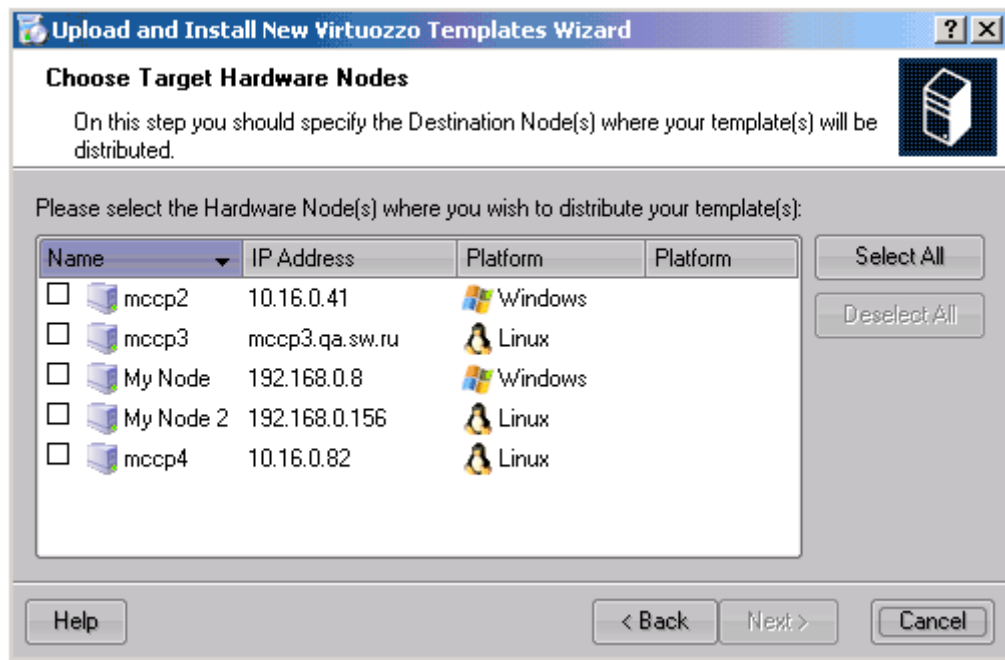


Figure 4: Management Console - Choosing Target Hardware Nodes

Parallels Management Console may upload and install the specified template(s) to any registered Hardware Node. Choose the Nodes where you wish the template(s) to be installed and click Next. In case of a large number of Nodes, it is reasonable to make use of the Select All or Deselect All buttons.

The last page of the wizard allows you to review Virtuozzo Templates deploying settings:

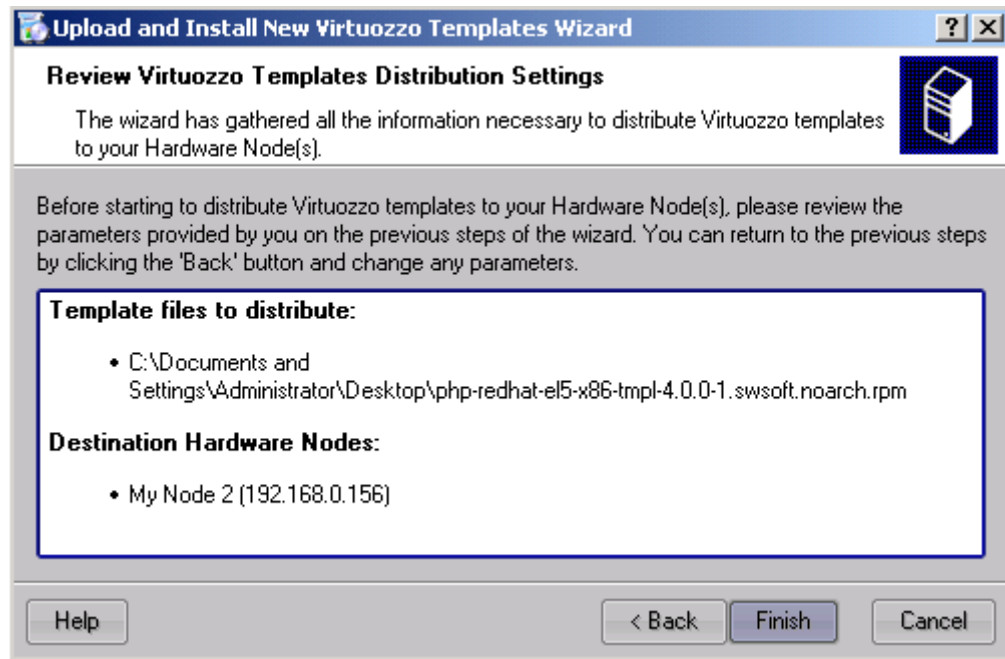


Figure 5: Management Console - Reviewing Templates Distribution Parameters

It summarizes the information provided by you on the previous steps, namely, what Virtuozzo Templates to install and where to install them. If you are satisfied with the settings, click the **Finish** button. This will launch the upload and installation process. Otherwise, click **Back** to make the necessary changes to the information entered.

Listing EZ Templates

The `vzpkg list` command allows you to list the EZ templates installed on the Hardware Node. They may be already used or not used by certain Containers:

```
# vzpkg list
redhat-el5-x86
fedora-core-8-x86      2008-01-24 15:45:15
```

As you see, the `redhat-el5-x86` and `fedora-core-8-x86` EZ templates are available on the Hardware Node. The characters opposite the `fedora-core-8-x86` EZ template informs you of the date and time when software packages included in the template were updated for the last time. In its turn, the characters absence beside the `redhat-el5-x86` EZ template indicates that the template has not yet been cached at all.

Specifying a Container number as the parameter, `vzpkg list` prints the EZ templates used by the specified Container:

```
# vzpkg list 101
fedora-core-8-x86      2008-01-27 14:27:19
```

In Parallels Management Console, it is sufficient to choose the **Templates** item under the corresponding Hardware Node name and select either the **OS Templates** or **Application Templates** tab to see a list of the OS or application templates installed on the Node, respectively. For example:

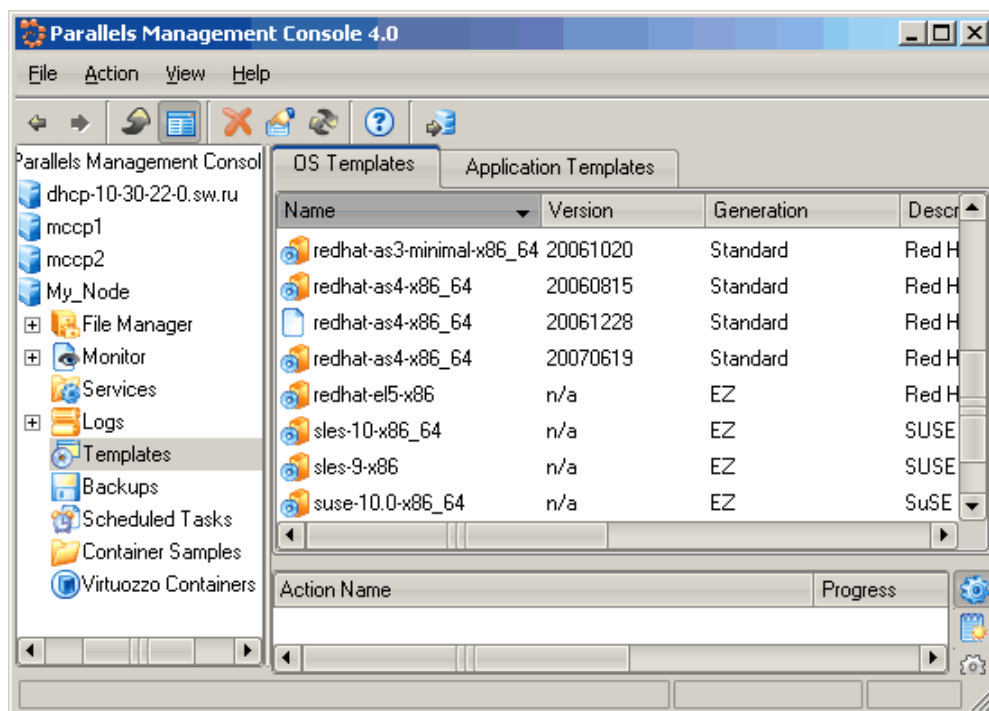


Figure 6: Management Console - Listing EZ Templates

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

Column Name	Descriptions
Name	The name of the EZ template.
Version	Valid for standard OS and application templates only.
Generation	The type of the template: 'standard' or 'EZ'. All EZ templates have the EZ legend in this column.
Description	The description of the EZ template.
Platform	The platform where the EZ template can be used.
Architecture	The system architecture where the EZ template can be used.
Cached	Denotes whether the EZ OS template has been cached or not. For EZ OS templates only.
OS	The name of the Linux distribution under which the EZ application template can be run. For EZ application templates only.

To see the EZ templates used by a particular Container, double-click the needed Container in the Container list in the right pane to open this Container management window, and then again choose the **Templates** item and click the **OS Templates** or **Application Templates** tab in the left pane of the new window.

Adding Application EZ Templates to Container

To add an application EZ template to an existing Container, you should use the `vzpkg install` command. To successfully add an application EZ template to a Container, this Container should be running; otherwise, it is impossible to run the installation process in the Container context.

In the example below, the `mysql` application EZ template meant for the usage with Red Hat Enterprise Linux 5 and already installed on the Hardware Node is added to Container 101:

```
# vzctl status 101
CTID 101 exists mounted running
#vzpkg list
redhat-el5-x86          2007-05-21 02:21:56
redhat-el5-x86  mysql
...
# vzpkg install 101 mysql
...
Installed:
mysql          i386      0:4.1.12-3.RHEL5.1
mysql-bench   i386      0:4.1.12-3.RHEL5.1
mysql-devel   i386      0:4.1.12-3.RHEL5.1
...
```

You can also add an application EZ template in Parallels Management Console by performing the following operations:

- 1 In the Management Console left pane, select the **Templates** item under the corresponding Hardware Node name and then click the **Application Templates** tab:

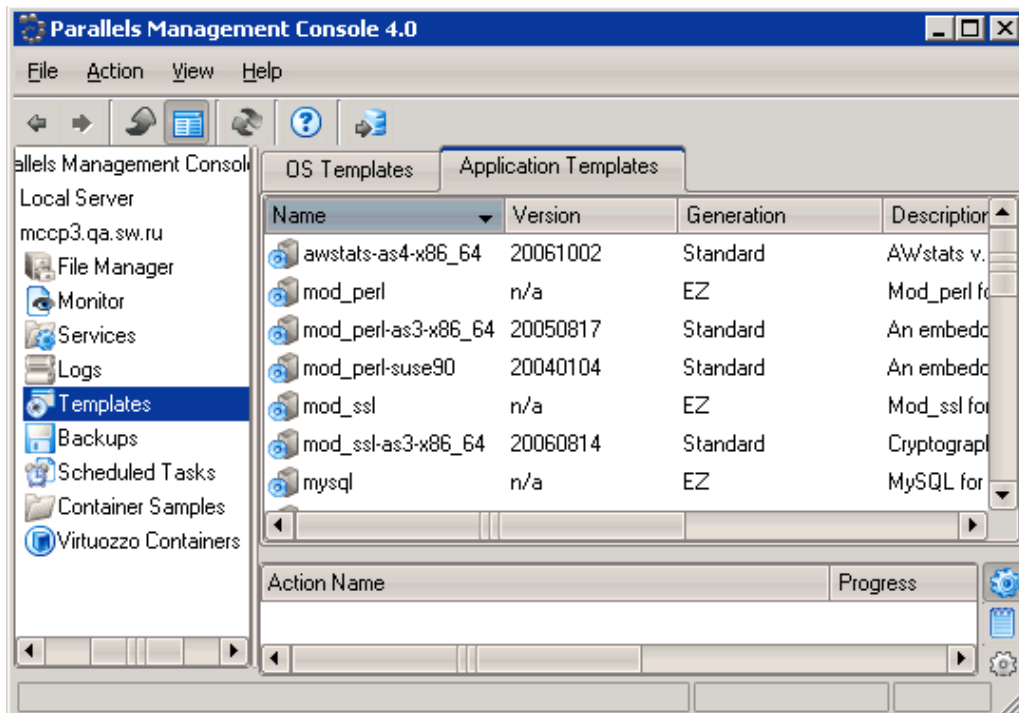


Figure 7: Management Console - Adding Application EZ Templates to Container

- 2 In the Management Console right pane, right-click the application EZ template you wish to add to your Container and select **Install Into Containers** on the context menu.
- 3 Select the check boxes of the Containers where the application EZ template is to be added and click **Next**. You can use the **Select All/Deselect All** buttons to select/deselect all Containers listed in the **Choose Target Containers** window.

Note: To facilitate working with your application EZ templates, you can select the check box at the bottom of the window to display only the Containers running the Linux distributions that are compatible with the given application EZ template.

- 4 On the next screen, you can select the **Force template installation** check box to force the template installation inside the Container. In this case no dependencies and no available versions of the application template will be checked during its installation, which may cause the installed application template to malfunction.
- 5 The last window allows you to review the information entered on the previous steps. If you are satisfied with the data entered, click **Finish** to start adding the application EZ template to the Container; otherwise, click **Back** and change the necessary parameters.

If you are adding an EZ template to only one Container, you can as well do the following:

- 1 Open a list of Containers in the Management Console main window by selecting the **Virtuozzo Containers** item in the Hardware Node tree.
- 2 Double-click the name of the Container where you want to add an EZ template.
- 3 Select the **Templates** item in the main tree of the opened Container Manager, click the **Application Templates** tab, right-click somewhere in the top part of the Management Console right pane, and select the **Add Virtuozzo Application Template** option on the context menu.
- 4 Follow the instructions of the wizard.

Alternatively, you can:

- 1 Open a list of Containers in the Management Console main window.
- 2 Right-click the Container where you wish to add the template and select **Templates --> Add/Update Virtuozzo Templates** on the context menu.
- 3 Follow the instructions of the wizard.

Keeping EZ Templates Up To Date

Virtuozzo Containers 4.0 allows you to update your OS and application EZ templates as follows:

- update any of the EZ templates installed on the Hardware Node;
- update the caches of OS EZ templates installed on the Hardware Node;
- update the packages that are included in the EZ templates (OS and application) applied to particular Containers.

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

Updating EZ Templates on Hardware Node

Sometimes, you may need to update one or more EZ templates (either OS or application) installed on your Hardware Node. The process of updating EZ templates consists in updating one or more EZ template configuration files located in the `/vz/template/<os_name>/<os_version>/<arch>/config` directory on the Node. Virtuozzo Containers 4.0 allows you to use one of the following tools to update the EZ templates installed on the Hardware Node:

- the `vzup2date` utility;
- the `vzpkg update template` utility;
- Parallels Management Console.

Updating Templates with `vzup2date`

The `vzup2date` utility allows you to update any of the EZ templates installed on the Hardware Node. This utility can also be used to download new EZ templates to the Hardware Node and install them there. `vzup2date` can be launched in two modes:

- graphical mode and
- command line mode.

Updating Virtuozzo EZ templates in the graphical mode takes place if you have executed the `vzup2date` utility with the `-z` option. After launching the utility, you will be presented with a greeting screen:

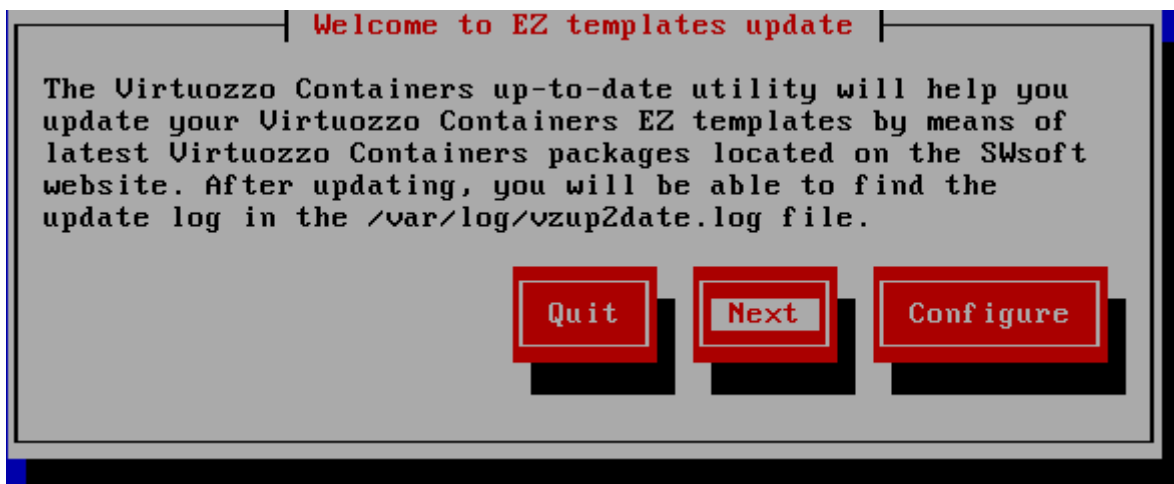


Figure 8: Updating Virtuozzo Containers - Welcome Screen

In this window you can do one of the following:

- click the **Next** button to connect to the repository storing the latest EZ templates (either the Parallels default repository or your own one);
- click the **Configure** button to configure the parameters used to connect to the EZ templates repository.

As soon as you press **Next** in the **Welcome...** window, the utility will try to connect to the EZ templates repository (either the Parallels default repository or your own one) and, if the connection is successful, display the **EZ Templates Selection** window listing all EZ templates that have one or more updates available or that are not installed on your Node at all. For example:



Figure 9: Updating Virtuozzo Containers - Selecting Linux Distribution

This window allows you do one of the following:

- If you wish to download and install all available EZ templates/template updates for a certain Linux distribution, select this distribution by placing the cursor beside it and pressing the space bar on your keyboard; then click **Next**.
- If you wish only certain EZ templates of the corresponding Linux distribution to be installed/updated on the Hardware Node, place the cursor beside this distribution and press **F2** on your keyboard. You will be presented with the **Templates selection** window where you can select the corresponding EZ templates:

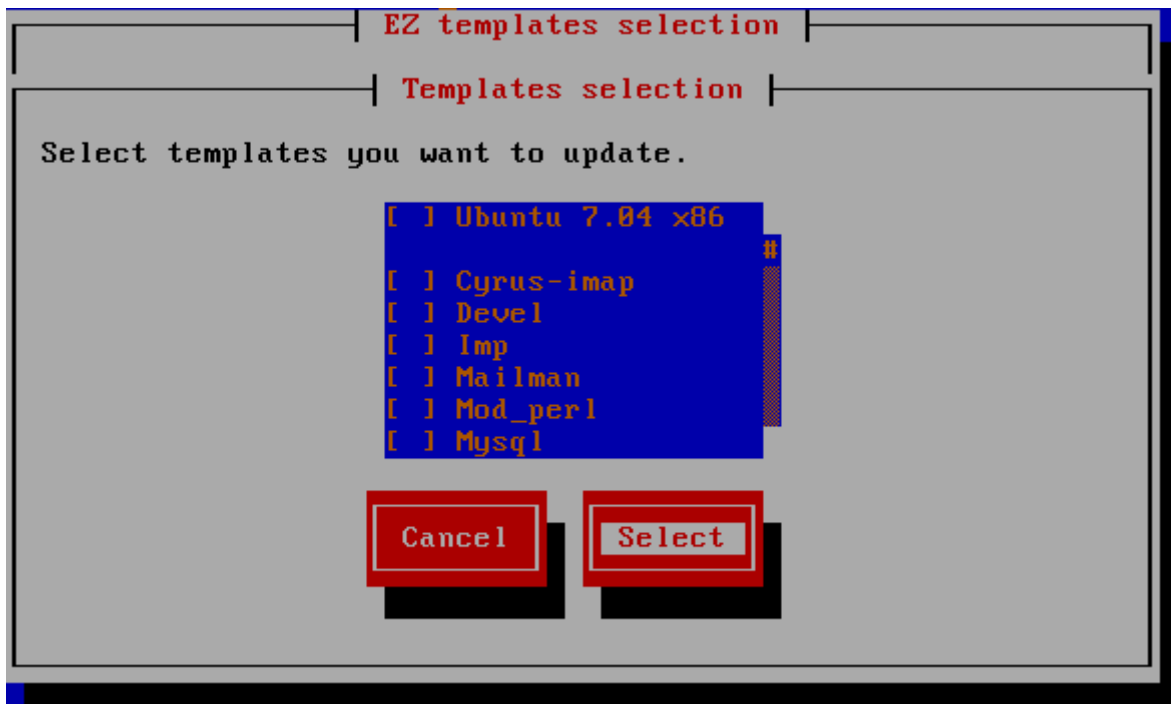


Figure 10: Updating Virtuozzo Containers - Selecting EZ Templates

After choosing the right EZ templates, click the **Select** button to close the displayed window and then click **Next** to proceed with the wizard.

Note: New application EZ templates for a Linux distribution can be installed on the Hardware Node only if the corresponding OS EZ template is already available on this Node.

On the next step, you can review the EZ templates/template updates you selected on the previous step and scheduled for downloading and installing on your Hardware Node. If you are not satisfied with the chosen templates/template updates, click the **Back** button to return to the previous step and modify the set of templates; otherwise, click **Next** to start downloading the templates/template updates to the Node.

After the EZ templates/template updates have been successfully downloaded to the Hardware Node, the Installing EZ template window is displayed:

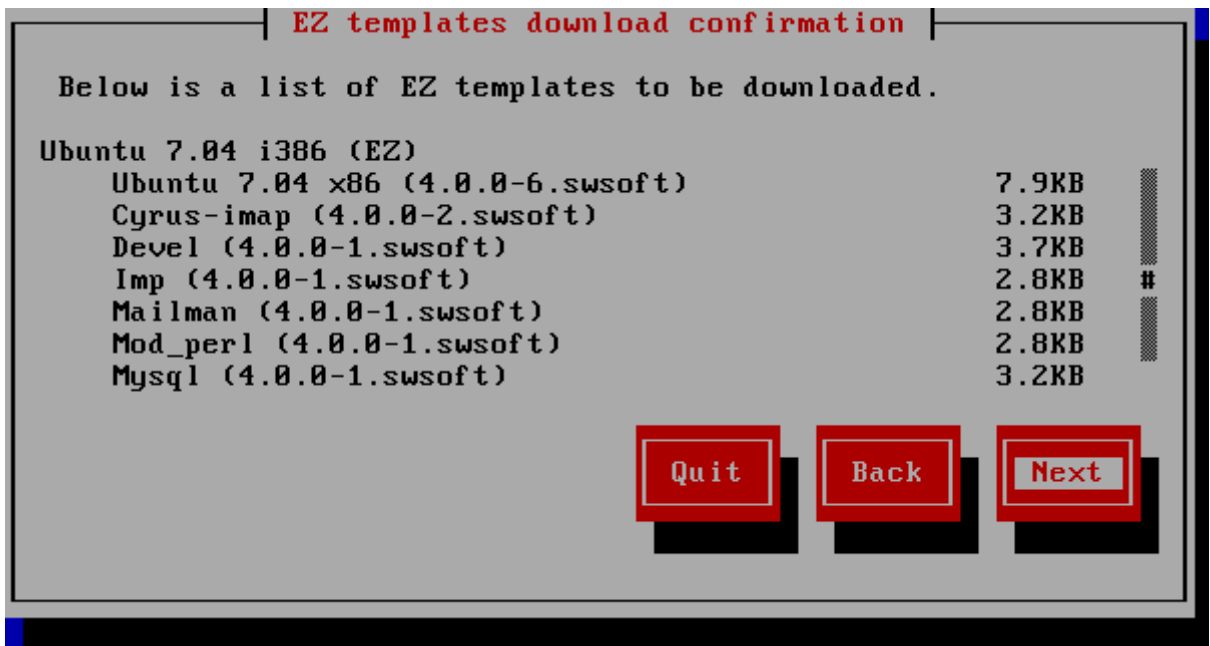


Figure 11: Updating Virtuozzo Containers - Viewing EZ Templates to Install

In this window you can view the templates/template updates ready to be installed on your Node. If you are installing a new OS EZ template/template update, you can make use of the Run vzpkg cache after installation check box to cache the corresponding OS EZ template/template update right after its installation on the Node. By default, all OS EZ templates are just installed on the Hardware Node without being cached; however, you can select the provided check box and schedule your OS EZ template/template update for caching. Clicking Next starts installing the selected EZ templates/template updates on the Hardware Node. By the time the wizard finishes you should have updated OS and application EZ templates on your system.

Another way of updating your EZ templates is to run the vzup2date utility in the command line mode, which can be done by passing the corresponding commands, switches, and options to vzup2date. For example, the following command will update the fedora-core-8-x86 OS EZ template to the latest version:

```
# vzup2date -z -m batch install fedora-core-8-x86
```

Detailed information on all options that can be passed to the vzup2date utility is given in the Virtuozzo Command Line Interface chapter of Parallels Virtuozzo Containers Reference Guide.

Updating Templates With `vzpkg update template`

Another way of updating your EZ templates installed on the Hardware Node is to use the `vzpkg update template` utility. This utility allows you to update OS or application EZ templates from the corresponding local RPM packages. For example, you can execute the following command to update the CentOS 5 OS EZ template installed on the Node from the `centos-5-x86-ez-4.0.0-2.swsoft.noarch.rpm` package located in the `/root` directory:

```
# vzpkg update template /root/centos-5-x86-ez-4.0.0-2.swsoft.noarch.rpm
```

You can update a number of EZ templates at once by specifying the corresponding packages and separating them by spaces. For example, the following command

```
# vzpkg update template /root/centos-5-x86-ez-4.0.0-2.swsoft.noarch.rpm  
/root/redhat-el5-x86-ez-4.0.0-4.swsoft.noarch.rpm
```

will simultaneously update the CentOS 5 and Red Hat 5 OS EZ templates installed on the Hardware Node.

Updating Templates in Parallels Management Console

Parallels Management Console provides you with the **Virtuozzo Templates Update** wizard allowing you to update any of EZ templates installed on your Hardware Node. You can also use this wizard to download new EZ templates to the Hardware Node and install them there. To invoke the **Virtuozzo Templates Update** wizard, right-click the **Templates** item under the corresponding Hardware Node name and select **Check for Template Updates** on the context menu. When launched, the wizard tries to connect to the EZ templates repository (either the Parallels default repository or your own one) and, if the connection is successful, display the **Select Updates** window listing all EZ templates that have one or more updates available or that are not installed on your Node at all. For example:

Note: If the connection to the Virtuozzo update server cannot be established, you will be presented with the **Repository Update Settings** window where you will be asked to provide the correct information to connect to the update server. Detailed information on how to change the parameters in this window is given in the **Checking Virtuozzo Update Server Settings** subsection of the **Parallels Virtuozzo Containers User's Guide**.

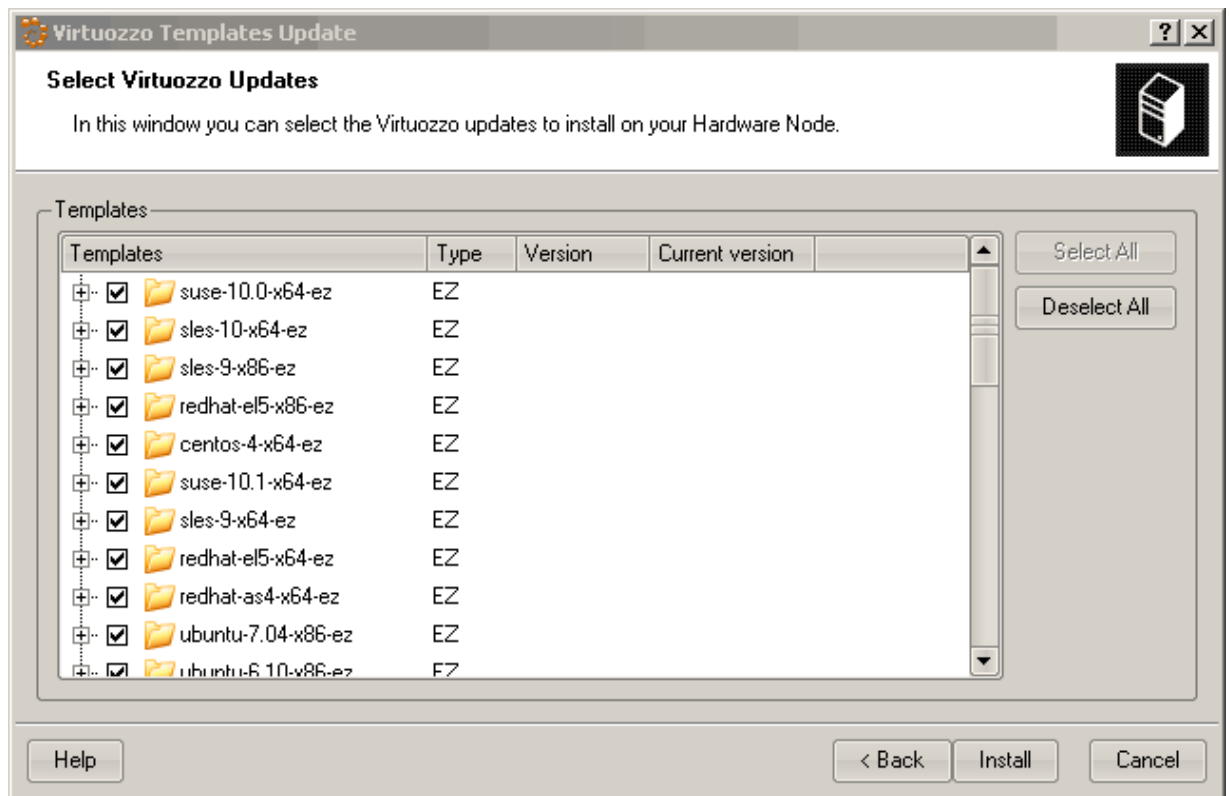


Figure 12: Management Console - Updating EZ Templates

In this window you can do one of the following:

- If you wish to download and install all available EZ templates/template updates for a certain Linux distribution, click the **Next** button to go to the next step of the wizard.
- If you wish only certain EZ templates of a Linux distribution to be installed/updated on the Hardware Node, click on the plus sign beside the corresponding Linux distribution to display a list of application templates available for this distribution. You can then get detailed information about a particular template by selecting the corresponding template and viewing its data in the right part of the displayed window. By default, all new templates/template updates are set for downloading to and installing on the Hardware Node. To prevent this or that EZ template from being downloaded/installed, just clear its check box. When you are ready, click **Next**.

Click **Finish** to start installing the selected EZ templates/EZ template updates on the Hardware Node.

Updating OS EZ Template Caches

With the release of new updates for the corresponding Linux distribution, the created OS EZ template cache can become obsolete. So, Virtuozzo Containers 4.0 provides the `vzpkg update cache` command allowing you to quickly update any of the OS EZ template caches available on the Hardware Node.

Note: If you are going to update the cache of a commercial OS EZ template (e.g. Red Hat Enterprise Server 5 or SLES 10), you should first update software packages in the remote repository used to handle this OS EZ template and then proceed with updating the EZ template cache. Detailed information on how to manage repositories for commercial Linux distributions is provided in the [Setting Up Repositories and Proxy Servers for EZ Templates](#) section (p. 26).

When executed, `vzpkg update cache` checks the cache directory in the template area (by default, the template area is located in `/vz/template`) on the Hardware Node and updates all existing tarballs in this directory. However, you can explicitly indicate the tarball for what OS EZ template should be updated by specifying the OS EZ template name. For example, to update the tarball for the `fedora-core-8-x86` OS EZ template, you should issue the following command:

```
# vzpkg update cache fedora-core-8-x86
Loading "rpm2vzrpm" plugin
Setting up Update Process
Setting up repositories
base0          100% |=====| 951 B    00:00
base1          100% |=====| 951 B    00:00
base2          100% |=====| 951 B    00:00
base3          100% |=====| 951 B    00:00
...
```

Upon the `vzpkg update cache` execution, the old tarball is renamed by receiving the `-old` suffix (e.g. `fedora-core-8-x86.tar.gz-old`):

```
# ls /vz/template/cache
fedora-core-8-x86.tar.gz  fedora-core-8-x86.tar.gz-old
```

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

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

To update an OS EZ template cache in Parallels Management Console, you should:

- 1 Select the **Templates** item under the corresponding Hardware Node name in the Management Console left tree.
- 2 In the Management Console right pane, click the **OS Templates** tab to display a list of OS templates installed on the Node.
- 3 Right-click the template you wish to cache in the right pane and select **Cache OS Template** on the context menu. For example:

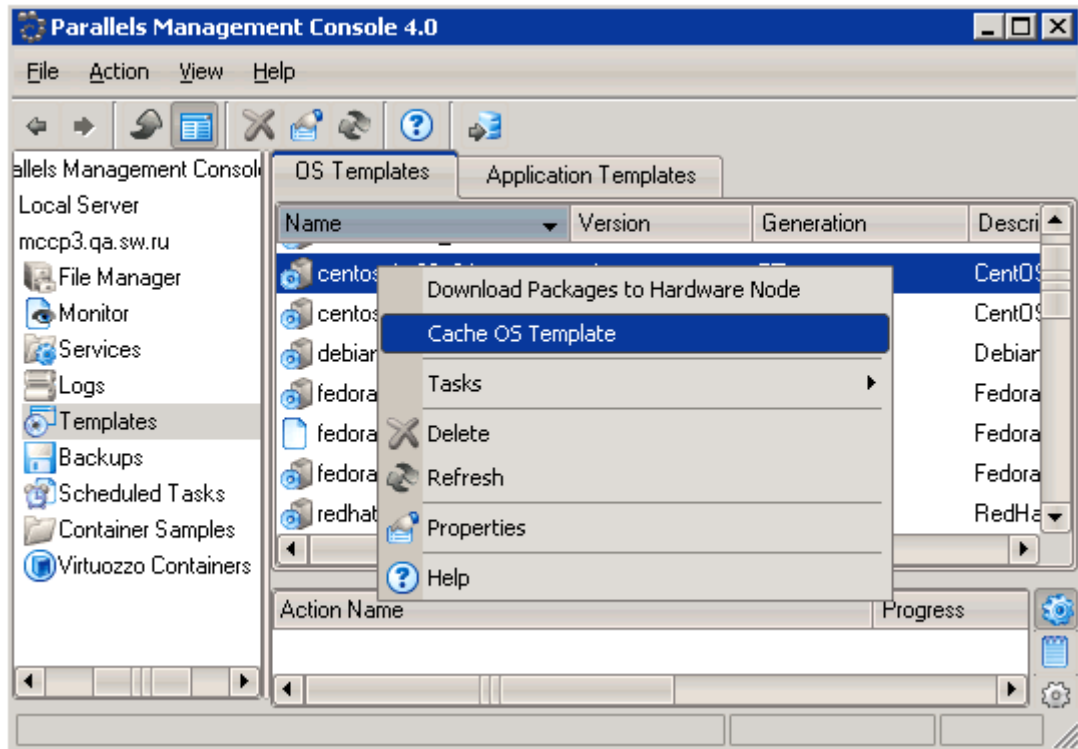


Figure 13: Management Console - Caching OS Template

Updating EZ Templates Packages Inside Container

Virtuozzo Containers 4.0 allows you to update software packages of the OS EZ template a Container is based on and of any application EZ templates applied to the Container. You can do it by using the `vzpkg update` utility. Assuming that Container 101 is based on the `redhat-e15-x86` OS EZ template, you can issue the following command to update all packages included in this template:

```
# vzpkg update 101 redhat-e15-x86
...
Updating: httpd                ##### [1/4]
Updating: vzdev                ##### [2/4]
Cleanup  : vzdev                ##### [3/4]
Cleanup  : httpd               ##### [4/4]

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

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

2. If you are going to update the cache of a commercial OS EZ template (e.g. Red Hat Enterprise Server 5 or SLES 10), you should first update software packages in the remote repository used to handle this OS EZ template and then proceed with updating the EZ template cache. Detailed information on how to manage repositories for commercial Linux distributions is provided in the [Setting Up Repositories and Proxy Servers for EZ Templates](#) section (p. 26).

As you can see from the example above, the `httpd` and `vzdev` applications have been updated for the `redhat-e15-x86` OS EZ template. If you wish to update all EZ templates (including the OS EZ template) inside Container 101 at once, you should execute the following command:

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

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

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

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

- 1 Open a list of Containers in the Management Console main window by selecting the **Virtuozzo Containers** item in the Hardware Node tree.
- 2 Double-click the name of the Container where you wish to add an EZ template to open the Container Manager.
- 3 Click the **Templates** item in the main tree of the opened Container Manager.
- 4 In the Management Console right pane, click either the **OS Templates** or **Application Templates** tab depending on what EZ template you wish to update.

- 5 Right-click the corresponding EZ template and select the **Update Installed Packages** option on the context menu.
- 6 In the **Select Updates** window, all the available Linux distributions are shown that fit your Virtuozzo Containers installation. In this window you can do one of the following:
 - If you wish to download and install all available EZ templates/template updates for a certain Linux distribution, click the **Finish** button to start updating to start updating your templates on the Hardware Node.
 - If you wish only certain EZ templates of a Linux distribution to be installed/updated on the Hardware Node, click on the plus sign beside the corresponding Linux distribution to display a list of application templates available for this distribution. You can then get detailed information about a particular template by selecting the corresponding template and viewing its data in the right part of the displayed window. By default, all new templates/template updates are set for downloading to and installing on the Hardware Node. To prevent this or that EZ template from being downloaded/installed, just clear its check box. When you are ready, click **Finish**.

On this screen, you can also select the **Force template(s) installation** check box to force the EZ template installation inside the Container. In this case no dependencies and no available versions of the application EZ template will be checked during its installation, which may cause the application EZ template to malfunction.

Creating Historical Mirror for Backed Up Container

If you have one or several Containers that are based on OS EZ templates and that were backed up long time ago, you may come across problems when trying to restore them on a Destination Node other than the Source Node (i.e. the Node where the Containers were hosted during their backing up). This may happen when the following conditions are simultaneously met:

- the Destination Node does not have one or several software packages included in the OS EZ template the restorable Container is based on and
- the missing packages are not available in the public repositories set to handle the OS EZ template; so, they cannot be downloaded and installed on the Destination Node. The absence of certain software packages in the public repositories may come out of the fact that some packages that were installed in the Container at the moment of its backing up have become obsolete in the course of time and been replaced with newer versions.

To avoid the aforementioned problems in the future, you can create the so-called 'historical' mirrors which will store an archive of all software packages present in the public repositories and containing the packages installed in the Container during its backing up.

Let us assume that you backed up a number of Containers based on the Fedora 8 OS EZ template. Now to be sure that you always have a repository containing all the necessary RPM packages for your backed up Containers, you wish to create a historical mirror of an official Fedora 8 repository. In our example below, we presume the following:

- The historical mirror will be located inside Container 101. You can use any OS template to base the Container on.

Note: We recommend that you always place your local repositories inside separate Containers to not compromise the Hardware Node security. In particular, it is of significant importance if you are going to provide access to your repositories through the `http` and `ftp` protocols.

- Container 101 is started and has the IP address of `123.145.145.123` assigned to it.
- The mirror will be created on a web server, i.e. it can be accessed from other networks through the `http` protocol.
- The `apache` web server is installed and running inside Container 101; the default document root for `apache` is `/var/www/html`, i.e. the `apache` web server stores its sites in the `/vz/root/101/var/www/html` directory on the Hardware Node.
- The `apache` user and group inside Container 101 are `apache`.

To create a historical mirror for Fedora 8, you should perform the following operations:

- 1 Change to the `/vz/root/101/var/www/html` directory and create the `fed8mirror` subdirectory within it:

```
# cd /vz/root/101/var/www/html
# mkdir fed8mirror
```

The `fed8mirror` subdirectory will store an archive of the Fedora 8 repository.

2 Change to the fed8mirror subdirectory

```
# cd fed8mirror
```

and execute the following command:

```
# rsync -av http://ftp.rhd.ru/pub/fedora/linux/releases/8/Everything/i386/os
```

This command will make a copy of the entire Fedora 8 repository located at <http://ftp.rhd.ru/pub/fedora/linux/releases/8/Everything/i386/os>. Please keep in mind that it may take a rather long run to copy all RPM packages to Container 101 depending on your bandwidth and the load on the Fedora mirror server.

Note: You can use any alternative Web or FTP site containing the Fedora 8 repository instead of the one indicated above.

3 Add your historical mirror to a list of repositories to be checked while performing operations on EZ templates related to Fedora 8 (in particular, while restoring Fedora-based Container backups). You can do it as follows:

- Create the `/vz/template/fedora-core/8/x86/config/os/default/repositories` file on the Hardware Node, if it is not yet present.
- Add the following string to the file:

```
http://123.145.145.123/var/www/html/fed8mirror
```

- Save the file.

Copying EZ Templates to Another Hardware Node

Parallels Virtuozzo Containers allows you to copy the installed OS and application EZ templates from one Hardware Node to another using the `vzmttemplate` utility. For example, you can copy the `fedora-core-8-x86` OS template installed on the Source Node to the Destination Node with the IP address of `192.168.0.9` by executing the following command:

```
# vzmttemplate -z root@192.168.0.9 fedora-core-8-x86
root@192.168.0.9's password:
Connection to Destination Node (192.168.0.9) is successfully established
Copying Template ".fedora-core-8-x86"
...
```

During the command execution, `vzmttemplate` will do the following:

- 1 Ask you for the password of the `root` user on the Destination Node.
- 2 Check whether the `fedora-core-8-x86` OS EZ template already exists on the Destination Node. If this templates is installed on the Destination Node, the command will exit.
- 3 Copy the `fedora-core-8-x86` configuration files from the Source Node to the Destination Node. Please keep in mind that the `fedora-core-8-x86` OS EZ template is not removed from the Source Node.
- 4 Run the `vzpkg create cache` command on the Destination Node to cache the OS EZ template and prepare it for the Container creation. Detailed information on this command is provided in the [Preparing OS EZ Template for Container Creation](#) section (p. 40).

Note: If you are going to copy an application EZ template, you should make sure that the corresponding OS EZ template (i.e. the OS template with which the application EZ template can be used) is installed on the Destination Node; otherwise, the operation will fail.

To check that the `fedora-core-8-x86` OS template has been successfully copied to the Destination Node, you can run the following command on this Node:

```
# vzpkg list
fedora-core-8-x86                2007-12-12 07:05:39
```

In Parallels Management Console, you can copy the installed OS and application EZ templates from the Source Node to any other Node registered in Parallels Management Console by doing the following:

- 1 Select the **Templates** item under the Hardware Node name where the template to be copied is installed and click either the **OS Templates** or **Application Templates** tab depending on whether you wish to copy an OS or application template, respectively.
- 2 In the Management Console right pane, right-click the template and select **Tasks --> Copy to Another Hardware Node** on the context menu:

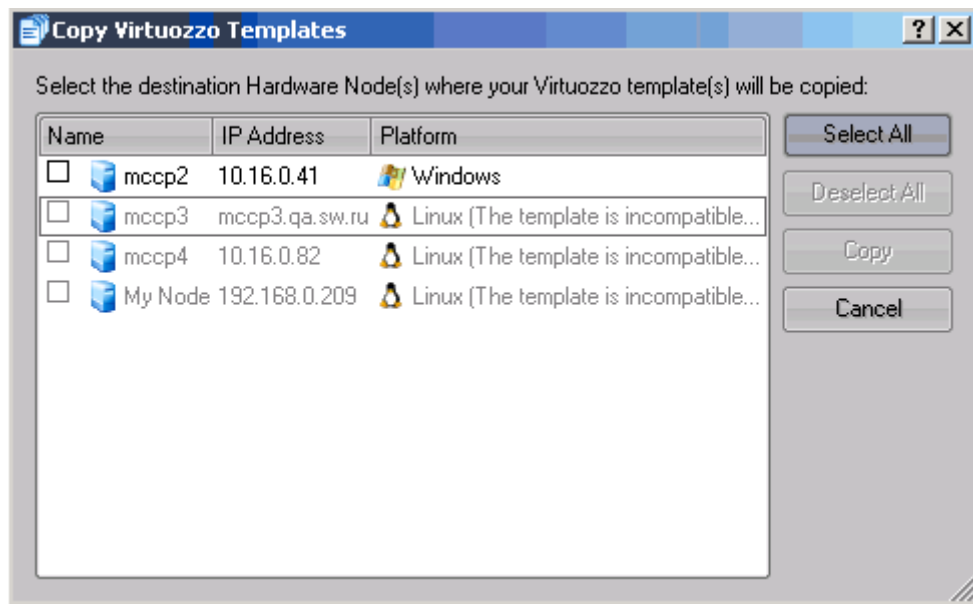


Figure 14: Management Console - Copying Templates

In the displayed window, you can view a list of Hardware Nodes currently registered in Parallels Management Console. You can copy the template to any of the listed Nodes provided it is compatible with the Host OS installed on this Node. To this effect, select the check box next to the template name and click the **Copy** button to the right of the table.

Removing Application EZ Template From Container

The `vzpkg remove` command allows you to remove one or several application EZ templates from a Container. A session below shows you an example how to remove the `mysql` EZ template from Container 101:

```
# vzpkg list 101
redhat-el5-x86                2007-05-21 02:21:56
redhat-el5-x86  mysql         2007-05-21 05:36:42
# vzpkg remove 101 mysql -w
vzpkg remove 101 mysql
Removed:
mysql
mysql-server
mysql-devel
mysql-bench
perl-DBD-MySQL
# vzpkg list 101
redhat-el5-x86                2007-05-21 02:21:56
```

You can see that the `mysql` EZ template has been successfully removed from Container 101. The `-w` option tells the `vzpkg remove` command to also delete from the Container all packages having interdependencies with `mysql`.

To remove an application EZ template from a Container in Parallels Management Console, do the following:

- 1 Open a list of Containers in the Management Console main window by selecting the **Virtuozzo Containers** item in the Hardware Node tree.
- 2 Double-click the name of the Container where from you want to delete a template to open the Container Manager.
- 3 Select the **Templates** item in the main tree of the opened Container Manager and click the **Application Templates** tab to see a list of application templates currently applied to the Container.
- 4 Right-click the application template you wish to delete and select the **Delete** option on the context menu.
- 5 Confirm your decision by clicking the **Remove** button in the displayed dialog window. In this window you can also do the following:
 - Select the **Forced removal** check box to force the template deletion from the Container.
 - Select the **Remove dependent packages** check box to also remove the packages having dependencies with the template.

Removing EZ Templates From Hardware Node

The `vzpkg remove template` command allows you to remove from the Hardware Node those EZ OS and application templates that you do need any more. The process of removing an EZ template includes deleting all the RPM packages comprising this template and all the caches available for this template (for EZ OS templates only). Please keep in mind that the template to be removed should not be applied to any Container; otherwise, the template deletion will fail. A session below demonstrates how to remove the `redhat-el5-x86` EZ OS template from the Hardware Node:

```
# vzpkg list
redhat-el5-x86                2008-02-16 12:50:17
fedora-core-7-x86            2008-02-18 14:23:12
# vzpkg remove template redhat-el5-x86
redhat-el5-x86 template was removed
# vzpkg list
fedora-core-7-x86            2008-02-18 14:23:12
```

To remove an EZ application template from the Node, you should additionally specify the `-F` option after the `vzpkg remove template` command. This option denotes the EZ OS template with which the EZ application template is compatible. For example, the following command can be used to remove the `mailman` EZ application template that is intended for running under RHEL 5 from your Node:

```
# vzpkg remove template -F redhat-el5-x86 mailman
redhat-el5-x86 mailman template was removed
```

You can also remove several EZ templates at once by specifying their names after `vzpkg remove template` and separating them by spaces. However, when handling application templates, keep in mind that you can delete only those application templates that relate to one and the same EZ OS template. For example:

```
# vzpkg remove template -F redhat-el5-x86 sitebuilder4 mailman
redhat-el5-x86 sitebuilder4 template was removed
redhat-el5-x86 mailman template was removed
```

In this example the `sitebuilder4` and `mailman` EZ application templates intended to run under RHEL 5 have been removed from the Hardware Node.

To remove one or more EZ templates in Parallels Management Console, do the following:

- 1 In the left pane of the Parallels Management Console main window, expand the Hardware Node where the template to be deleted is installed and select the **Templates** item:

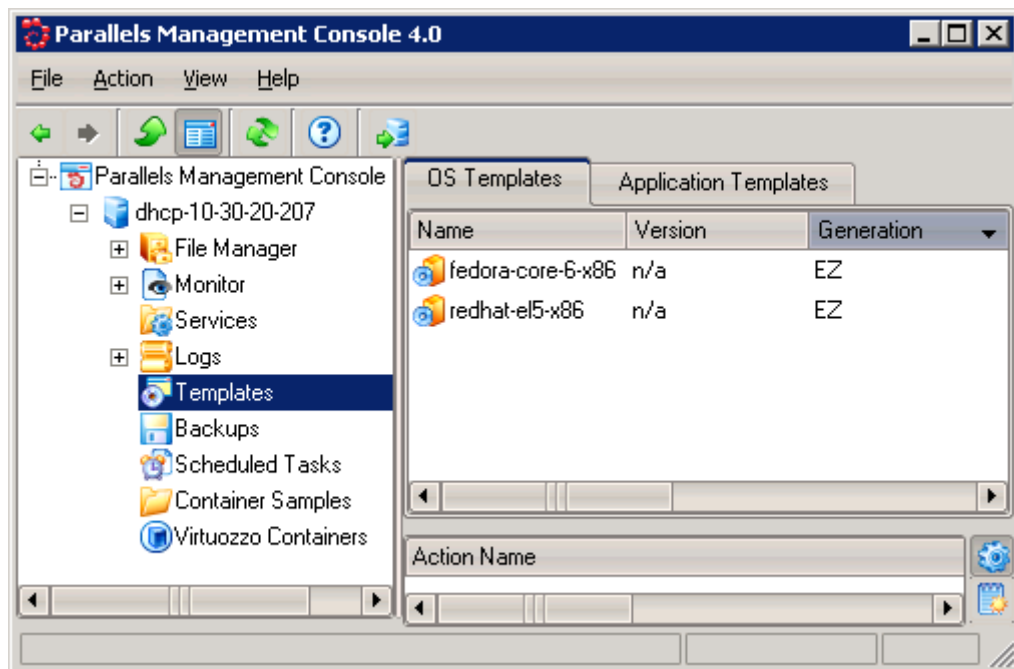


Figure 15: Removing EZ Templates From Hardware Node - Selecting Templates

- 2 In the right pane of the Parallels Management Console main window, do one of the following:
 - If you want to remove an EZ OS template from the Hardware Node, go to the OS Templates tab, right-click the EZ OS you wish to remove, and select Delete on the context menu.
 - If you want to remove an EZ application template from the Hardware Node, go to the Application Templates tab, right-click the EZ application template you wish to remove, and select Delete on the context menu.
- 3 Confirm your decision by clicking the Remove button in the displayed dialog window. In this window you can also do the following:
 - Select the Forced removal check box to force the template deletion from the Hardware Node.
 - Select the Remove dependent packages check box to also remove the packages having dependencies with the template.

Converting Containers Based on Standard Templates

You can use the `vzveconvert` utility to convert your existing Containers based on standard OS templates (and possibly having a number of standard application templates applied to them) to the EZ template-based ones. Let us assume that you wish to transform Container 101 based on the `fedora-core-3` standard OS template and having the `psa-fc3` standard application template applied to it and make it use the corresponding EZ templates: the `fedora-core-4-x86` OS and the `plesk8` application EZ templates. To this effect, you can issue the following command on the Hardware Node:

Notes: 1. You are highly recommended to create a Container backup before starting to convert your Container.

2. We also recommend that you read the `/usr/share/vztt/convert/README` file on the Hardware Node to learn about the specifics of converting Containers based on certain standard templates to EZ-based ones.

```
# vzveconvert 101
```

During the command execution, the following main operations are performed:

- 1 The `vzveconvert` utility checks if the `fedora-core-4-x86` EZ OS template and the `plesk8` EZ application templates are installed on the Hardware Node. If any of these EZ templates cannot be found on the Node, the command will fail.
- 2 The `pre-fedora-core-4` script located in the `/usr/share/vztt/convert/scripts` directory on the Hardware Node is executed. This script is meant to perform a number of pre-upgrade operations in the Container context.
- 3 The packages comprising the `fedora-core-3` standard OS template and the `psa-fc3` standard application template are compared with those available in the repositories for the `fedora-core-4-x86` EZ OS template and the `plesk8` EZ application template and if any updates are available, they are downloaded to the Hardware Node and installed there. You can also pass the `-C` option to the command to make `vzveconvert` look for the packages updates in the `vzpkg` cache only.
- 4 The packages that are included in the `fedora-core-3` and the `psa-fc3` standard templates and that cannot be found in the repositories for the `fedora-core-4-x86` and `plesk8` EZ templates, if any, are copied to the `/vz/template/vc/<UUID>/<rpm_name>` directory on the Node where `<UUID>` denotes the unique identifier of the Container (you can find out what unique ID is assigned to the Container by viewing the value of the `UUID` parameter in the Container configuration file after the `vzveconvert` execution) and `<rpm_name>` is the name of the package (together with its version, release, and architecture) absent from the EZ template repositories.

- 5 The packages installed in the `/vz/template/fedora-core/4/x86` directory on the Hardware Node (i.e. in the template area of the `fedora-core-4-x86` and `plesk8` EZ templates) are compared with those specified in the `/vz/template/fedora-core/4/x86/os/default/packages` and `/vz/template/fedora-core/4/x86/app/plesk8/default/packages` meta files and if some packages listed in these files are not present in this directory, they are also downloaded from the repository to the Node and installed there.
- 6 The symlinks to the package files in the `/vz/template/fedora-core-3/` and `/vz/template/psa` directories on the Node are replaced with symlinks to the files in the `/vz/template/fedora-core/4/x86` EZ template directory.
- 7 The following parameters in the `/etc/sysconfig/vz-scripts/101.conf` file are updated or set anew in accordance with the changes made: `OSTEMPLATE`, `TEMPLATES`, `UUID`, `TECHNOLOGIES`. Detailed information on these parameters is provided in the *Configuring Virtuozzo Containers* chapter of the *Parallels Virtuozzo Containers Reference Guide*.
- 8 The `post-fedora-core-4` script from the `/usr/share/vztt/convert/scripts` directory on the Node is executed. This script is meant to perform a number of post-upgrade operations in the Container context.

A full list of "standard OS template --> EZ OS template" and "standard application template --> EZ application template" transformations which can be performed in the current version of Virtuozzo Containers is provided in the `/usr/share/vztt/convert/os_table` and `/usr/share/vztt/convert/app_table` files on the Hardware Node, respectively. For example, you can issue the following command to learn what standard OS templates can be converted to their EZ counterparts:

```
# cat /usr/share/vztt/convert/os_table
...
fedora-core-1          fedora-core-4-x86
fedora-core-2          fedora-core-4-x86
fedora-core-3          fedora-core-4-x86
fedora-core-4          fedora-core-4-x86
fedora-core-5          fedora-core-5-x86
fedora-core-4-x86_64  fedora-core-4-x86_64
...
```

The left column provides information on standard OS templates whereas the right one informs you of the corresponding OS EZ templates.

Note: The `/usr/share/vztt/convert/app_table` file contains only the names of the EZ application templates having its own repositories for handling the packages included in these templates (e.g. all `plesk` EZ templates).

CHAPTER 4

Managing Standard Templates

This chapter familiarizes you with Virtuozzo standard templates and explains the ways to manage them.

Note: Virtuozzo standard OS and application templates have evolved to new OS and application EZ templates. Although Virtuozzo Containers 4.0 still supports standard templates to provide the compatibility with the previous versions of Virtuozzo Containers, you are highly recommended to use EZ templates on the Hardware Node and inside your Containers. This recommendation becomes even more actual taking into account the fact that all new versions of Linux distributions and applications (e.g. RHEL 5 and all applications to be used with it) are shipped as EZ templates only.

In This Chapter

Standard Template Lifecycle	68
Listing Standard Templates.....	69
Creating Container on Basis of OS Standard Template	70
Creating and Installing Application Standard Templates	73
Creating and Installing Standard Template Updates.....	75
Adding Standard Templates and Template Updates to Containers.....	77
Copying Templates to Another Hardware Node.....	79
Removing Standard Templates and Template Updates From Containers	80

Standard Template Lifecycle

Any template of any kind should be *created* first. The `vzpkgcreat` utility allows you to create a template. In addition, a number of already created templates are shipped by Parallels with Virtuozzo Containers 4.0.

Then, the template should be *installed on the Hardware Node* using the `rpm -i` command. For OS templates, as well as for OS template updates, the `vzpkgcache` program should be launched to *cache the new template into a tarball* for a precipitated creation of new Containers based on this template.

The template that has been installed on the Hardware Node, may be either *added to any number of Containers* with `vzpkgadd` (for application templates or their updates), or *a new Container may be created* on the basis of an OS template with the `vzctl create` command.

Any template excluding OS templates may be *removed from the Container* with the `vzpkgrm` command.

Finally, a template that is not used by any Container may be completely removed from the Hardware Node with the `rpm -e` command.

Listing Standard Templates

The `vzpkgls` utility allows you to list the templates installed on the Hardware Node. They may be already used or not used by certain Containers:

```
# vzpkgls
postgresql-as4      20060822
redhat-as4          20060918
```

As you see, the `redhat-as4` and `postgresql-as4` templates are available on the Hardware Node. Specifying a Container number as the parameter, this command prints the templates used by the specified Container:

```
# vzpkgls 101
redhat-as4          20060918
```

In Parallels Management Console, it is sufficient to choose the **Templates** item under the corresponding Hardware Node name and select either the **OS Templates** or **Application Templates** tab to see a list of the OS or application templates installed on the Node, respectively. For example:

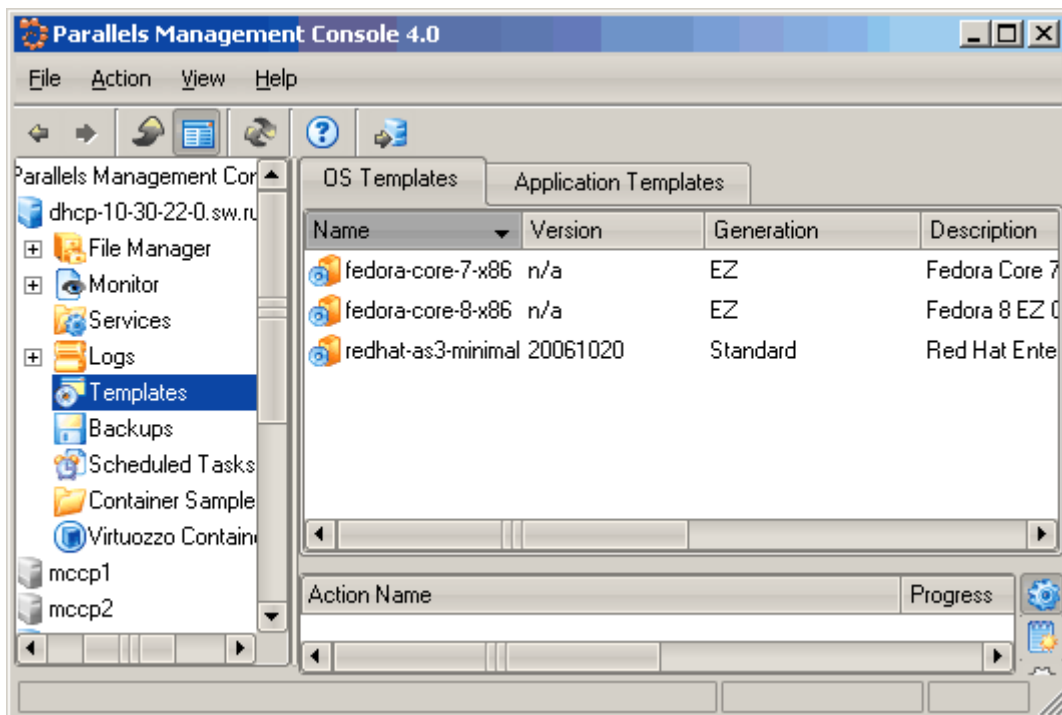


Figure 16: Management Console - Listing Templates

To see the templates used by a particular Container, double-click the needed Container in the Container list in the right pane to open this Container management window, and then again choose the **Templates** item and click the **OS Templates** or **Application Templates** tab in the left pane of the new window.

Creating Container on Basis of OS Standard Template

This subsection provides information on how you can create new Containers by using Virtuozzo OS standard templates.

Note: If you are running the Virtuozzo Containers 64-bit version for the IA-64 processors, you cannot use 32-bit OS templates to base your Containers on.

Choosing OS Template

Before starting to create your Container, you shall decide on which OS template you want to base the new Container. There might be several OS templates installed on the Hardware Node; use the `vzpkgls` command to find out the templates installed on your system:

```
# vzpkgls
postgresql-as4      20060322
redhat-as3-minimal 20060119
redhat-as4          20060418
```

As you can guess from the names, `redhat-as4` and `redhat-as3-minimal` are OS templates, whereas `postgresql-as4` is an application template. You can run an additional checkup with the `vzpkginfo` command to single out OS templates from application templates with confusing names; OS templates always have the `cached` section of their configuration file set to 'yes':

```
# vzpkginfo -g cached -b redhat-as4
yes
# vzpkginfo -g cached -b postgresql-as4
no
```

OS templates are pre-cached, so 'yes' in the output means that this is an OS template indeed. You can also list packages included into the template by running the `vzpkginfo -g base_version -b redhat-as4` command (for listing the packages of the base version) or `vzpkginfo -p -b redhat-as4` (for listing the packages of the current version). You can consult the [Parallels Virtuozzo Containers Reference Guide](#) for complete information on the `vzpkginfo` command.

Note: The `redhat-as3-minimal` OS template is used to create only the Service Container - a special Container used to manage your Containers by means of Parallels Management Console and Parallels Infrastructure Manager. You should use other OS templates installed on the Hardware Node to create regular Containers on their basis.

In Parallels Management Console, things are much simpler: just click consecutively the name of your Node, the **Templates** item and then the **OS Templates** tab to see a list of the installed OS templates:

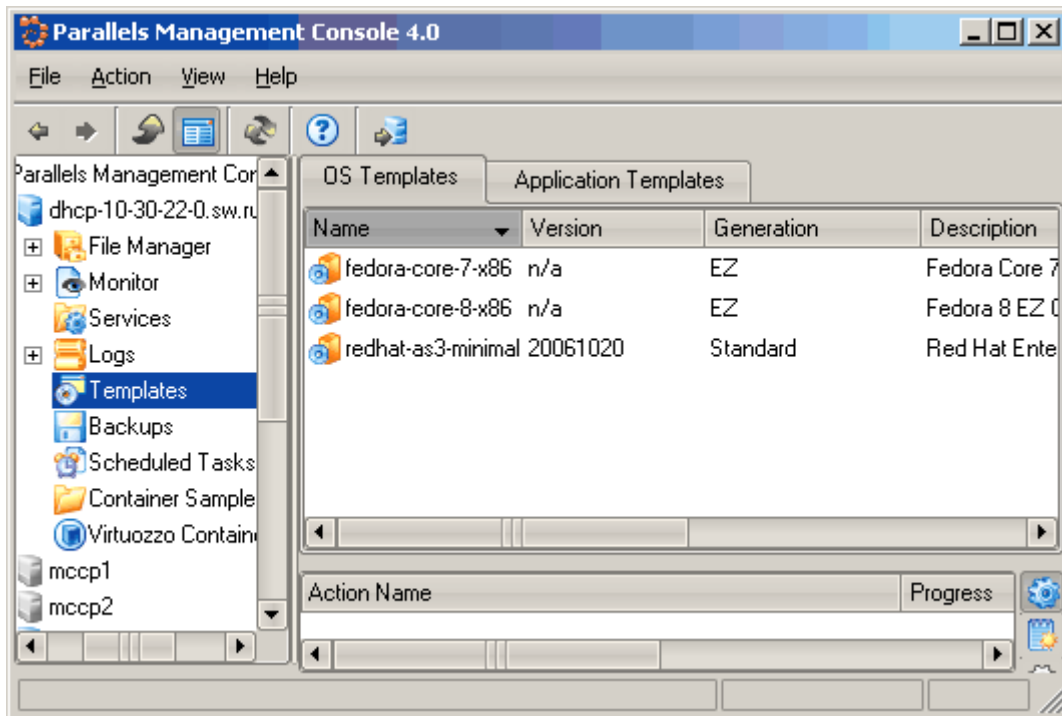


Figure 17: Management Console - Listing Templates

Creating Container

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

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

Thus, for example, you can create a new Container by typing the following string:

```
# vzctl create 101 --pkgset redhat-as4 --config basic
Creating Container private area
Container private area was created
```

In this case, Virtuozzo Containers 4.0 will create a Container with ID 101, the private area based on the `redhat-as4` OS template, and configuration parameters taken from the `ve-basic.conf-sample` sample configuration file.

If you specify neither an OS template nor a sample configuration, `vzctl` will try to take the corresponding values from the global Virtuozzo configuration file - `/etc/vz/vz.conf` . So you can set the default values in this file using your favorite text file editor, for example:

```
DEF_OSTEMPLATE="fedora-core-4"
CONFIGFILE="basic"
```

and do without specifying these parameters each time you create a new Container.

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

```
# vzctl create 101
Creating Container private area
Container is mounted
Postcreate action done
Container is unmounted
Container private area was created
```

In principle, now you are ready to start your newly created Container. However, typically you need to set its network IP address, hostname, DNS server address and `root` password before starting the Container for the first time. Please see the next subsection for information on how to perform these tasks.

Creating and Installing Application Standard Templates

As you may notice, the samba server and client applications are not included into the `redhat-as4` template. However, if any Container needs this application, a new application template may be created.

The `vzpkgcreat` utility serves to create any template. `vzpkgcreat` is part of the `vzpkgtools-build` package shipped with Virtuozzo Containers 4.0 4.0. This package is located in the `/virtuozzo/RPMS` directory of your Virtuozzo Containers distribution and is not installed by default during the Virtuozzo Containers installation. So, before you can start using the `vzpkgcreat` utility, you should first install the `vzpkgtools-build` package on your Hardware Node with the `rpm -i` command. For example:

```
rpm -ihv vzpkgtools-build-4.0.0-15.swsoft.i386.rpm
Preparing... ##### [100%]
 1:vzpkgtools-build ##### [100%]
```

Now that you have successfully installed the `vzpkgtool-build` package on your Node, you can start creating an application template for the samba server. In the example below, the samba RPM files from `redhat-as4` are used to create the template compatible with the `redhat-as4` OS template. The application template created is the *base* template (unlike a template update); this is specified with the `-b` command line option. The `-s` option specifies the OS template the application template being created is compatible with. The `-n` and `-d` options specify the template name and description, correspondingly.

To create this template, you will need Red Hat Enterprise Linux 4 CD 2 mounted:

```
# mount /media/cdrom
# cd /mnt/cdrom/RedHat/RPMS
# vzpkgcreat -b -s redhat-as4 -n 'samba template' -o /tmp/ \
-d 'samba server and client template for RHEL 4' samba\
samba-3.0.10-1.4E.i386.rpm samba-client-3.0.10-1.4E.i386.rpm\
samba-common-3.0.10-1.4E.i386.rpm
```

The new template is created in the directory specified by the `-o` option. Use the `rpm -ivh` command to install this template on the Hardware Node:

```
# rpm -ivh /tmp/samba-template-20060211-1.0-1.i386.rpm
Preparing... ##### [100%]
 1:samba-template ##### [100%]
# vzpkgls
redhat-as4          20060118
samba              20060211
```

As you see, the `samba` template is now available for installation inside any Container(s).

Important! Before starting to create an application template, please make sure that the 'version' and 'name' designations in the source RPM file correspond to those in the `rpm -qpi file_name.rpm` command output. If they do not, please rename the file accordingly. For example, if the RPM out of which the template is to be made has the name of `jdk-1_5_0_11-linux-i386.rpm`, however, the output of the `rpm -qpi` command is the following:

```
# rpm -qpi jdk-1_5_0_11-linux-i386.rpm
```

```
Name      : jdk
Version   : 1.5.0-11
Release   : fcs
...
```

you should rename the RPM file from `jdk-1_5_0_11-linux-i386.rpm` to `jdk-1.5.0_11-fcs-i386.rpm`.

Creating and Installing Standard Template Updates

Often, the system administrator needs to upgrade packages included in a template. This may happen, for example, in case the distribution vendor provides a security update. If such an update pertains to an application added from a template to certain Containers, these Containers should be updated to the new version with the corresponding bug fixed. The easiest way to do it is to create the corresponding template update, install it on the Hardware Node and add it to the needed Containers.

Note: Sometimes the auto-update feature of certain applications updates these applications directly inside the corresponding Containers, thus ignoring the file sharing capabilities of Virtuozzo Containers 4.0. To restore file sharing in such cases, the `vzcache` utility should be used, as is described in the *Cleaning Up Containers* subsection of the *Managing Resources* chapter in the *Parallels Virtuozzo Containers User's Guide*.

Let us assume that the `redhat-as4` OS template contains Version 3.6p1-6 of the `openssh` package. In the example below, a template update with `openssh-3.9p1-8` is created and installed. For this template, you will need the `openssh` binary RPM files compiled for `redhat-as4`:

```
# vzpkgcreat -c redhat-as4 -o /tmp openssh-3.9p1-8.RHEL4.1.i386.rpm \  
openssh-server-3.9p1-8.RHEL4.1.i386.rpm \  
openssh-clients-3.9p1-8.RHEL4.1.i386.rpm  
[ output skipped ...]  
# rpm -i /tmp/redhat-as4-template-20060328-1.0-1.i386.rpm  
# vzpkgls  
redhat-as4          20060328  
samba              20060211
```

Specify the `-c` option to indicate that the new template will be an OS template, and omit the `-b` option to indicate that you are creating a template update.

As you see, a new version of the `redhat-as4` template (named 20060328) is available now. After installing this template, do not forget to run the `vzpkgcache` utility to create a new Container cached archive with the latest version of the template.

Notes: 1. `vzpkgcreat` is part of the `vzpkgtools-build` package shipped with Virtuozzo Containers 4.0. This package is located in the `/virtuozzo/RPMS` directory of your Virtuozzo Containers distribution and is not installed by default during the Virtuozzo Containers installation. So, before you can start using the `vzpkgcreat` utility, you should first install the `vzpkgtools-build` package on your Hardware Node with the `rpm -i` command.

2. By default, if the template version is not specified explicitly, the `vzctl create` command creates Containers using the latest available version of the indicated OS template. Hence, from now on, new Containers in the example above will be created using the 20060328 version of the `redhat-as4` template.

3. When creating and installing template updates, the packages dependencies are not checked. So, you should make sure beforehand that the packages you add to a template update do not have dependencies conflicts with the packages of the Containers where the template update is to be added.

Adding Standard Templates and Template Updates to Containers

A number of application templates may be added to a Container immediately upon the Container creation. It happens if the sample configuration file on the basis of which the Container is being created has the `TEMPLATES` parameter enumerating the necessary templates. To add application templates to an existing Container, you should use the `vzpkgadd` utility.

The `vzpkgadd` utility handles the adding of both application templates and template updates to Containers. To successfully add a template to a Container, this Container should be running; otherwise, it is impossible to run the installation process in the Container context.

In the example below, the `samba` application template is added to Container 101:

```
# vzctl status 101
VEID 101 exist mounted running
# vzpkgadd 101 samba
Updating Container 101      |*****| [100%]
# vzpkgls 101
redhat-as4                20060628
samba                     20060628
```

Template updates may be applied to Containers in the same way. In this case, specify the version of the template. For example:

```
# vzpkgadd 101 redhat-as4 20060628
Updating Container 101      |*****| [100%]
# vzpkgls 101
redhat-as4                20060628
samba                     20060628
```

Now, Container 101 uses the updated RHEL 4 template, for example, with the 3.9p1-8 version of the `openssh` application, coming from the OS template update.

In Parallels Management Console, you can do this by performing the following operations:

- 1 Select the **Templates** item under the corresponding Hardware Node name and then click the **Application Templates** tab.
- 2 Select all the templates or template updates holding down the CTRL or SHIFT keys where necessary.
- 3 Right-click the selection and choose the **Install Into Containers** option.
- 4 Follow the instructions of the wizard.

If you are adding a template to only one Container, you can as well do the following:

- 1 Open the list of Containers in the Management Console main window by selecting the **Virtuozzo Containers** item in the Hardware Node tree.
- 2 Double-click the name of the Container where you want to add a template.
- 3 Select the **Templates** item in the main tree of the opened Container Manager, click the **Application Templates** tab, right-click somewhere in the top part of the Management Console right pane, and select the **Add Virtuozzo Application Template** option on the context menu.
- 4 Follow the instructions of the wizard.

Notes: 1. If a template update has been created without manually checking its possible dependencies conflicts with the Container software, adding this template to a Container might fail. You should create a template update with compatible packages.

2. The Virtuozzo Containers version for the 64-bit processors allows you to add only 64-bit application templates to your Containers.

Copying Templates to Another Hardware Node

Virtuozzo Containers 4.0 allows you to copy the installed OS and application standard templates from one Hardware Node to another using the `vzmtemplate` utility. For example, you can copy the `redhat-as4` OS template installed on the Source Node to the Destination Node with the IP address of `192.168.0.9` by executing the following command:

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

To check that the `redhat-as4` OS template has been successfully copied to the Destination Node, you can run the following command on this Node:

```
# vzpkg list
redhat-as4          20060918
```

In Parallels Management Console, you can copy the installed OS and application templates from the Source Node to any other Node registered in Management Console by doing the following:

- 1 Select the **Templates** item under the Hardware Node name where the template to be copied is installed and select either the **OS Templates** or **Application Templates** tab depending on whether you wish to copy an OS or application template, respectively.
- 2 In the Management Console right pane, right-click the template and select **Tasks --> Copy to Another Hardware Node** on the context menu:

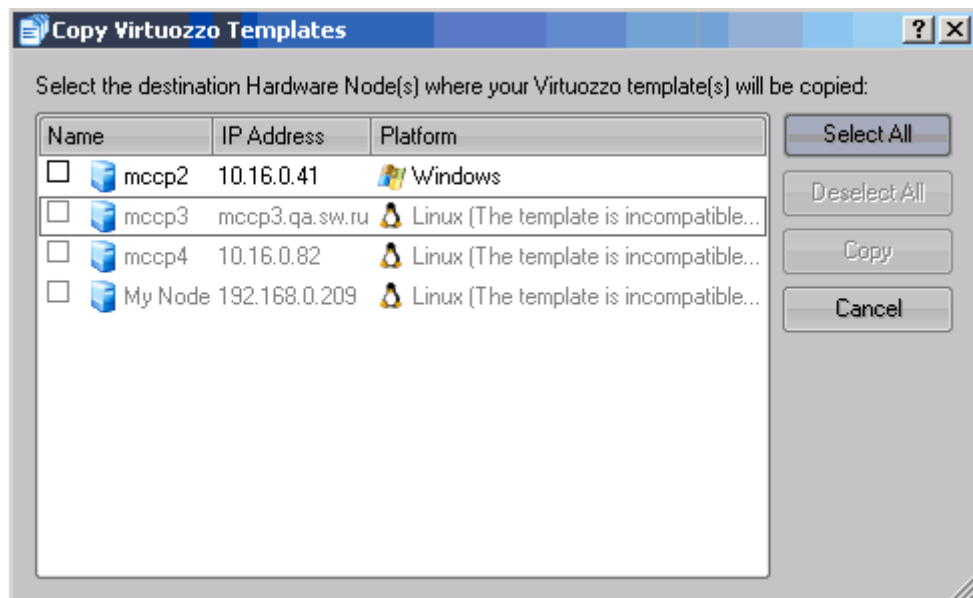


Figure 18: Management Console - Copying Templates

In the displayed window, you can view a list of Hardware Nodes currently registered in Parallels Management Console. You can copy the template to any of the listed Nodes provided it is compatible with the Host OS installed on this Node. To this effect, select the check box next to the template name and click the **Copy** button to the right of the table.

Removing Standard Templates and Template Updates From Containers

The Virtuozzo `vzpkgrm` utility allows either to downgrade a template added to a Container or to completely remove it. Thus, if Container users experience difficulties with `openssh-3.9p1-8`, it is possible to roll back this application to the original version (`openssh-3.6p1-6`). In this case, you should simply specify the destination version (mark) of the template:

```
# vzpkgrm 101 redhat-as4 20060628
[ output skipped ...]
# vzpkgls 101
redhat-as4          20060123
samba              20060628
```

By default, `vzpkgrm` removes template updates only, performing rolling back to the previous template version. To remove a particular template from a Container completely, the `-r` command line option should be specified. For example, after the `samba` template installation, it can be removed from the Container in the following way:

```
# vzpkgrm -r 101 samba
Updating Container 101      |*****| [100%]
# vzpkgls 101
redhat-as4                20061123
```

Now, the original state of Container 101 is restored – the `samba` application template and `redhat-as4` (Version 20060628) template update have been removed from this Container.

Parallels Management Console also allows you to remove a template or template update from a Container. To this effect, you should:

- 1 Open a list of Containers in the Management Console main window by selecting the **Virtuozzo Containers** item in the Hardware Node tree.
- 2 Double-click the name of the Container where from you want to delete a template or a template update to open the Container Manager.
- 3 Select the **Templates** item in the main tree of the opened Container Manager and click the **Application Templates** tab in the Management Console right pane to see a list of application templates currently applied to the Container.
- 4 Right-click the template or template update you wish to delete in the right pane.
- 5 Select the **Delete** option on the context menu.
- 6 Confirm you decision by clicking the **Remove** button in the displayed dialog window. In this window you can also do the following:
 - Select the **Forced removal** check box to force the EZ template deletion; in this case no dependencies will be checked before removing the template.
 - Select the **Remove dependent packages** check box to also remove the packages having dependencies with the template.

Index

A

About Parallels Virtuozzo Containers • 7
 About This Guide • 8
 Action Scripts • 20
 Adding Application EZ Templates to Container • 46
 Adding Standard Templates and Template Updates to Containers • 77
 Applications • 67, 73

C

Choosing OS Template • 70
 Container
 creating • 20, 70, 72
 private area • 20
 symlinks • 20
 updating • 56
 Converting Containers Based on Standard Templates • 65
 Copying EZ Templates to Another Hardware Node • 60
 Copying Templates to Another Hardware Node • 79
 Creating and Installing Application Standard Templates • 73
 Creating and Installing Standard Template Updates • 75
 Creating Container • 72
 Creating Container on Basis of OS Standard Template • 70
 Creating EZ Template • 21
 Creating Historical Mirror for Backed Up Container • 58
 Creating Local Repository • 28
 Creating Metafile for EZ Template • 22
 Creating Proxy Server for EZ Templates • 35

D

Differences Between OS and Application EZ Templates • 20
 DNS server • 72
 Documentation Conventions • 9

E

EZ Template

 adding to Container • 16, 20, 46
 application • 17, 20, 41, 46, 56
 area • 17
 caching • 16, 20, 40
 concepts • 16
 directory structure • 17
 installing • 20, 40, 41
 lifecycle • 20
 listing • 41, 44
 OS • 16, 20, 27, 28, 32, 40, 44, 54, 56
 removing • 20, 62
 repository • 16, 26, 40
 scripts • 17
 sets • 16, 17
 updating • 16, 20, 56

EZ Template Directory Structure • 17
 EZ Template Lifecycle • 20
 EZ Templates Basics • 16
 EZ Templates Overview • 15

F

Feedback • 11

G

Getting Help • 10

H

HN • See Hardware

I

Installing Application EZ Templates on Hardware Node • 41
 IP Address
 Container • 72

K

Keeping EZ Templates Up To Date • 47
 Kernel
 2.4 • 26

L

Listing EZ Templates • 44
 Listing Standard Templates • 69

M

Managing Default Repository • 27

- Managing EZ Templates • 14
- Managing Repositories for Commercial Linux Distributions • 32
- Managing Standard Templates • 67
- N**
- Node
 - Hardware • 69
- O**
- Organization of This Guide • 8
- P**
- Password
 - root • 72
- Pool • 20
- Preface • 6
- Preparing OS EZ Template for Container Creation • 40
- R**
- Removing Application EZ Template From Container • 62
- Removing EZ Templates From Hardware Node • 63
- Removing Standard Templates and Template Updates From Containers • 80
- Repository
 - default • 26, 27
 - for commercial Linux distributions • 26, 27, 32
 - local • 26, 28
- root
 - password • 72
- S**
- Scripts • 20
- Setting Up Proxy Server for EZ Templates • 35
- Setting Up Repositories and Proxy Servers for EZ Templates • 26
- Setting Up RHN Proxy Server for RHEL OS EZ Templates • 38
- Standard Template Lifecycle • 68
- Symlinks
 - overview • 72
- T**
- Tarball • 20
- Template
 - adding • 68, 77
 - application • 20, 67, 69, 70, 73, 77, 80
 - base • 73, 75
 - caching • 20, 68
 - choosing • 72
 - creating • 68, 73
 - directory • 20, 73
 - installing • 68, 73
 - lifecycle • 68
 - listing • 69
 - management utilities • 69, 73, 75, 80
 - OS (operating system) • 20, 67, 69, 70
 - overview • 20
 - removing • 68, 80
 - updates • 67, 75, 77, 80
- U**
- Understanding EZ Templates • 15
- Update
 - adding to Container • 77
 - creating • 75
 - installing • 75
 - removing • 80
 - template • 67, 75, 77
- Updating EZ Templates on Hardware Node • 48
- Updating EZ Templates Packages Inside Container • 56
- Updating OS EZ Template Caches • 54
- Updating Templates in Parallels Management Console • 52
- Updating Templates With vzpkg update template • 52
- Updating Templates with vzup2date • 48
- Upload and Install New EZ Templates on Hardware Node in Management Console • 42
- User
 - Container • 72
- Using vzmktml to Create EZ Template • 25
- Utilities
 - Container management • 72
 - EZ template management utilities • 40, 41, 44, 46, 56, 62
 - template management • 69, 73, 75, 77, 80
- V**
- Virtuozzo Containers
 - configuration file • 72
- Virtuozzo File System • 72
- Virtuozzo Templates Overview • 12
- VZFS • See Virtuozzo File System