

---

# Koan Documentation

*Release 3.0.2*

**Jörgen Maas**

Oct 08, 2021



<b>1</b>	<b>Koan</b>	<b>3</b>
1.1	Synopsis . . . . .	3
1.2	Description . . . . .	3
1.3	Listing remote Cobbler objects . . . . .	3
1.4	Learn more about Cobbler objects . . . . .	4
1.5	Reinstalling existing systems . . . . .	4
1.6	Installing virtualized systems . . . . .	4
1.7	Configuration management . . . . .	4
1.8	Environment variables . . . . .	5
1.9	Additional . . . . .	5
1.10	Author . . . . .	5
<b>2</b>	<b>Cobbler-Register</b>	<b>7</b>
2.1	Synopsis . . . . .	7
2.2	Description . . . . .	7
2.3	Details . . . . .	7
2.4	Environment variables . . . . .	8
2.5	Additional . . . . .	8
2.6	Author . . . . .	8
<b>3</b>	<b>Support for OpenVZ containers in Cobbler</b>	<b>9</b>
3.1	How to use it? . . . . .	9
3.2	How it works? . . . . .	9
3.3	Options for creating OpenVZ containers . . . . .	10
<b>4</b>	<b>Koan with ISO's</b>	<b>11</b>
<b>5</b>	<b>Installing virtual guests</b>	<b>13</b>
5.1	Example of installing a VM using a profile virtually . . . . .	13
5.2	Example of installing a VM using a system record . . . . .	13
5.3	Overrides . . . . .	13
<b>6</b>	<b>Reinstallation</b>	<b>15</b>
<b>7</b>	<b>Virtual Networking Setup</b>	<b>17</b>
7.1	Notice . . . . .	17
7.2	Networking . . . . .	17
7.3	Basics . . . . .	17
7.4	Network Manager . . . . .	18
<b>8</b>	<b>VMWare support for Koan</b>	<b>19</b>
8.1	Installing . . . . .	19

8.2	Setup . . . . .	19
8.3	What Happens . . . . .	20
<b>9</b>	<b>Release Notes for Cobbler 3.0.0</b>	<b>21</b>
9.1	Enhancements . . . . .	21
9.2	Bugfixes . . . . .	22
9.3	Upgrade notes . . . . .	22
<b>10</b>	<b>Indices and tables</b>	<b>25</b>

Koan - kickstart over a network, client side helper for cobbler

Contents:



Koan - kickstart over a network, client side helper for Cobbler

## 1.1 Synopsis

```
koan --server=hostname [--list=type] [--virt|--replace-self|--display] [--  
↪profile=name] [--system=name] [--image=name] [--add-reinstall-entry] [--virt-  
↪name=name] [--virt-path=path] [--virt-type=type] [--nogfx] [--static-  
↪interface=name] [--kexec]
```

## 1.2 Description

Koan stands for “kickstart-over-a-network” and is a client-side helper program for use with Cobbler. Koan allows for both network provisioning of new virtualized guests (Xen, QEMU/KVM, VMware) and re-installation of an existing system.

When invoked, Koan requests install information from a remote Cobbler boot server, it then kicks off installations based on what is retrieved from Cobbler and fed in on the Koan command line. The examples below show the various use cases.

## 1.3 Listing remote Cobbler objects

To browse remote objects on a Cobbler server and see what you can install using Koan, run one of the following commands:

```
koan --server=cobbler.example.org --list=profiles  
koan --server=cobbler.example.org --list=systems  
koan --server=cobbler.example.org --list=images
```

## 1.4 Learn more about Cobbler objects

To learn more about what you are about to install, run one of the following commands:

```
koan --server=cobbler.example.org --display --profile=name
koan --server=cobbler.example.org --display --system=name
koan --server=cobbler.example.org --display --image=name
```

## 1.5 Reinstalling existing systems

Using `--replace-self` will reinstall the existing system the next time you reboot.

```
koan --server=cobbler.example.org --replace-self --profile=name
```

```
koan --server=cobbler.example.org --replace-self --system=name
```

Additionally, adding the flag `--add-reinstall-entry` will make it add the entry to grub for reinstallation but will not make it automatically pick that option on the next boot.

Also the flag `--kexec` can be appended, which will launch the installer without needing to reboot. Not all kernels support this option.

## 1.6 Installing virtualized systems

Using `--virt` will install virtual machines as defined by Cobbler. There are various overrides you can use if not everything in Cobbler is defined as you like it.

```
koan --server=cobbler.example.org --virt --profile=name
koan --server=cobbler.example.org --virt --system=name
koan --server=cobbler.example.org --virt --image=name
```

Some of the overrides that can be used with `--virt` are:

Flag	Explanation	Example
<code>--virt-name</code>	name of virtual machine to create	testmachine
<code>--virt-type</code>	forces usage of qemu/xen/vmware	qemu
<code>--virt-bridge</code>	name of bridge device	virbr0
<code>--virt-path</code>	overwrite this disk partition	<i>/dev/sda4</i>
<code>--virt-path</code>	use this directory	<i>/opt/myimages</i>
<code>--virt-path</code>	use this existing LVM volume	<i>VolGroup00</i>
<code>--nogfx</code>	do not use VNC graphics (Xen only)	(does not take options)

Nearly all of these variables can also be defined and centrally managed by the Cobbler server.

If installing virtual machines in environments without DHCP, use of `--system` instead of `--profile` is required. Additionally use `--static-interface=eth0` to supply which interface to use to supply network information. The installer will boot from this virtual interface. Leaving off `--static-interface` will result in an unsuccessful network installation.

## 1.7 Configuration management

Using `--update-config` will update a system configuration as defined by Cobbler.

```
koan --server=cobbler.example.org --update-config
```



Additionally, adding the flag `--summary` will print configuration run stats.

Koan passes in the system's FQDN in the background during the configuration request. Cobbler will match this FQDN to a configured system defined by Cobbler.

The FQDN (Fully Qualified Domain Name) maps to the system's hostname field.

## 1.8 Environment variables

Koan respects the `COBBLER_SERVER` variable to specify the Cobbler server to use. This is a convenient way to avoid using the `--server` option for each command. This variable is set automatically on systems installed via Cobbler, assuming standard kickstart templates are used. If you need to change this on an installed system, edit `/etc/profile.d/cobbler.{csh,sh}`.

## 1.9 Additional

Reading the Koan manpage, <https://cobbler.github.io> or this [readthedocs](#) project is highly recommended.

## 1.10 Author

Michael DeHaan < michael.dehaan AT gmail >

Revised by: Enno Gotthold < matrixfueller@ gmail.com >



---

## Cobbler-Register

---

cobbler-register - create a Cobbler system record

### 2.1 Synopsis

```
cobbler-register [--server=<cobbler.example.org>] --profile=<cobbler-profile-name> └  
↔ [--fqdn=<hostname>]
```

### 2.2 Description

Running `cobbler-register` on a system will create a Cobbler system record for that system on a remote Cobbler server. No changes will be made on the system itself.

### 2.3 Details

When installing new machines into a Cobbler managed datacenter/lab, it helps to not have to manually enter in the network information for those systems. Using `cobbler-register` either from a kickstart or a live environment (or even SSH) can help seed the Cobbler database.

Network information is discovered automatically for all physical interfaces. `cobbler-register` will attempt to discover the hostname, though if `localhost.localdomain` is found, it will have to use some other data for the Cobbler system record. This is probably not what you want, so specify `--fqdn` in this instance to override that registration value.

For this to work, the `register_new_installs` setting must be enabled on the remote Cobbler server.

When the remote system record is created, for safety reasons, it will be set in Cobbler to be “netboot disabled”. Use `cobbler system edit --name=foo --netboot-enabled=1` to set the machine to reinstall, where “foo” is the name of the new system record.

## 2.4 Environment variables

`cobbler-register` respects the `COBBLER_SERVER` variable to specify the Cobbler server to use. This is a convenient way to avoid using the `--server` option. This variable is set automatically on systems installed via Cobbler, assuming standard kickstart templates are used. If you need to change this on an installed system, edit `/etc/profile.d/cobbler.{csh,sh}`.

## 2.5 Additional

Reading the Koan manpage, [www.cobbler.github.io](http://www.cobbler.github.io) or this Readthedocs-Project is highly recommended.

## 2.6 Author

Michael DeHaan <[michael.dehaan AT gmail](mailto:michael.dehaan@gmail.com)>

Revised by: Enno Gotthold <[matrixfueller@gmail.com](mailto:matrixfueller@gmail.com)>

---

## Support for OpenVZ containers in Cobbler

---

THIS FUNCTIONS CONSIDERED AS ALPHA STAGE FOR TESTING AND LIMITED USAGE! USAGE IN PRODUCTION CAN BE DANGEROUS! YOU WARNED!

Cobbler is amazing tool for deploying barebones and virtual machines and I think it is suitable for deploying OpenVZ containers too.

Current support for OpenVZ is rather basic, but I think this functionality can reach level we have now for KVM.

### 3.1 How to use it?

Because OpenVZ container is in nature chrooted environment we use Cobbler + Koan to create this on OpenVZ-enabled node. For Cobbler and Koan in case of OpenVZ all operations is similar - we should define distros, automated installation files, profiles, systems and so on with some additions. Now we do all operations only for RHEL/CentOS 6. It may be suitable for recent Fedoras, but we do nothing for other distributions.

### 3.2 How it works?

All options keeps on cobbler side as for other VMs. Besides of common options you can use openvz-specific ones by defining them as `vz_` prefixed, low-cased variables from this list: `KMEMSIZE`, `LOCKEDPAGES`, `PRIVVMPAGES`, `SHMPAGES`, `NUMPROC`, `VMGUARPAGES`, `OOMGUARPAGES`, `NUMTCP SOCK`, `NUMFLOCK`, `NUMPTY`, `NUMSIGINFO`, `TCPSNDBUF`, `TCPRCVBUF`, `OTHERSOCKBUF`, `DGRAMRCVBUF`, `NUMOTHERSOCK`, `DCACHESIZE`, `NUMFILE`, `AVNUMPROC`, `NUMIPTENT`, `DISKINODES`, `QUOTA-TIME`, `VE_ROOT`, `VE_PRIVATE`, `SWAPPAGES`, `ONBOOT` (See `ctid.conf(5)` for meaning of this parameters). Because cobbler does not have a place to keep CTID you MUST use it in `ks_meta` (as you can see in example below)! We use it on cobbler-side to be able allocate them from one place. We turn off PXE-menu creation for OpenVZ containers to not pollute this menu.

For example:

```
# cobbler profile add --name=vz01 --distro=CentOS6-x86_64 --autoinst=/your/
↳autoinst.cfg \
    --ks_meta="lang=ru_RU.UTF-8 keyb=ru vz_ctid=101 vz_swappages=0:2G vz_
↳numproc=120:120" \
    --repos="centos6-x86_64-os centos-x86_64-updates" \
```

(continues on next page)

(continued from previous page)

```

--virt-type=openvz \
--virt-ram=1024 \
--virt-cpus=1

# cobbler system add --name=vz01 \
--profile=vz01 \
--virt-type=openvz \
--virt-ram=1024 \
--virt-cpus=1

# cobbler system edit --name=vz01 \
--hostname=vz01.example.com \
--interface=eth0 \
--mac=YOUR_MAC_HERE \
--static=1 \
--ip-address=YOUR_IP \
--netmask=MASK \
--gateway=GATEWAY_IP \
--name-servers=NAME_SERVERS_IPs

```

On the Koan side:

```
# koan --server=COBBLER_IP --virt --system=vz01
```

This will start installation process. `ovz-install` script will install all packages and groups listed in `$packages` section. As root for installation `ovz-install` will use `/vz/private/$VEID` (`/vz/private/101` for example above), that can be overridden with `vz_ve_private` variable in `ks_meta` (eg. `vz_ve_private=/some/path` or `vz_ve_private=/other/path/$VEID` or `vz_ve_private=/some/path/101` - `$VEID` will be replaced with `CTID`). After installation `ovz-install` will process “services” option from `autoinst` like it do `anaconda` and run post-installation script, defined in `autoinst` (only in `chroot`), so you can tune the container for your needs. At the end of process `ovz-install` process installed tree to be truly OpenVZ container - creates dev files, change init scripts etc. Created container started after that, so you should be able to log in to it with root and password you defined for root in `autoinst` file.

### 3.3 Options for creating OpenVZ containers

You should set `virt-type` to “openvz” in profile or system to create OpenVZ container.

```

--virt-file-size    not used for now. We think we can use it for logical volume_
↳ creation, or quoting
                    filesystem usage, or for creating containers in ploop-file.
--virt-ram          as for other VMs
--virt-cpus         as for other VMs
--virt-path         not used now. Container will be created in /vz/private/
↳ $VEID, where $VEID will be replaced by
                    openvz with CTID (container ID). Can be redefined by vz_ve_
↳ private variable you can place in ks_meta.
--virt_bridge      not used now.

```

---

## Koan with ISO's

---

While most of Cobbler installing is about enabling scripted network installation, can also define image objects which track ISOs that Koan can find and see. Currently this only works when using Koan with QEMU/KVM for installation (sorry, no Xen or VMware support yet). This can be used, for instance, to install Windows via Koan.

For this to work best, the ISO must be available by the same NFS path on all hosts. It need not be mounted and Cobbler nor Koan will copy it.

```
cobbler image add --name=image_name --file=nfs://hostname.example.org:/path/  
↳example/acme-os-installer-image.iso [--virt-ram=512] [--virt-file-size=10] [...  
↳etc...]
```

And on the Koan side, just run:

```
koan --list=images --server=cobbler.example.org  
koan --virt --image=image_name --server=cobbler.example.org
```

Koan will then mount the NFS location and begin a fully virtualized installation using the virtual metadata and info stored in Cobbler.

You may remember that Cobbler has objects like “distros” and “profiles”. Images are another type of object, but act similarly.

System objects in Cobbler may also attach to an image instead of a profile, though not all attributes of the system apply to an image – for instance, we may care about the number of interfaces, but the networking configuration automatic that normally happens as part of a kickstart can’t happen for an image based install.

Avoid image based installs if at all possible – use kickstart where you can, and images for foreign content where you can not.





---

## Installing virtual guests

---

The main feature of Koan is contact the Cobbler server, learn about a configuration from Cobbler, and make that virtualized installation happen. One other feature is [KoanDoesReinstall reinstallation of existing systems] but perhaps the more important one is how to do virtualized installs.

Koan is its own separate tool, a separate package from Cobbler, that is designed for use with a remote Cobbler server. (The same folks that work on Cobbler work on Koan and it's available from the same repositories)

```
yum install koan
```

It is a very small tool and generally does not always need to be updated when Cobbler is updated, but keeping Koan updated ensures you have all the latest features available. In general, the major release numbers of Cobbler and Koan should match, but it's not so important if the minor release numbers don't match.

### 5.1 Example of installing a VM using a profile virtually

```
koan --server=cobbler.example.org --virt --profile=foo
```

### 5.2 Example of installing a VM using a system record

```
koan --server=cobbler.example.org --virt --system=foo
```

### 5.3 Overrides

Koan is designed to install things as set up in Cobbler to ensure installs are consistent and repeatable.

Often though, users of Koan may not be Cobbler server administrators or may want to install a VM on a test system – so they'll want to override some things as stored in Cobbler. Koan allows an extensive system of overrides to tweak what Cobbler tells us about how a particular Cobbler profile should be installed.

Flag	Explanation	Example
<code>--virt-name</code>	name of virtual machine to create	testmachine
<code>--virt-type</code>	forces usage of qemu/xen/vmware	qemu
<code>--virt-bridge</code>	name of bridge device	virbr0
<code>--virt-path</code>	overwrite this disk partition	<i>/dev/sda4</i>
<code>--virt-path</code>	use this directory	<i>/opt/myimages</i>
<code>--virt-path</code>	use this existing LVM volume	<i>VolGroup00</i>
<code>--nogfx</code>	do not use VNC graphics (Xen only)	(does not take options)

Nearly all of these variables can also be defined and centrally managed by the Cobbler server and are also described in the Cobbler manpage in depth.

## CHAPTER 6

---

### Reinstallation

---

Cobbler's helper program, Koan, can be installed on remote systems.

It can then be used to reinstall systems, as well as its original purpose of installing virtual machines.

Usage is as follows:

```
koan --server cobbler.example.com --profile profileName
koan --server cobbler.example.com --system systemName
```

Koan will then configure the bootloader to reinstall the system at next boot. This can also be used for OS upgrades with an upgrade kickstart as opposed to a kickstart that specifies a clean install.



---

## Virtual Networking Setup

---

### 7.1 Notice

For Xen and QEMU/KVM virtual machines to be able to get outside access they will need to have a virtual bridge configured on the virtual host. (If you're using VMware this page won't apply to you)

While "virbr0" should automatically be set up if you are using a newer libvirt, it's not a real bridge and you won't be able to contact your guests from outside – it's a private network. So you most likely do NOT want to use virbr0 if you are doing anything useful. "xenbr0" if you have that, is fine to use.

The following instructions show about how to set up bridging manually which must be done on a host to make things work as you would expect.

Remember if you have a "xenbr0", you can use that though – it's a real bridge. If you want something more specific you can still create your own. xenbr0 is created in most versions of RHEL by xend startup.

### 7.2 Networking

Virtualization networking in Koan uses "bridged" mode. This is so that guests by default can be connected to from the outside world, which is very important for them to be able to do useful things.

If a network bridge already exists, Koan will be able to use that, though in some cases, you'll have to create your own bridge in order to get Koan to work. You do this by modifying the network configuration on your virtual host, and then using the Koan parameter `--virt-bridge=bridgename`.

(As we mentioned, if you use virbr0, it's a fake bridge, so be aware you won't be able to ssh into your guests... However Koan can use that if you REALLY want to)

### 7.3 Basics

The configuration in this section is adapted from [here](#). We're going to be ignoring the parts in that article about using virt-install as we want to use Koan – we want to make our virtualized configurations be managed server side, by Cobbler – and to take advantage of things that Cobbler provides for us, like syslog setup, templating, remote profile browsing, etc.

So here's the short rundown of what you need to do to create a bridge if you do not already have one. Once you do this once for each guest, you're set up – so if you have bare metal profiles for Cobbler set up, it may make sense to make those profiles set up your bridge at install time as well.

Set up `/etc/sysconfig/network-scripts/ifcfg-peth0` to define your physical NIC:

```
DEVICE=peth0 ONBOOT=yes BRIDGE=eth0 HWADDR=XX:XX:XX:XX:XX:XX
```

Substitute the X's in HWADDR for the mac address of the NIC you'd get from `/sbin/ifconfig`

Now set up the bridge interface: `/sbin/sysconfig/network-scripts/ifcfg-eth0`:

```
DEVICE=eth0 BOOTPROTO=dhcp ONBOOT=yes TYPE=Bridge
```

As the above link recommends, “You also want to add an iptables rule that allows forwarding of packets on the bridged physical NIC (otherwise DHCP from your guests won't work)”.

```
# service iptables start # iptables -I FORWARD -m physdev --physdev-is-bridged -j ACCEPT # service  
iptables save
```

Alternatively, you could also disable iptables (at your own risk).

Now you should be able to use Koan as follows:

```
koan --server=bootserver.example.org --profile=RHEL5-i386 --virt
```

To force a specific choice, you can use `--virt-bridge` and specify the name of any bridge you like. Note that this must be a `/real/` bridge, and not a physical interface. If you use a physical interface things will not work.

```
koan --server=bootserver.example.org --profile=RHEL5-i386 --virt --virt-  
↪bridge=peth0
```

Hopefully that helps address some of the basics around virtual networking setup. The above instructions work for both Xen and QEMU/KVM.

(If you encounter problems with the above please bring them up on the gitter-chat list)

## 7.4 Network Manager

At the time of writing this (Fedora 10), Network Manager does not support bridging.

You should also set `NM_MANAGED=No` in the configuration file for your physical interface to disable Network-Manager on hosts that use it.

(This has the side effect of tricking firefox into offline mode on startup, but we are hopefully talking about servers here... if this bothers you, go into `about:config` and search for `networkmanager`. Turn off the firefox network manager toolkit).

---

## VMWare support for Koan

---

One goal of Cobbler is to abstract out all your installation types. Basically the idea is create a profile one, use it for PXE, reinstallations (`--replace-self`) and virtualized machines all at once.

### 8.1 Installing

To get this working, you will need to be using Cobbler to host your PXE infrastructure.

You can then, using special Koan syntax (one single command line invocation), install a Cobbler profile `/inside/VMware`, pulling the RAM and Disk storage parameters from Cobbler.

This allows you to treat VMware just as another installation type... The same kickstart and parameters you use to install inside your bare metal machines and other virtualization types can be used for VMware in very similar ways.

### 8.2 Setup

First, grab yourself some caffeine.

Second, set up VMware as normal. (As of the time of writing this, we support VMware server and workstation, and there is a patch being worked on to support nearly all VMware variants)

- Install the latest VMware server. At the time of testing, I managed to get VMware to work on EL 5.
- Install the latest any-any patch to make VMware actually work on said platform
- run `/usr/bin/vmware-config.pl` as appropriate
- enter your free serial number when requested
- make sure the VMware service is enabled and started

Then do some minimal Cobbler setup:

- add Cobbler profiles and Cobbler system records to taste
- Cobbler must be set up to be your PXE infrastructure, which it should be doing already hopefully
- “cobbler system add” a new system and specify a mac address in the range of `00:50:56:00:00:00` to `00:50:56:3F:FF:FF` (VMWare’s allowed range).

Note that MAC addresses outside that range will not work, and you cannot skip the step of creating them in that range. Unlike Xen/KVM they are also not auto-generated by Koan. . . The reason for this is, without the mapping in Cobbler, PXE control over what profile is installed is meaningless.

Then, on the guest OS, run the following:

```
koan --server=cobbler.example.org --system=foosball --virt --virt-type=vmware
```

### 8.3 What Happens

Koan will ask Cobbler for the RAM and Disk needs associated with “foosball”, or rather, the profile associated with “foosball”. It will then automatically create your VMware configuration file (vmx) and disk image, and set that image up so that the rest of the installation happens over PXE.

Thus the installation is in tandem between PXE and the VMware libraries, though all of the profile data is still managed by Cobbler.



## 9.1 Enhancements

- Use new dracut ip option for configuring static interfaces (koan).
- Add a whitelist of directories in order to persist a `cobbler sync`.
- Add proxy support for get-loaders, signature update and reposync.
- Add initial support for DJBDNS.
- Enable external YUM repo mirroring through a proxy server.
- DHCP configuration now also supports the per interface gateway setting.
- A new `interface_type BMC` was added which also can be managed with DHCP.
- Yaboot was updated to 1.3.17.
- Add ability to have per-profile/per-system `next_server` values (#1196).
- Add `--graphics` option to Koan.
- Improved input validation and error handling.
- Support `virtio26` for generic QEMU fallback in Koan.
- Debian network config: add support for tagged vlan only bonding interfaces.
- Documentation has been converted into rST and is now included with the source tree.
- Integrated pyflakes into the build system and resolved hundreds of issues.
- Integrated pep8 (coding style) into the build system and resolved thousands of issues.
- Add a new field to the system type `ipv6_prefix` (#203).
- Minor update to CSS; make better use of screen (tables) (cobbler-web).
- Add support for an empty system status.
- If `dns-name` is specified, set it as DHCP hostname in preference to the `hostname` field.
- Allow user to choose whether or not to delete item(s) recursively (cobbler-web).
- Set `ksdevice` kernel option to MAC address for ppc systems as `bootif` is not used by yaboot.

- Return to list of snippets/kickstarts when snippet/kickstart is saved (cobbler-web).
- Layout in snippet/kickstart edit form has been improved (cobbler-web).
- Better handling of copy/remove actions for subprofiles (API and cobbler-web).
- Make kickstart selectable from a pulldown list in cobbler-web (#991).

## 9.2 Bugfixes

- Changed Apache configuration directory in Ubuntu 14.04 (#1208).
- build\_reporting no longer fails with an empty string in ignorelist (#1248).
- Kickstart repo statement, filter invalid values: `gpgcheck`, `gpgkey` and `enabled` (#323).
- Several improvements to Debian/Ubuntu packaging.
- Some class/method names have been changed to make the code more intuitive for developers.
- Remove `root=` argument in Koan when using `grubby` and `replace-self` to avoid booting the current OS.
- Exit with an error if the `cobblerd` executable can't be found (#1108, #1135).
- Fix Cobbler sync bug by `xmlrpclib` returning `NoneType` object.
- Dont send the Puppet environment when system status is empty (#560).
- Cobbler-web kept only the most recent interface change (#687).
- Fix broken `gitdate`, `gitstamp` values in `/etc/cobbler/version`.
- Prevent disappearing profiles after `cobblerd` restart (#1030).
- Add missing icons to `cobbler_web/content` (#679).
- `cobbler-ext-nodes` was broken with `mgmt_classes` defined at the profile level (#790).
- Properly name the VLAN interface in the manual page.
- Fix wrong address of the Free Software Foundation.
- Remove legacy (EL5/6) cruft from the RPM specfile.
- Koan: use the `print` function instead of the `print` statement.
- Minor improvement to LDAP configuration (#217).
- Improvements to the unittest framework.
- Removed several unused functions from `utils`.
- List of authors is now automagically generated.

## 9.3 Upgrade notes

- Support for LDAP configuration through Koan has been removed.
- Support for `redhat_management` (Spacewalk/Satellite) has been moved to contrib. Users of this functionality should checkout `contrib/redhat-management/README`.
- Monit support has been removed; you really need to use a CMS to manage your services.
- Support for remote kickstart templates and files been removed (eg. `kickstart=http://`).
- All object names are now validated like that of the system object.
- The use of `parent` and `distro` on subprofiles are now mutually exclusive.
- Support for `s390/s390x` has been removed.

- Support for ia64 (Itanium) has been removed.
- Support for the MySQL backend has been removed.
- Support for deprecated fieldnames (`subnet`, `bonding_master`, `bonding`) has been removed.
- Cobbler now requires python 2.7 and Koan now requires python 2.6.
- Red Hat specific default kernel options have been removed from the settings file.
- Support for Func integration has been moved to contrib. Users of this functionality should checkout `contrib/func/README`.
- Deprecated Koan LiveCD: moved to contrib.



# CHAPTER 10

---

## Indices and tables

---

- `genindex`
- `modindex`
- `search`