

ProActive Parallel Suite

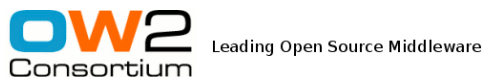


An Open Source Middleware For Parallel, Distributed, Multicore Computing

Virtual Machine Management

Version 1.1.0

The OASIS Research Team and ActiveEon Company



Virtual Machine Management v1.1.0 Documentation

Legal Notice

This library is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; version 3 of the License.

This library is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this library; if not, write to the Free Software Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA

If needed, contact us to obtain a release under GPL Version 2, or a different license than the GPL.

Contact: proactive@ow2.org or contact@activeeon.com

Copyright 1997-2010 INRIA/University of Nice-Sophia Antipolis/ActiveEon.

Mailing List

proactive@ow2.org

Mailing List Archive

<http://www.objectweb.org/wws/arc/proactive>

Bug-Traking System

<http://bugs.activeeon.com/browse/PROACTIVE>

Contributors and Contact Information

Team Leader

Denis Caromel
INRIA 2004, Route des Lucioles, BP 93
06902 Sophia Antipolis Cedex
France
phone: +33 492 387 631
fax: +33 492 387 971
e-mail: Denis.Caromel@inria.fr

Contributors from OASIS Team

Brian Amedro
Francoise Baude
Francesco Bongiovanni
Florin-Alexandru Bratu
Viet Dung Doan
Yu Feng
Imen Filali
Fabrice Fontenoy
Ludovic Henrio
Fabrice Huet
Elaine Isnard
Vasile Jureschi
Muhammad Khan
Virginie Legrand Contes
Eric Madelaine
Elton Mathias
Paul Naoumenko
Laurent Pellegrino
Guilherme Peretti-Pezzi
Franca Perrina
Marcela Rivera
Christian Ruz
Bastien Sauvan
Oleg Smirnov
Marc Valdener
Fabien Viale

Contributors from ActiveEon Company

Vladimir Bodnartchouk
Arnaud Contes
Cédric Dalmasso
Christian Delbé
Arnaud Gastinel
Jean-Michel Guillaume
Olivier Helin
Clément Mathieu
Maxime Menant
Emil Salageanu
Jean-Luc Scheefer
Mathieu Schnoor

Former Important Contributors

Laurent Baduel (Group Communications)
Vincent Cave (Legacy Wrapping)
Alexandre di Costanzo (P2P, B&B)
Abhijeet Gaikwad (Option Pricing)
Mario Leyton (Skeleton)
Matthieu Morel (Initial Component Work)
Romain Quilici
Germain Sigety (Scheduling)
Julien Vayssiere (MOP, Active Objects)

Table of Contents

Chapter 1. Hardware Virtualization Overview	1
1.1. Hardware Virtualization Overview	1
1.2. Software compatibility	2
Chapter 2. Virtual Machine Management QuickStart	3
2.1. Virtualization Layer Setup	3
2.1.1. Overall prerequisites	3
2.1.2. Editor dependent	3
2.2. Troubleshooting	12

Chapter 1. Hardware Virtualization Overview

Hardware Virtualization has reached an important point of interest in the datacenter and the workstation. Whether you want to save money by saving power, footprint or cooling system, or improve reactivity, fault tolerance and availability of your services, you will probably use hardware virtualization capabilities. You can also improve desktop management and delivery by using hardware virtualization for your workstations. This tool is designed to manage the main virtualization softwares.

This section gives a brief introduction about how hardware virtualization works with the different kind of virtualization softwares we can meet.

1.1. Hardware Virtualization Overview

Hardware virtualization allows to run several operating systems on a unique machine. This is done thanks to a specific software called “Virtual Machine Monitor” (VMM). Every virtualization solution needs a particular operating system to work (“Dom0” in case of bare metal VMM and “host OS” in case of hosted virtualization. We will dig deeper into this later on). The VMM can either emulates specific hardware devices or grants access to real hardware to the virtual machine (or guests operating system ie. VM). Thus, one can benefit from many features for different purposes. We identify two kinds of VMM:

- **Type 1 hypervisor** (or bare metal)

This type of hypervisor is named "bare metal" because it does not need any operating system beneath to work. Depending on what kind of product you are using, you will take advantage of different drivers for different hardwares and, maybe, you will not be able to get the software functional because of unsupported hardwares (see a description at: <http://en.wikipedia.org/wiki/Hypervisor>). Most often, you will need a “DOM 0” virtual machine, which is in fact the operating system that stands for your workspace to manage your virtual infrastructure. The virtual machine monitor (VMM) itself is a small footprint software (about 50Mo) which will only check and schedule underlying hardware access. To set up your virtual environment you need an extra software (xm for XenOss, xe for XenServer, vmx for ESX, ...) which is useable directly from your DOM 0. This type of hypervisor is, in general, more efficient and faster than others hypervisors as it implements its own hardware access policy at the lowest possible level. Thus, you completely avoid overhead induced by an underlying operating system. Here are some well-known hypervisors: [XenServer](#)¹, [Hyper-V](#)², [xVM Server](#)³, [VMware ESX/ESXi](#)⁴, [Xen OSS](#)⁵

- **Type 2 hypervisor** (or hosted)

This type of hypervisor is running on top of an operating system and is seen as a common process so does any virtual machine container. We can thus measure the host operating system overhead that treats the guest operating system as a child process whereas it was more a "brother" in the case of type one hypervisor. Furthermore, every kind of virtualization process cannot be used in hosted virtualization for some reasons (browse http://en.wikipedia.org/wiki/Platform_virtualization for more details). Here are the main type two hypervisors: [Virtualbox](#)⁶, [VMware Server](#)⁷, [KVM](#)⁸

The most important hardware emulation a user has to be informed of is the network part. We can essentially distinguish three sorts of network provisions:

- **Bridge Networking** - Here, the network routing is made at the 3rd level of the OSI/ISO stack. Your computer's Network Interface Card (NIC) binded to your company/internet network is setup in “PROMISCUOUS MODE” to intercept not only packet intended for the host/dom0 IP but also for newly created Virtual interfaces designed to provide network to virtual machines. Depending on what kind of virtualization product you use, you may have to create the virtual interfaces by yourself (brctl/openvpn on linux systems and Network Manager on Windows). With this solution, your guest operating systems are part of the company network like your host computer. That means a fully point-to-point connectivity between both hardware/virtual machine on your network.

¹ <http://community.citrix.com/cdn/xs>

² <http://www.microsoft.com/hyper-v-server/en/us/default.aspx>

³ <http://www.sun.com/software/products/xvmserver/index.xml>

⁴ <http://www.vmware.com/products/vi/esx/>

⁵ <http://www.xen.org/>

⁶ <http://www.virtualbox.org/>

⁷ <http://www.vmware.com/products/server/>

⁸ <http://www.qumranet.com/>

- **Nat/Route** - The network routing is also made at the 3rd level of the OSI/ISO stack. This time, a newly created virtual interface will ensure POST/PRE-ROUTING and MASQUERADE for your virtual machine to have network access. It is really easy to find such a configuration example on the internet ([Here for linux for instance](#)⁹).
- **NAT user** - This time network routing is made at the virtualization software layer. For the outer world (included for the host machine), the virtual machines are unreachable. Most often, the associated subnet has the pattern 10.0.0.1/24 ; 10.0.0.1 and 10.0.0.2 are the IPs for the host machine (only seen by the virtual machines) and every virtual machine belongs to its own subnet (virtual machines cannot speak to each other).

1.2. Software compatibility

The table below lists the different supported virtualization softwares. To get more information on a particular VMM, you can refer to the appropriate subsection.

Vendor	Product	Virtualization type
VMware	Server<2, Workstation<6.5	Hosted
	Server > 2	Hosted
	ESX/ESXi > 2.5	Bare Metal
Citrix	XenServer	Bare Metal
Xen.org	Xen OSE	Bare Metal
Oracle	Oracle xVM Virtualbox	Hosted
Microsoft	Hyper-V	Bare Metal
Open Source Community	Qemu - KVM - LXC - OpenVZ	Hosted

⁹ <http://www.revsys.com/writings/quicktips/nat.html>

Chapter 2. Virtual Machine Management QuickStart

2.1. Virtualization Layer Setup

To be able to deploy using this Virtual Machine Management API, you must first choose and setup the virtualization software you want to use. Refer to the associated section to get pieces of information.

2.1.1. Overall prerequisites

You must setup the virtualization software. That begins by choosing the kind of VMM you want to use depending on your needs. You essentially have to focus on the kind of operating system you want to virtualize, the kind of hardware you have, the type of network to provide for your virtual machines to access network, and if performances are a critical requirement for your application.

Most often and at the time this documentation was written (07/02/2009), the bare metal hypervisors do not allow advanced settings for network. The main solution is to deploy a dedicated virtual machine that will handle network traffic for NAT-ing or Bridge-ing (that can represent a major flaw). Bare metal VMMs do not support many hardwares. You have to ensure that your configuration is compatible (depending on what kind of guest OS you want to run) and you also may need specific hardware. For instance, using XenServer for Windows virtualization, it is compulsory to own an intel vt, vtx or vtx2 chip. Besides, notice that for compliant kernel in case of [paravirtualization](#)¹, you have to use modified guest operating system). And finally, the main products are designed for datacenter virtualization or advanced desktop delivery, they may lack some important features like hierarchical snapshots, removable devices handling, sounds... The main advantage to run type one hypervisors is that they expose better performances and benefit from fast pace evolution due to hard pressure from industry client users. Hosted virtualization softwares are more desktop user friendly and often used for desktop/client virtualization, critical software development... They always expose at least NAT/Bridge networking capabilities and do not require to run modified guest operating system. The main flaw is that they run pretty much slower than type one hypervisor due to host operating system overhead...

Once you have chosen the kind of VMM you want to use, you have to “deploy” (here, “deploy” has a special meaning. For virtual infrastructure managers, it means “register your virtual machines”). Just create a virtual machine referring to your virtualization software documentation. Once this is done, depending on the VMM you chose, you may have different capabilities. Refer to the associated section to know how to use this api with your VMM.

2.1.2. Editor dependent

Here are listed the steps to follow depending on the VMM you have chosen for your Infrastructure.

2.1.2.1. VMware

For all handled VMware products, if you want to be able to pass information to your virtual machines (using pushData & getData methods), you have to install VMware guest tools. This step is really straight forward. For VMware Server prior 2, power the virtual machine on, on the VMware Server console, click Devices, install guest tools. This mounts a CD into your VM. If you run a Windows OS, launch the installer.exe, and on a Linux one, run the installer.sh. For VMware Server posterior 2, launch your virtual machine, from the virtual machine administration page, right column, click install guest tools. This will also mount a CD within your VM. For VMware ESX/ESXi, follow the step from the Vi Client software and ditto for Workstation client. To be able to manage this kind of environment, you need to use classes from the package org.ow2.proactive.virtualizing.vmwarevi.*; see the corresponding section below.

2.1.2.1.1. VMware ESX/ESXi

ESX is currently the killer software at VMware and is declined in different version depending on what kind of support you have subscribed. However you can use the last ESXi which is pretty akin to ESX except that it is free and comes with less features and

¹ <http://en.wikipedia.org/wiki/Paravirtualization>

hardware support. Both ESX and ESXi can be purchased in “installable and embedded” versions. The embedded release is distributed by OEM vendors with their servers solution hence you do not have to bother with it. You can see it as a firmware settled on the hardware flash memory. It only contains the drivers it needs to work and no more. You cannot imagine make it run on an other platform than the one it is made for. The installable release comes with more hardware support and can be installed on a common hard disk or directly boot from a USB key. See <http://www.vm-help.com/> and user guide for more information about that.

2.1.2.1.2. VMware Server and Workstation

VMware Server (formerly GSX) is a major free software for server virtualization as type two hypervisor at VMware's. Depending on what release of VMware Server you are running you will have to focus on different things.

- **VMware Server < 2.** The VMware Server releases prior to 2 are shipped with a rich client (VMware Server Console) to be able to connect to every compatible server. This means that you need two different softwares to manage your virtual environment. If you want to use that product: First, create a new user belonging to the vmware group ; this user will be the administrator for your system. Download the software from [VMware web site](#)² and install it. The installation is straight forward, the only thing you have to focus on is setup. Prior to launch your VMware Server for the first time you have to run a configuration script, “vmware-config”, that will fix several useful information. Among these information, there are the previously created user for administration, directory for virtual machine disks storage, directory for documentation, libraries etc. and the port for remote authentication. Set this port and remember its value. To be able to manage this kind of environment, you need to use VMware-Vix, see the corresponding section below.
- **VMware Server >= 2.** VMware Server after version 2 are [VMware Virtual Infrastructure](#)³ compliant. They come with a web user interface for management (allowing end user to remotely manage his virtual environment). If you want to use this software, first create a new user belonging to the vmware group. This user will be the administrator of your virtual environment. Then download VMware Server from <http://www.vmware.com/products/server/> and install it. Prior to launch VMware Server for the first time, you have to run a perl script, vmware-config, to setup your VMware environment. During that step, the most important thing to do is to remember the authentication port you set. We may need it at the deployment time. To be able to manage this kind of environment, you need to use VMware-VI, see the corresponding section below.

2.1.2.1.3. VMware-Vix

This section shows how to use this API to manage VMware products compliant with [VMware-vix](#)⁴. Be sure to download and install the latest release if not installed during your VMware product installation. You also need to put every jar located in `src/vmware-vix/lib` and `src/core/lib` in the classpath. You need to set the `jna.library.path` variable when launching you java application (`-Djna.library.path=path-to-vmwarevixlibrary`). The last requirement is due to the fact that VMware-vix is a management API written in C and that we use [JNA](#)⁵ to use the C library from a Java environment. Package to use: `org.ow2.proactive.virtualizing.vmwarevix.*`

Here is an example showing how to connect to and manage your environment:

```
VMwareVMM vmm = new
VMwareVMM("psychoquack.inria.fr", "user", "password", 904, VMwareVMM.Service.vmwareServer);
VMwareVM myVM = vmm.getNewVM("/user/jmguilla/home/vmware/myUbuntu/myUbuntu.vmx");
myVM.pushData("guestinfo.myvar", "myval");
myVM.powerOn();
//some work
myVM.powerOff();
```

The first line connects to the VMM thanks to `new VMwareVMM(String,String,String,int,VMwareVMM.Server)`. The first tree parameters are: url of the machine running the VMware Server, a user belonging to the vmware group and his password. The following parameter is the authentication port set during the installation and the last parameter represents the type of VMM you are communicating with (in this case, `vmwareServer`. You can use VMware-vix to manage several VMware softwares, but we strongly

² http://www.vmware.com/support/server/doc/releasenotes_server.html

³ <http://www.vmware.com/products/vi/>

⁴ <http://www.vmware.com/support/developer/vix-api/>

⁵ <https://jna.dev.java.net/>

recommend to use it only with VMware server prior to release 2.0). Then you either can get all registered virtual machines thanks to `vmm.getVirtualMachines()` or get a given virtual machine with `vmm.getNewVM("/vmx/file/path")`. Before powering on your vm, you can pass it some data that you will be able to read from the inside:

```
myVM.pushData("guestinfo.myvar", "myval");
```

Note that to use that feature, you are compelled to have the guest tools installed within the virtual machine as explained above. To read the data within the virtual machine:

```
vmware-guestd --cmd "info-get guestinfo.myvar"
```

This is useful to define some parameters for the applications running inside the virtual machine. Finally, just power your vms on and off.

2.1.2.1.4. VMware-VI

This section shows how to use this API to manage VMware products compliant with [VMware-VI](#)⁶. Put every jar located in `src/vmware-vi/lib` and `src/core/lib` in the classpath. Package to use: `org.ow2.proactive.virtualizing.vmwarevi.*`

Here is an example about how to connect to and manage your environment:

```
VMwareVMM vmm = new VMwareVMM("https://psychoquack.inria.fr:8333/sdk", "user", "password");
VMwareVM myVM = vmm.getNewVM("myUbuntu");
VMwareVM myClone = myVM.clone("myUbuntuClone");
myClone.pushData("guestinfo.myvar", "myval");
myClone.powerOn();
//some work
System.out.println("myvar: " + myClone.getData("guestinfo.myvar"));
myClone.powerOff();
myClone.destroy();
```

The first line connects to the VMM thanks to `new VMwareVMM(String,String,String)`. The three parameters are: url (protocol aware, `http://localhost:8222/sdk` and `https://localhost:8333/sdk` by default but can be overridden during installation) of the machine running the VMware Server, a user belonging to the vmware group and his password. Then you either can get all registered virtual machines thanks to `vmm.getVirtualMachines()` or get a given virtual machine with `vmm.getNewVM("machinesName")`. Before powering on your VM, you can pass it some data that you will be able to read from the inside:

```
myVM.pushData("guestinfo.myvar", "myval");
```

Note that to use that feature, you are obliged to have the guest tools installed within the virtual machine as explained above. To read the data within the virtual machine:

```
vmware-guestd --cmd "info-get guestinfo.myvar"
```

This is useful to define some parameters for the applications running inside the virtual machine. You are also able to clone the virtual machine. The clone is an immutable snapshot of the parent virtual machine and can be destroyed. Immutable means that once the vm is powered off, every changes will be lost. Furthermore, be sure to destroy a clone as this feature erases all vm components from the computer. Finally, you can power the vm on and off.

2.1.2.2. XenServer

This section shows how to use this API to manage Citrix's XenServer product. Put every jar situated in `src/xenserver/lib` and `src/core/lib` in the classpath. Package to use: `org.ow2.proactive.virtualizing.xenserver.*`

⁶ <http://www.vmware.com/products/vi/>

XenServer is the property of [Citrix](http://www.citrix.fr/)⁷ and is based on the free and open source [Xen OSS](http://www.xen.org/)⁸. It is a type one hypervisor compatible with several OEM and which is said to be faster than VMware's equivalent products as it uses [paravirtualization technique](http://en.wikipedia.org/wiki/Paravirtualization)⁹ (whereas VMware uses [full-virtualization - binary translation](http://en.wikipedia.org/wiki/Full_virtualization)¹⁰). To use it, just refer to the [user guide](#)¹¹ for installation. You can also install it on an external usb device thanks to that [tutorial](#)¹². If you want enable efficient cloning feature for your XenServer environment, be sure to use an ext3 file system and nothing else. If you use an LVM base repository, when cloning a virtual machine you won't benefit the Copy on Write (COW) feature that allows two virtual machines to share a common backing disk file and saves changes in separated files. An ext3 base repository does. To change the backing file system of your virtual machine pool try [this tutorial](#)¹³

Here is an example about how to connect to and manage your environment:

```
XenServerVMM vmm = new XenServerVMM("https://psychoquack.inria.fr/", "user", "password");
XenServerVM myVM = vmm.getNewVM("machinesName");
XenServerVM myClone = myVM.clone("clonesName");
myClone.pushData("myvar", "myval");
myClone.powerOn();
//some work
System.out.println("myvar: " + myClone.getData("myvar"));
myClone.powerOff();
myClone.destroy();
```

The first line connects to the VMM thanks to `new XenServerVMM(String,String,String)`. The three parameters are: url (protocol aware, `https://my.xenserver.com/`) of the machine running the Xen Server, a user with sufficient permissions to manage your infrastructure and his password. Then you can either get all registered virtual machines thanks to `vmm.getVirtualMachines()` or get a given virtual machine with `vmm.getNewVM("machinesName")`. Before powering on your vm, you can pass it some data that you will be able to read from the inside:

```
myVM.pushData("myvar", "myval");
```

This is useful to define some parameters for the applications running inside the virtual machine. You are also able to clone and destroy the virtual machine. Furthermore, be sure to destroy a clone as this feature erases all vm components from the computer. Finally, you can power the vm on and off.

The way we provide the capability to pass information to the virtual machine is special. The implementation of the virtualization layer embeds a small database to handle infrastructure information. We use that database to store and retrieve the information. To store the information, we tag a VM's datastore space with its NIC MAC address (this information is used as key, that's why we need to ensure that every virtual machine doesn't own a similar MAC address. If you don't change XenServer settings, it ensures that every assigned NIC MAC addresses are different.). Finally, you can use this API from the inside of the virtual machine to get the information you set outside.

2.1.2.3. Virtualbox

This section shows how to use this API to manage Sun's xVM Virtualbox product. Be sure to use a xVM Virtualbox release shipped with `vboxwebsrv` application and start it (`vboxwebsrv -p 18083 -t 20`). If you want to use `pushData` & `getData` methods, install guest additions in the vm. Put every jar situated in `src/virtualbox/lib` and `src/core/lib` in the classpath. Package to use: `org.ow2.proactive.virtualizing.virtualbox.*`

Virtualbox is a type two hypervisor created by Innotek and since February 2008, it is a property of Sun Microsystems. This virtualization software is currently the only one running on Linux, Windows, MacOS and Solaris and is a precursor of some virtualization techniques

⁷ <http://www.citrix.fr/>

⁸ <http://www.xen.org/>

⁹ <http://en.wikipedia.org/wiki/Paravirtualization>

¹⁰ http://en.wikipedia.org/wiki/Full_virtualization

¹¹ http://www.citrix.com/lang/English/lp/lp_1688615.asp

¹² <http://community.citrix.com/blogs/citrite/dannyw/2009/03/12/Installing+XenServer+on+an+USB+Drive>

¹³ http://www.tokeshi.com/index.php?option=com_content&task=view&id=5025

such as seamless virtualization and 3D hardware virtualization. At the moment, only Virtualbox non open source edition is shipped with vboxwebsrv application which is compulsory. Be sure to install a version of Virtualbox that comes with that add-on. The installation process is really straight forward and the software easy to use, see [the project web site](#)¹⁴ for more information.

To be able to manage xVM Virtualbox environnement, start vboxwebsrv on the machine where xVM Virtualbox is installed. You can use personal settings, the only one with which you have to be careful is the session time out. We encourage a session time out of 20s. We also notices that some vboxwebsrv releases had some troubles with authentication. If it is your case, just submit the following command:

```
VBoxManage setproperty websrvauthlibrary null
```

and then restart the vboxwebsrv program.

Here is an example showing how to connect to and manage your environment:

```
VirtualboxVMM vmm = new VirtualboxVMM("http://psychoquack.inria.fr:18083/", "user", "password");
VirtualboxVM temp = vmm.getNewVM("machinesName");
VirtualboxVM clone = temp.clone("clonesName");
clone.pushData("test", "value");
clone.powerOn();
//some work
System.out.println("test: " + clone.getData("test"));
clone.powerOff();
clone.destroy();
```

The first line connects to the VMM thanks to new VirtualboxVMM(String,String,String). The tree parameters are: url (protocol aware, http://my.vboxwebsrv.com:0000/) of the vboxwebsrv software, a user belonging to vboxusers group and his password. Then you can either get all registered virtual machines thanks to vmm.getVirtualMachines() or get a given virtual machine with vmm.getNewVM("machinesName"). Before powering on your vm, you can pass it some data that you will be able to read from the inside:

```
myVM.pushData("myvar","myval");
```

To get the information from the inside of the virtual machine:

```
VBoxControl guestproperty get myvar
```

This is useful to define some parameters for the applications running inside the virtual machine. You are also able to clone and destroy the virtual machine. To clone the virtual machine, you must snapshot the state of the virtual machine thanks to the xVM Virtualbox management tool. This shows that the clone is in fact a descendant of the vm's snapshot. Finally, you can power the vm on and off.

2.1.2.4. Microsoft Hyper-V

Microsoft Hyper-V is a type one hypervisor shipped with Microsoft Server 2008 and following releases. To use this management tool, you have the choice between two different implementations:

- [WMI Management tools](#).¹⁵
- [WS-Management](#)¹⁶ Windows implementation.

The latter is pure SOAP management interface binded to WMI thanks to a Microsoft plugin. The former is a WBEM windows implementation using DCOM protocol for remote management instead of HTTP as specified in the profile. Refer to the appropriate section for more information.

¹⁴ <http://www.virtualbox.org/>

¹⁵ http://en.wikipedia.org/wiki/Windows_Management_Instrumentation

¹⁶ <http://en.wikipedia.org/wiki/WS-Management>

2.1.2.4.1. WS-Management Windows implementation

To be able to manage Microsoft Hyper-V with this implementation, you must put every jars located in `src/hyperv-winrm/lib` and `src/core/lib` in the class path. This module uses [Wiseman](https://wiseman.dev.java.net/)¹⁷ to leverage WS-Management profile. Ensure that the user you use to connect to remote Hyper-V server owns sufficient permissions to manage Hyper-V.

All network communications are made over HTTP or HTTPS. With Windows Server 2008 releases, the WS-Management implementation is installed by default, you can find it in `WINROOT/system32/winrm.exe`. To be able to use it, you just have to activate it. You can use the following tutorials:

- [http://msdn.microsoft.com/en-us/library/aa384372\(VS.85\).aspx](http://msdn.microsoft.com/en-us/library/aa384372(VS.85).aspx)
- <http://blogs.dirteam.com/blogs/sanderberkouwer/archive/2008/02/23/remotely-managing-your-server-core-using-winrm-and-winsr.aspx>

If you decide to use HTTPS, you may have to set some variables on the Java client side and to add the server's public key to one of your keystores:

- <http://java.sun.com/j2se/1.4.2/docs/tooldocs/windows/keytool.html>
- <http://java.sun.com/j2se/1.5.0/docs/guide/security/jsse/JSSERefGuide.html>

You may have noticed that this solution has a real advantage over the WBEM implementation, it is firewall friendly...

Here is an example showing how to connect to and manage your environment:

```
System.setProperty("javax.net.ssl.keyStore", "/home/user/.keystore");
System.setProperty("javax.net.ssl.trustStore", "/home/user/.keystore");
System.setProperty("javax.net.ssl.trustStorePassword", "keyStorePassword");
System.setProperty("javax.net.ssl.keyStorePassword", "keyStorePassword");
HyperVVMM vmm = new HyperVVMM("https://hyperv.address.com/
wsman", "hypervUser", "hypervUserPassword");
HyperVVM vm = vmm.getNewVM("Ubuntu");
HyperVVM clone = vm.clone("UbuntuClone");
clone.pushData("testVar", "testVal");
clone.powerOn();
//some work
System.out.println("test: " + clone.getData("testVar"));
clone.powerOff();
clone.destroy();
```

The four first lines allows HTTPS communication, the fifth line connects to the VMM thanks to `new HyperVVMM(String,String,String)`. The three parameters are: url, user with enough permissions and his password. For the url, don't forget the protocol and the trailing /wsman... Then you can either get all registered virtual machines thanks to `vmm.getVirtualMachines()` or get a given virtual machine with `vmm.getNewVM("machinesName")`. Before powering on your vm, you can pass it some data that you will be able to read from the inside:

```
myVM.pushData("myvar", "myval");
```

To get the information from the inside of the virtual machine, if guest tools are installed, read the registry key: `HKLM/SOFTWARE/Microsoft/Virtual Machine/External`. You also can use the method `HyperVVM#getKvpExchangeData("testVar")` to read this value. To set a variable from the inside of the virtual machine, if guest tools are installed, refer to: [this blog entry](#)¹⁸. and read it thanks to `HyperVVM#getGuestExchangeItem("testVar")`. If guest tools are not installed, use `HyperVVM#pushData` and `HyperVVM#getData`.

¹⁷ <https://wiseman.dev.java.net/>

¹⁸ <http://blogs.msdn.com/taylorb/archive/2008/07/07/hyper-v-wmi-kvp-exchange-aka-data-exchange-adding-new-items-from-guest.aspx>

pushData first tries to delete a previously set kvp with the same key and then call pushKvpExchangeData. getData first tries to getGuestExchangeItems, if this call fails, then tries getGuestExchangeItem.

2.1.2.4.2. WBEM Windows implementation (using WMI)

To be able to manage Microsoft Hyper-V with this implementation, you must put every jars located in src/hyperv-wmi/lib and src/core/lib in the class path. This module uses [J-Interop](http://www.j-interop.org/)¹⁹ to use DCOM communication protocol. Ensure that the user you use to connect to remote Hyper-V server owns sufficient permissions to manage Hyper-V.

To manage this tool, you must allow WMI/MSRPC communication in your firewall configuration. Administrative tools -> Component Services -> right click on My Computer Edit limits & default for Acces Permissions and Launch and Activation Permissions for the user you will use to connect to virtualization service. For firewall rules and remote access: <http://www.windowsreference.com/windows-server-2008/install-configure-hyper-v-in-windows-server-2008-64-bit-core-install/> With Hyper-V running on a Microsoft Server R2 edition you also must edit registry. Launch regedit.exe, select the KEY HKEY_CLASSES_ROOT\CLSID\{76A64158-CB41-11D1-8B02-00600806D9B6} (ie. WBEM Scripting Locator) The owner of this key is Trusted Installer which is a service with special permissions. To be able to use advanced scripting features using j-interop, we need full permissions for this KEY. The way to get that done is straight forward, we must change the owner of the key (right click on the directory, advanced, owner...) and allow full access to the new owner. An other alternatives may be available, see:

- <http://support.microsoft.com/kb/198891/en-us>
- <http://blog.tjtj.com/index.php/2007/02/regedit-cannot-create-value-error.html>
- <http://social.technet.microsoft.com/Forums/en/windowsserver2008r2management/thread/14b31477-ff1f-4f6b-818d-af6d675c3531>
- <http://sourceforge.net/projects/j-interop/forums/forum/840678#at>
- <http://www.wictorwilén.se/Post/Fix-the-SharePoint-DCOM-10016-error-on-Windows-Server-2008-R2.aspx>

Here is an example showing how to connect to and manage your environment:

```
HyperVMM vmm = new HyperVMM("hyperv.address.com", "user", "pwd");
HyperVMM vm = vmm.getNewVM("myUbuntu");
HyperVMM clone = vm.clone("myUbuntuClone");
clone.pushData("testVar", "testVal");
clone.powerOn();
//some work
System.out.println("test: " + clone.getData("testVar"));
clone.powerOff();
clone.destroy();
```

The first line connects to the VMM thanks to new HyperVMM(String,String,String). The tree parameters are: url, user with enough permissions and his password. Then you can either get all registered virtual machines thanks to vmm.getVirtualMachines() or get a given virtual machine with vmm.getNewVM("machinesName"). Before powering on your vm, you can pass it some data that you will be able to read from the inside:

```
myVM.pushData("myvar", "myval");
```

To get the information from the inside of the virtual machine, if guest tools are installed, read the registry key: HKLM\SOFTWARE\Microsoft\Virtual Machine\External. You also can use the method HyperVMM#getKvpExchangeData("testVar") to read this value. To set a variable from the inside of the virtual machine, if guest tools are installed, refer to: [this blog entry](#)²⁰. and read it thanks to HyperVMM#getGuestExchangeItem("testVar"). If guest tools are not installed, use HyperVMM#pushData and HyperVMM#getData.

¹⁹ <http://www.j-interop.org/>

²⁰ <http://blogs.msdn.com/taylorb/archive/2008/07/07/hyper-v-wmi-kvp-exchange-aka-data-exchange-adding-new-items-from-guest.aspx>

2.1.2.5. Xen Open Source, KVM, Qemu-KVM, Qemu - Libvirt

To be able to manage one of XenOSS, KVM, Qemu-KVM or Qemu, you must install [LibVirt](#)²¹ on the computer hosting the VMM. Put every jars located in `src/libvirt/lib` and `src/core/lib` in the class path. You need to set the `jna.library.path` variable when launching your java application (`-Djna.library.path=path-to-libvirt-lib`). The last requirement is due to the fact that Libvirt is a management API written in C and that we use [JNA](#)²² to use the C library from a Java environment. Package to use: `org.ow2.proactive.virtualizing.libvirt.*`. Refer to the Libvirt section to see how to use that API to manage your infrastructure and see the section tied to your VMM to gather extra information.

2.1.2.5.1. Libvirt

Libvirt (see <http://libvirt.org/>) is a management API project for virtualized infrastructures. Its aim is to provide an unique API / Interface to manage several VMM. You can notice that it currently supports: Xen (on Linux and Solaris), QEMU, KVM, LXC, OpenVZ, User Mode Linux, VirtualBox, and that other supports are announced (VMware server and ESX).

If you used Virt-manager to install and set up your virtualized environment, you must already have Libvirt installed on your computer and your virtual machines registered. Otherwise, download the project [here](#)²³ and install it. Then, you have to register your previously created virtual machines. You can read the associated [documentation](#)²⁴ to get it done.

When evaluating this environment, we noticed different "hot points" about which one you have to be careful. Libvirt may use [different mates](#)²⁵ for authentication, just be sure that you have enable the good user with sufficient permissions and using the good authentication mechanism. Be sure to understand the basic knowledge about [URLs](#)²⁶. Understand that, when using Xen Oss, you will have to perform extra configuration on the server side to set up your Xen VMM the good way (to use either sockets, http, legacy remote control ...).

Here is an example showing how to connect to and manage your environment:

```
LibvirtVMM vmm = new LibvirtVMM("qemu+ssh://psychoquack.inria.fr/system");
LibvirtVM myVM = vmm.getNewVM("machinesName");
myVM.powerOn();
//some work
myVM.powerOff();
```

The first line connects to the VMM thanks to `new LibvirtVMM(String)`. The String parameter is the url (see [URL](#)²⁷) used to connect to the Libvirt daemon. Then you can either get all registered virtual machines thanks to `vmm.getVirtualMachines()` or get a given virtual machine with `vmm.getNewVM("machinesName")`. With that implementation, you can only power the vm on and off and cannot pass information to it.

2.1.2.5.2. Xen Open Source

Xen OSS is the base component of XenServer. It is a type one hypervisor which comes as a particular kernel for several linux distribution (redhat, fedora, ubuntu, ...). The best way to have a fully personalized Xen is to compile sources and build its own kernel. If you want to use that software: Download the [source tarball](#)²⁸, CAREFULLY read the readme (you also can use [that tutorial](#)²⁹) and build your own kernel(s):

```
cd /usr/src
```

²¹ <http://libvirt.org/>

²² <https://jna.dev.java.net/>

²³ <http://libvirt.org/downloads.html>

²⁴ <http://libvirt.org/docs.html>

²⁵ <http://libvirt.org/auth.html>

²⁶ <http://libvirt.org/uri.html>

²⁷ <http://libvirt.org/uri.html>

²⁸ <http://www.xen.org/download/>

²⁹ http://www.howtoforge.org/debian_etch_xen_3.1

```
wget http://bits.xensource.com/oss-xen/release/3.3.1/xen-3.3.1.tar.gz
tar xvf xen-3.3.1.tar.gz
ln -s xen-3.3.1 xen
cd xen
```

Now we compile Xen with 2 targets kernels

```
make world vmxassist=y KERNELS="linux-2.6-xen0 linux-2.6-xenU"
install.sh
```

The `vmxassist=y` flag means that we want to allow HVM (Hardware-enhanced Virtual Machine, requires hardware support) guests to run, so we ask Xen to build a specific container that will be available in `/usr/lib/xen/boot/hvmloder`. To be able to compile such a Kernel, you must ensure that the packages `glibc-devel`, `dev86`, `libvncserver`, `SDL` and `SDL-devel` are installed. If you want, you can make them more efficient...

```
cd /usr/src/xen/build-linux-2.6.18.8-xen0_[arch]
make menuconfig
```

Check the following options:

```
Xen --> [*] Privileged Guest (domain 0)
File systems --> [*] Quota support
    [M] Old quota format support
    [M] Quota format v2 support
```

```
Device Drivers ---> Networking support --> [M] Dummy net driver support
```

```
Networking support --> Networking options --> [*] Network packet filtering (replaces ipchains) --> IP: Netfilter
Configuration --> [M] IP tables support (required for filtering/masq/NAT).
```

then:

```
make
make modules
make modules_intall
make install
```

Just repeat the whole procedure unchecking

```
Xen --> [ ] Privileged Guest (domain 0)
```

for the domU kernel. Note that you can forget that last build as the last releases of Xen OSS allows to run both host and guest with the same kernel config and the same efficiency. Just be sure that the modified `/boot/grub/menu.lst` is up to date and reboot on your newly created dom0 install.

Once your Xen Oss virtual environment is installed, you have to setup your virtual machines inside. To help you, you have several client software, among those one: Convirt (aka Xenman) and virt-manager. Our favorite one for the installation step is [Convirt](http://www.convirture.com/)³⁰. It

³⁰ <http://www.convirture.com/>

will allow you to install both paravirtualized and hardware assisted virtualized guest operating systems (the setup "by hand is feasible" for paravirtualized but more complicated for others and deals with Xen configuration files, thus we encourage to avoid it...). Here is one of [the good tutorial](#)³¹ you can find on the web. Virt-manager is better for virtual machine management and handles more virtual environments (furthermore it is built on top of [libvirt API](#)³²).

2.1.2.5.3. KVM, Qemu-KVM, Qemu

That hypervisors are all based on [Qemu](#)³³. They are all type two hypervisors that allow [emulation](#)³⁴ (Qemu) and [hardware assisted virtualization](#)³⁵ (Qemu-KVM and KVM). The best advantage of using that softwares (in fact, only Qemu is concerned) is that it is possible to use them entirely in user space (do not need root account for installation and use). If you want to use that solution, download one of the products (Qemu-KVM is an add-on for Qemu): [Qemu](#)³⁶ or [KVM](#)³⁷ and install it thanks to the README explanations.

To install virtual machines and manage your environment you can use [Convirt \(aka Xenman\)](#)³⁸ or [virt-manager](#)³⁹ or try to get it done by hand.

```
dd if=/dev/zero of=disk.img bs=512k count=1 seek=2000
mkfs.ext2 disk.img
```

build a raw disk image with a capacity of 1Go and an ext2 file system (for linux based systems).

```
kvm -net nic -net user -m 512 -boot d -cdrom /home/jmguilla/isos/fedora.iso disk.img
```

install your OS thanks to the .iso file

```
kvm -net nic -net user -m 512 -boot c disk.img
```

to boot your virtual machine once the operating system is installed.

2.2. Troubleshooting

This section describes the most common errors you can encounter.

- **Libvirt**

```
libvir: Remote error : socket closed unexpectedly
Exception in thread "main" org.ow2.proactive.virtualizing.core.error.VirtualServiceException:
Unable to connect to qemu+ssh:///system. Unable to connect to libvirt service: qemu+ssh:///system
```

Be sure that the libvirt daemon is started and to provide a good [libvirt url](#)⁴⁰

³¹ <http://blog.fedora-fr.org/smoothfrogz/post/XenMan>

³² <http://libvirt.org/>

³³ <http://www.nongnu.org/qemu/>

³⁴ <http://en.wikipedia.org/wiki/Emulator>

³⁵ http://en.wikipedia.org/wiki/Hardware-assisted_virtualization

³⁶ <http://www.nongnu.org/qemu/download.html>

³⁷ http://sourceforge.net/project/showfiles.php?group_id=180599&package_id=209008

³⁸ <http://www.convirture.com/>

³⁹ <http://virt-manager.et.redhat.com/download.html>

⁴⁰ <http://libvirt.org/uri.html>

- **VMware VI/Vix**

```
Exception in thread "main" org.ow2.proactive.virtualizing.core.error.VirtualServiceException:
Unable to connect to https://localhost:8333/sdk; nested exception is:
java.net.ConnectException: Connection refused
```

Be sure that the vmware service is started.

```
Exception in thread "main" org.ow2.proactive.virtualizing.core.error.VirtualServiceException: Unable to connect to
http://localhost:8333/sdk: Connection reset
Caused by: java.net.SocketException: Connection reset
```

Your service is up but you provided a bad url. Check protocol and port. By default for VMware server 2 https mapped to port 8333 and http to 8222.

```
Exception in thread "main" java.lang.UnsatisfiedLinkError: Unable to load library 'vixAllProducts':
/usr/lib/vmware-vix/lib/libvixAllProducts.so: wrong ELF class: ELFCLASS64
at com.sun.jna.NativeLibrary.loadLibrary(NativeLibrary.java:127)
```

Use a JRE/JDK that handles the native 32/64bits libraries.

When using vmware-vix to connect to VI compliant softwares, if you want to get a virtual machine you have to supply the vmx file path from a datastore root, for instance “[standard] myUbuntu/myUbuntu.vmx”, where [standard] is the datacenter's name.

```
- Unable to find the Domain [standard]myUbuntu/myUbuntu.vmx
- mess: The virtual machine cannot be found
Exception in thread "main" org.ow2.proactive.virtualizing.core.error.VirtualServiceException:
The machine [standard]myUbuntu/myUbuntu.vmx isn't registered for the current host.
```

For the example above, the good path is: “[standard] myUbuntu/myUbuntu.vmx” . Note the space (" ") between the datastore's name and the beginning of the path...