



Automating Management for Weblogic Server on JRockit-VE

Marius Sandu-Popa

[CERN openlab](#)

8 August 2010



oTN-2010-01

openlab Summer Student Report



Automating Management for WebLogic Server on JRockit-VE

Marius George, Sandu-Popa
Carlos Garcia, Fernandez
26 August 2010
Version 1.0

Distribution: **Public**

Abstract	2
1 Introduction	3
1.1 What Is WebLogic Server (WLS)?	3
1.2 What Is Virtualization?	3
1.3 Why Virtualization?	3
1.4 What is Oracle VM?	4
1.5 What is Jrockit-VE?	5
1.6 What is WebLogic on JRockit VE (WLS-VE)?	5
1.7 What is a WebLogic Server Domain?	6
2 Automating WLS-VE VM Management.....	6
2.1 Creating a Virtual Machine	6
2.2 Deleting a Virtual Machine	7
2.3 Modifying a Virtual Machine	7
2.4 WLS-VE Tool	8
3 Automating WLS-VE Domain Management	8
3.1 WLS Scripts	8
3.2 Extending and Integrating WLS Scripts to WLS-VE	9
4 Conclusions and Future Work.....	11
Acknowledgements	11
Appendix	12
A. WLS-VE Tool Command-Line Option Reference	12
A.1 -c (-create)	12
A.2 -d (-delete)	13
A.3 -f (-file)	13
A.4 -r (-reconfigure)	14
A.5 --reconfigure-service	15
B. WLS-VE Tool Complex Usage Examples	15
B.1 Configure SSH service	15
B.2 Configuring a Shared Disk on NFS to Store Server Log Files	16

Abstract

The CERN IT-DB department is looking for ways to migrate to the Oracle WebLogic Server on JRockit Virtual Edition(WLS-VE), which is a special version of Oracle WebLogic Server. The goal of this project was to implement tools for automating WLS-VE VM and domain management.



1 Introduction

Within the context of openlab, significant work on virtualization has been completed and two versions of Oracle VM (2.1.5 and 2.2) have been packaged and made available for automatic installation and central management. The development of these two versions, following input from CERN and initial openlab evaluations, has been crucial as it enables the use of Oracle VM in large-scale environments; the package can be installed and configured quickly and without human interaction.

The team has also participated in a series of tests looking at the Oracle WebLogic Server on JRockit Virtual Edition (WLS-VE), a special version of Oracle WebLogic Server. The benefits for the organisation are impressive as this solution significantly simplifies the maintenance of the middleware solutions and provides cost-effective scalability on demand as it runs without a guest Operating System.

The goal of this project was to provide an automated management for WLS-VE Virtual Machines and domains. For this we built several tools which will be discussed in the next sections. In the remainder of this section we will explain some of the technologies and concepts involved with Oracle WebLogic Server Virtual Edition.

1.1 What Is WebLogic Server (WLS)?

In the version 11gR1, Oracle WebLogic consists of a JavaEE platform product family that includes:

- a JavaEE application server, WebLogic Application Server
- an enterprise portal, WebLogic Portal
- an Enterprise Application Integration platform
- a transaction server and infrastructure, WebLogic Tuxedo
- a telecommunication platform, WebLogic Communication Platform
- an HTTP web server

The Oracle WebLogic Server is part of Oracle Fusion Middleware, which consists of a portfolio of software products from Oracle Corporation, including J2EE and developer tools, integration services, business intelligence, collaboration, and content management.

1.2 What Is Virtualization?

Virtualization is the ability to run multiple virtual machines on a single piece of hardware. The hardware runs software that enables you to install multiple operating systems capable of running simultaneously and independently, in their own secure environment, with minimal impact on performance. Each virtual machine has its own virtual CPU, network interfaces, storage, and operating system.

1.3 Why Virtualization?

With increased server provisioning in the datacenter, several factors play a role in stifling growth. Increased power and cooling costs, physical space constraints, man power, and interconnection complexity all contribute significantly to the costs and feasibility of continued expansion.

Commodity hardware manufacturers have begun to address some of these concerns by shifting their design goals. Rather than focusing solely on raw gigahertz performance, manufacturers have enhanced the feature sets of CPUs and chip sets to include lower wattage CPUs, multiple cores per CPU die, advanced



power management, and a range of virtualization features. By employing appropriate software to enable these features, several advantages are realized:

- **Server Consolidation:** By combining workloads from a number of physical hosts into a single host, a reduction in servers can be achieved as well as a corresponding decrease in interconnect hardware. Traditionally, these workloads would need to be specially crafted, partially isolated and well behaved, but with new virtualization techniques none of these requirements are necessary.
- **Reduction of Complexity:** Infrastructure costs are massively reduced by removing the need for physical hardware, and networking. Instead of having a large number of physical computers, all networked together, consuming power and administration costs, fewer computers can be used to achieve the same goal. Administration and physical setup is less time consuming and costly.
- **Isolation:** Virtual machines run in sand-boxed environments. They cannot access each other, so if one virtual machine performs poorly, or crashes, it does not affect any other virtual machine.
- **Platform Uniformity:** In a virtualized environment, a broad, heterogeneous array of hardware components is distilled into a uniform set of virtual devices presented to each guest operating system. This reduces the impact across the IT organization: from support, to documentation, to tools engineering.
- **Legacy Support:** With traditional bare-metal operating system installations, when the hardware vendor replaces a component of a system, the operating system vendor is required to make a corresponding change to enable the new hardware to function properly (for example, an ethernet card). As an operating system ages, the operating system vendor may no longer provide hardware enabling updates. In a virtualized operating system, the hardware remains constant for as long as the virtual environment is in place, regardless of any changes occurring in the real hardware, including full replacement.

1.4 What is Oracle VM?

Oracle VM is the server virtualization offering from Oracle Corporation. Oracle VM is based on the open-source Xen hypervisor technology supports both Windows and Linux guests and includes an integrated Web browser based management console. It consists of the following components:

- **Oracle VM Manager:** Provides the user interface, which is a standard ADF (Application Development Framework) web application, to manage Oracle VM Servers, virtual machines, and resources. Oracle VM Manager is used to:
 - Create virtual machines from installation media or from a virtual machine template
 - Delete virtual machines
 - Power off virtual machines
 - Import virtual machines
 - Deploy and clone virtual machines
 - Perform live migration of virtual machines
 - Import and manage ISO
 - Create and manage virtual machine templates
 - Create and manage shared virtual disks



- **Oracle VM Server:** A self-contained virtualization environment designed to provide a lightweight, secure, server-based platform to run virtual machines. Oracle VM Server is based upon an updated version of the underlying Xen hypervisor technology, and includes Oracle VM Agent.
- **Oracle VM Agent:** Installed with Oracle VM Server. It communicates with Oracle VM Manager to manage virtual machines.

1.5 What is JRockit-VE?

JRockit is a Java Virtual Machine (JVM) originally developed by Appeal Virtual Machines and acquired by BEA Systems in 2002, became part of Oracle Fusion Middleware in 2008.

The purpose of JRockit-VE, is to build a Java platform that is simpler and faster than running Java on a general purpose OS. JRockit-VE is the JRockit JVM extended so it can run directly on virtual hardware, and optimized for running Java on Oracle VM and x86 hardware.

In comparison to a Virtual Machine with Standard Guest OS, the JRockit-VE has the following advantages:

- Size changed from ~1GB to ~2 MB
- Improved performance
- Simplified configuration
- Increased security

1.6 What is WebLogic on JRockit VE (WLS-VE)?

WebLogic on JRockit VE is a virtualized version of WebLogic Server that is optimized to run on JRockit Virtual Edition. It consists of a virtual machine containing WLS and JRockit-VE.

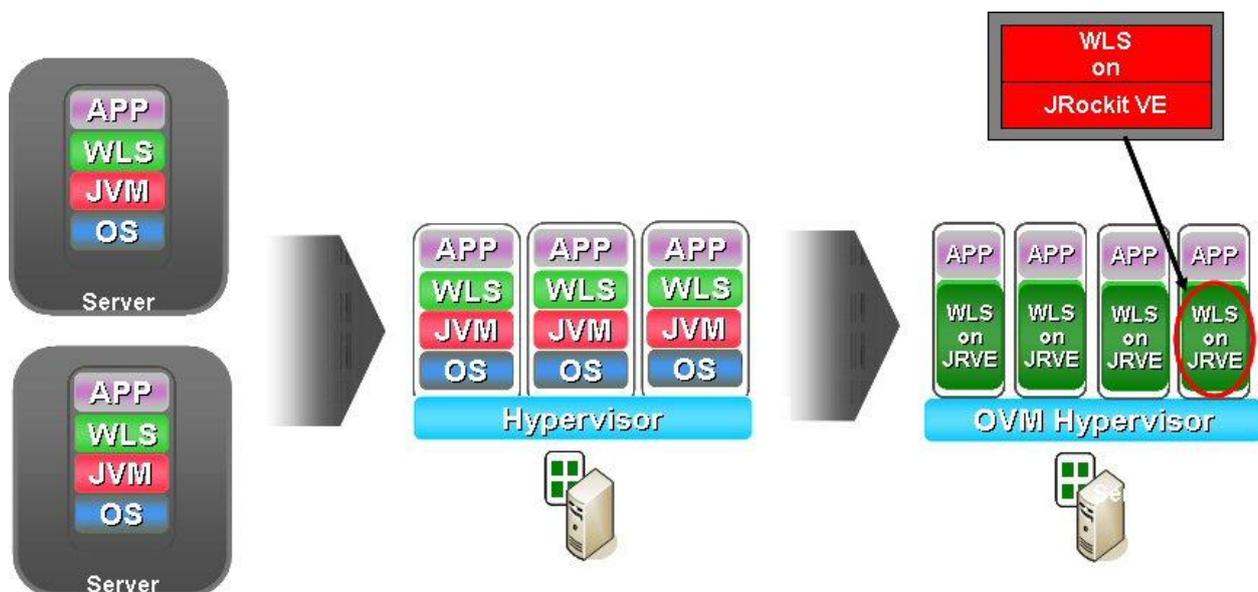


Figure 1: From WLS to WLS-VE



1.7 What is a WebLogic Server Domain?

A WebLogic Server administration domain is a logically related group of WebLogic Server resources. Domains include a special WebLogic Server instance called the Administration Server, which is the central point from which you configure and manage all resources in the domain. Usually, we configure a domain to include additional WebLogic Server instances called Managed Servers. We deploy Web applications, EJBs, and other resources onto the Managed Servers and use the Administration Server for configuration and management purposes only.

2 Automating WLS-VE VM Management

When we talk about WLS-VE VM management we refer to the following tasks:

- VM creation/deletion
- VM modification (of)
 - SSL certificates
 - Configuration parameters (ex: hardware, start-up parameters)
 - Filesystem
 - Services (ex: SSH)

In the following sections we will detail each task and show how we automated them.

2.1 Creating a Virtual Machine

Virtual Machines are created using the Oracle VM Manager (OVM). Depending on the available resources, there are several methods of creating a VM: templates, ISO files, using Preboot Execution Environment (PXE) over a network. We chose templates because WebLogic on JRockit VE bundles a standard VM image which we can easily import as a template in Oracle VM Manager.

Creating a VM from a template can be done using the Web Browser-based User Interface. This is an interactive process and cannot be automated. Fortunately, Oracle VM Manager also offers an experimental Command Line Interface which enables OVM commands to be scripted. We use this CLI to automate the creation process.

Besides the actual creation of the VM, we also have to register the MAC address of the VM into the CERN infrastructure (LanDB). This is done through perl scripts which use the NICE SOAP interface. Because the current version of the Oracle VM Manager Command-Line Interface does not allow users to specify MAC addresses we have to make an additional query to the OVM for the VM's MAC. This is also done with the CLI.

The final VM creation process looks like this:

1. Contact Oracle VM Manager via SSH and, using the CLI, create the VM.
2. Contact Oracle VM Manager via SSH and, using the CLI, get the MAC address of the newly created VM.
3. Register the VM in LanDB (using the SOAP interface) using the name of the VM and the MAC address from 2.



2.2 Deleting a Virtual Machine

The deletion process of a VM is similar to the creation process:

1. Contact the Oracle VM Manager and using the CLI delete the VM.
2. Unregister the VM from LanDB (using the SOAP interface).

2.3 Modifying a Virtual Machine

There are several ways to modify a virtual machine. One way would be to disassemble the VM, to update the *config.xml* file (a global configuration file for the VM), and assemble the VM back again. Although this process can be automated, it is very time consuming. We preferred to do it in another way.

WebLogic on JRockit VE includes an Image Tool (“*wlsveimagetool.jar*”) which allows us to reconfigure non-running virtual machines directly. With the Image Tool we can:

- Modify filesystem (copy files to/from the VM, delete files from VM, etc.)
- Reconfigure VM services (enable/disable SSH, add SSH keys/users, etc.)
- Reconfigure VM configuration settings (java-arguments for the WebLogic Server start-up command, NFS mount points, VM hardware parameters, etc.)

2.3.1 SSL Certificates Management

In addition to the modifications discussed above we also wanted the ability to import CERN SSL certificates into the virtual machines keystores. There are two types of keystores: identity keystore which holds the SSL certificate of the VM and the trust keystore which holds the Certificate Authorities (CA) SSL certificates.

For creating an identity keystore, we shutdown the virtual machine, and perform the following steps:

1. Generate private key and signed certificate (public key) for the VM using management scripts.
2. Create a chain of certificates with the VM signed certificate and the signing CAs certificates.
3. Create a new keystore from the key and the chain using WebLogic’s *ImportPrivateKey* utility.
4. Inject the newly created keystore into the VM using the Image Tool.

For CERN CA SSL certificates management we use the *cacerts* certificates file in the VM, which represents a system-wide trust keystore. The file is located in the “*/jrockitve/jrockit/jre/lib/security*” directory in the file system of the virtual machine image.

To add/remove CERN CA SSL certificates from the *cacerts* trust keystore, we shut down the virtual machine, and then perform the following steps:

1. Extract the *cacerts* file from the file system of the virtual machine to a local directory with the Image Tool.
2. Modify the *cacerts* file as required by using the java *keytool* utility.
3. Inject the modified *cacerts* file from the local directory to the file system of the virtual machine with the Image Tool.

2.4 WLS-VE Tool

All the functionality described in the sections above was bundled up in a bash script, which we refer to as WLS-VE tool. As shown in Figure 2, through the WLS-VE tool a management station issues create/delete commands to the Oracle VM Manager. The OVM then creates the VM from a registered template in an Oracle VM Server virtual machine pool. After the VM has been created, the management station uses the WLS-VE tool to modify the VM modify by connecting to the Oracle VM Server responsible for that VM.

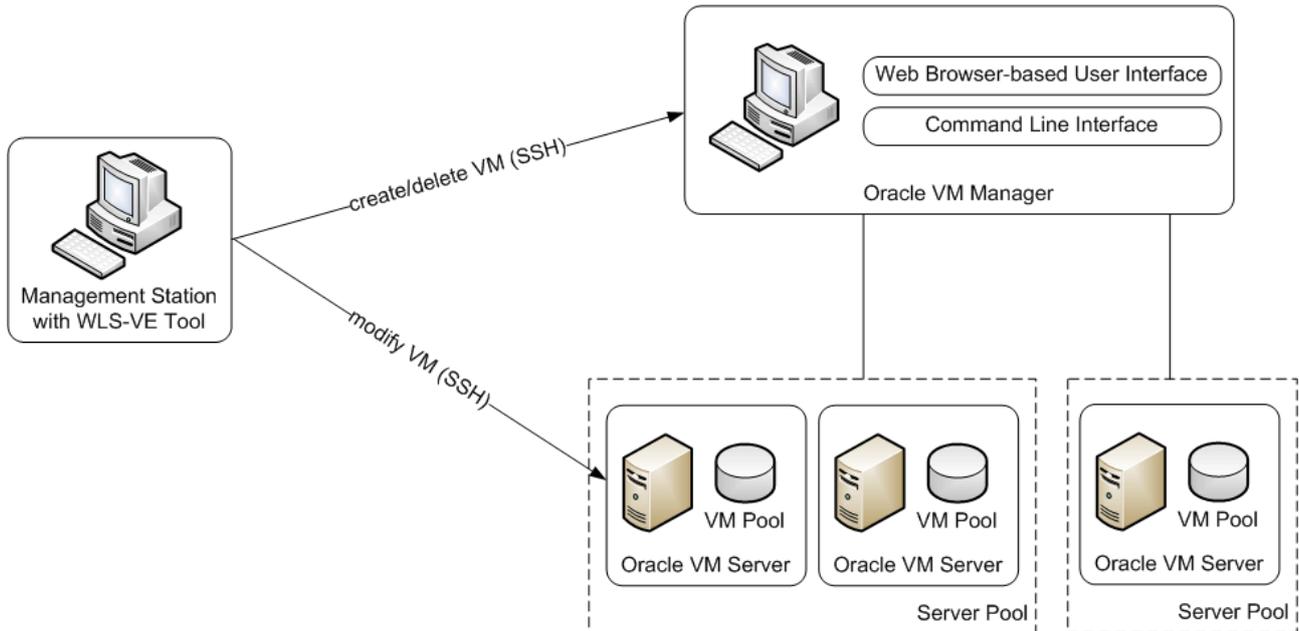


Figure 2: WLS-VE Tool Usage

For a detailed description of the WLS-VE tool and usage examples see [Appendix A](#).

3 Automating WLS-VE Domain Management

The second part of the project consisted in automating WLS-VE domain creation and configuration. We chose as starting point a set of scripts that automated management for the WLS. In the remainder of this section we will describe how we extended and integrated these scripts for the WLS-VE.

For a usage example of these scripts

3.1 WLS Scripts

This system of scripts consists of a set Jython scripts which perform domain configuration actions, some shell scripts which represent an intermediate layer between the Jython scripts and the shell scripts of Syscontrol (central management tool), and properties files which represent domain configuration.

The Jython scripts use the WebLogic Scripting Tool (WLST) which is a command-line scripting interface that system administrators and operators use to monitor and manage WebLogic Server instances and domains.

The properties files are divided into several categories:

- Domain: specifies domain properties



- Server: specifies Administration and Managed Servers properties
- Machine: specifies the physical machine that the server is running on and the Node Manager properties; (the Node Manager is a process agent running on the physical machine that hosts the servers; it is used for starting/stopping WebLogic Server instances).

A more detailed description of the scripts can be found at:

<https://twiki.cern.ch/twiki/bin/viewauth/DB/Private/DomainManagementScriptsAndSyscontrolIntegration>.

3.2 Extending and Integrating WLS Scripts to WLS-VE

3.2.1 Extending Properties Files

In WLS we have an association between a server and machine: this server instance is running on this physical machine. This is also illustrated by the WLS (server and machine) properties files. In WLS-VE we have two associations:

- between a VM and a server: this single server instance is running on this VM.
- between a VM and a Oracle VM Server Machine: this VM is running on this Oracle VM Server physical machine.

In order to express the associations above we added a *vmMachine* property in the Server properties files which specifies the name of the VM that the server is running on.

Another problem introduced by virtualization is that we cannot use Node Manager. The solution to this problem was to use Oracle VM Manager as Node Manager. All the properties from the machine properties file also apply to the OVM, so no change was necessary.

3.2.2 Creating a WLS-VE domain

To create a WLS domain a standard domain template is used. The template is unpacked and using WLST the following properties are set:

- server name, domain name, username, password
- listening address and port

After the modifications have been made the template is packed and uploaded to the physical machine that is hosting the WebLogic Server.

In WLS-VE, the VM image is practically the template. Therefore, we only need to update the domain properties for the VM. This can be achieved easily, by setting (with the WLS-VE tool) the appropriate java-arguments for the VM start-up command. These arguments have a different meaning for Administration and Managed Servers.

For a VM used as an Administration Server we have to set the following arguments:

- *-Dweblogic.Name*
- *-Dweblogic.Domain*
- *-Dweblogic.management.username*
- *-Dweblogic.managment.password*



- *-Dweblogic.ListeningAddress*
- *-Dweblogic.ListeningPort*

These arguments are used to specify: the server name, listening address and port and the name, username and password for the new domain. We also need to add the *-DWebLogic.management.GenerateDefaultConfig=true* argument to create the new domain upon booting the VM.

For a VM used as an Managed Server we set:

- *-Dweblogic.Name, -Dweblogic.Domain, -Dweblogic.ListeningAddress, -Dweblogic.ListeningPort* (as for the Administration Server)
- *-Dweblogic.management.username*
- *-Dweblogic.management.password*
- *-Dweblogic.management.server=admin.server.ip.address: admin.server.port*

where *admin.server.ip.address: admin.server.port* is the listening address, port of the Administration Server, and the *username* and *password* are the user name and password for the Administration Server.

3.2.3 Configuring a WLS-VE domain

The WLS scripts update the domain properties through managed beans (MBean) which are Java beans that provide a Java Management Extensions (JMX) interface. This also works with the WebLogic Server Virtual Edition, so no further modifications to the scripts are required.

3.2.4 Starting/Stopping a WLS-VE domain

The WLS scripts issue Node Manager commands to connect to the Node Manager and start/stop server instances. As shown above, the Node Manager process does not work for the WLS-VE virtual machines. Instead we replace the Node Manager with the Oracle VM. This changes the Node Manager WLST commands parameters as follows:

- `nmConnect()` command, used to connect to the Node Manager, is called with OVM properties (username, password, listening address) and the *VMM* Node Manager Type.
- `nmStart()`, `nmStop()` commands, used to connect to start/stop servers, are called with VM name instead of the Server name

A detailed syntax of these commands can be found at:

http://download.oracle.com/docs/cd/E13222_01/wls/docs90/config_scripting/reference.html#1030876 .

3.2.5 Deploying application

Deploying an application using the WLS scripts is a two-stage process:

- Copy the application to the physical machine
- Run the WLST deploy command which takes the application from the local filesystem and deploys it to the specified Server (cluster of Servers)

In WLS-VE we could insert the application into the VM using the WLS-VE tool and then run the deploy command. However there is an easier way to deploy an application: running the WLST deploy command with the *upload* flag set to true. When running the command like this the first step (copying the application) is done automatically.



4 Conclusions and Future Work

We have provided ways to fully automate WLS-VE VM creation/deletion and modification and we integrated this functionality into an easy to use command line tool(WLS-VE tool). We also managed to successfully extend and integrate WLS domain management scripts to WLS-VE.

As future work we can extend the WLS-VE tool with patching options. We can apply patches to WebLogic Server VMs by using a combination of SmartUpdate utility and WLS-VE Image Tool. The SmartUpdate utility offers a CLI, so the process can be fully automated.

Acknowledgements

I would like to thank the people in the IT-DB department, especially my supervisor Carlos, for making my stay at CERN a pleasant one.



Appendix

A. WLS-VE Tool Command-Line Option Reference

This appendix provides alphabetically sorted reference information about the command-line options that the WLS-VE Tool supports.

The WLS-VE Tool has the following syntax:

```
./wlsvetool -ov [OV] -vm [VM] [OPTIONS]
```

where [OV] is an Oracle VM Manager or an Oracle VM Server instance and [VM] is the name of the virtual machine.

The command-line options supported by the WLS-VE Tool are as follows:

-c (-create)	Creates a new VM and it registers the VM in LanDB
-d (-delete)	Deletes the specified VM and it unregisters the VM from LanDB
-f (-file)	Modify the VM filesystem
-h (-help)	Display help for the specified option
-r (-reconfigure)	View, add, change, and remove configuration settings for the specified VM
--reconfigure-service	Configure enabled services in the VM

Note: Before using any option of the WLS-VE Tool to view or modify the configuration or file system of a virtual machine, shut down the virtual machine.

Note: When specifying local directory paths, local file names, and Java arguments containing spaces, enclose the values in double quotation marks.

A.1 -c (-create)

The -c option enables you to create a VM using Oracle VM Manager and to register it into the LanDB.

Syntax:

```
./wlsvetool -ov [OVM] -vm [VM] -u [OVM_USER] -p [OVM_PASSWORD] -t  
[OVM_VM_TEMPLATE] -pool [OVM_VM_POOL]
```

Arguments:

OVM_USER	specifies Oracle VM Manager user name
OVM_PASSWORD	specifies Oracle VM Manager password
OVM_VM_TEMPLATE	specifies a Oracle VM Manager registered template to be used for the



	creation of a VM
OVN_VM_POOL	specifies the Oracle VM Manager Server Pool where the VM will actually be created

Example:

```
./wlsvetool -ov dbsrvg3304 -vm dbwslg008 -u admin -p oracle -t dbwslg000
-pool test2
```

The command creates a VM - with the name *dbwslg008*, from the *dbwslg000* template, in the *test2* server pool, using the OVM at *dbsrvg3304*- and registers it in LanDB.

A.2 -d (-delete)

The -d option enables you to delete a VM using Oracle VM Manager and to unregister it from the LanDB.

Syntax:

```
./wlsvetool -ov [OVM] -vm [VM] -u [OVM_USER] -p [OVM_PASSWORD] -pool
[OVN_VM_POOL]
```

Arguments:

Same as for the create option.

Example:

```
./wlsvetool -ov dbsrvg3304 -vm dbwslg008 -u admin -p oracle -pool test2
```

The command deletes a VM with the name *dbwslg008*, from the *test2* server pool using the OVM at *dbsrvg3304* and unregisters it from LanDB.

A.3 -f (-file)

The -f option enables you to perform the following tasks in the file system of a virtual machine image:

- Copy files from the file system of a virtual machine image to the local file system and vice versa.
- View directory listings.
- Move and rename files.
- Create and delete directories.
- Find files.
- Display the contents of text files.

Syntax:

```
./wlsvetool -ov [OVS] -vm [VM] -f [FILE_OPERATION] [PARAMETER]*
```

Operations and Parameters:



Same as “wlsvetool.jar “-f option. Please see:

http://download.oracle.com/docs/cd/E17090_01/doc.1111/e15206/commands005.htm#CIHGAFDC .

Example:

```
./wlsvetool -ov dbsrvg3309 -vm dbwlslg008 -f get /jrockitve/log/* /tmp
```

This command copies all the files in the */jrockitve/log* directory of VM *dbwlslg008* to the */tmp* directory of the local file system. The VM is located in the *dbsrvg3309* Oracle WebLogic Server virtual machine pool.

A.4 -r (-reconfigure)

The *-r* option enables you to view, add, change, and remove configuration settings for a virtual machine by modifying the virtual machine configuration directly.

Syntax:

```
wlsvetool -ov [OVS] -vm [VM] -r [CONFIG_OPTION] -action [ACTION]  
[CONFIG_OPTION_VALUE]*
```

Configuration options:

ssl	creates SSL identity keystore for the specified VM
cacerts	imports/delete/list CA certificates from VM cacerts trust keystore
Same as “wlsvetool.jar “-r option	See http://download.oracle.com/docs/cd/E17090_01/doc.1111/e15206/commands011.htm#CIHJADD

Examples:

1. Enable SSH

```
./wlsvetool -ov dbsrvg3309 -vm dbwlslg005 -r service -action enable sshd
```

2. Set java-arguments

```
./wlsvetool -ov dbsrvg3309 -vm dbwlslg005 -r java-arguments -action set  
“-Dweblogic.Name=WlsveAdmin -Dweblogic.Domain=wlsve_domain  
weblogic.Server “
```

3. Add NFS mount point

```
./wlsvetool -ov dbsrvg3309 -vm dbwlslg005 -r add mount nfs /home  
nfs.host.com /export/home uid=501 gid=502
```

This command adds an NFS mount point at the */home* directory within the virtual machine, pointing to the */export/home* directory on the *nfs.host.com* server, with 501 as the user ID and 502 as the group ID.

4. Modify VM cacerts trust keystore

```
./wlsvetool -ov dbsrvg3309 -vm dbwlslg005 -r cacerts -ca certrusted -  
action import
```



This command imports into the VM cacerts trust keystore, the SSL certificate of the CERN Trusted CA.

5. Create VM identity keystore

```
./wlsvetool -ov dbsrv3309 -vm dbwls005 -r ssl -ca CERN_CA -action new
./wlsvetool -ov dbsrv3309 -vm dbwls005 -r ssl -ca CERN_CA -action
install
```

The first command generates a new SSL certificate for dbwls005 signed by the CERN CA. The second command builds a keystore from the generated SSL certificate and installs it in the VM.

A.5 --reconfigure-service

The --reconfigure-service option can be used to configure services that are enabled in the virtual machine.

Syntax:

```
./wlsvetool -ov [OVS] -vm [VM] --reconfigure-service [SERVICE]
[CONFIG_OPTION] -action [ACTION] [CONFIG_OPTION_VALUE]*
```

Configuration options:

Same as “wlsvetool.jar” --reconfigure-service option. Please see:

http://download.oracle.com/docs/cd/E17090_01/doc.1111/e15206/commands012.htm#CBJIACI.

Example:

```
./wlsvetool -ov dbsrv3309 -vm dbwls005 --reconfigure-service sshd key -
action add /home/marius/.ssh/id_rsa.pub
```

The command adds the */home/marius/.ssh/id_rsa.pub* SSH public key to the sshd service.

B. WLS-VE Tool Complex Usage Examples

Note: For all examples in this section we use a VM named *dbwls005* which belongs to an Oracle VM Server instance installed at *dbsrv3309*.

B.1 Configure SSH service

In this example we enable the SSH service for the VM and we add the */home/marius/.ssh/id_rsa.pub* public key.

1. Enable SSH service

```
./wlsvetool -ov dbsrv3309 -vm dbwls005 -r service -action enable sshd
```

2. Add a SSH public key to ssh service

```
./wlsvetool -ov dbsrv3309 -vm dbwls005 --reconfigure-service sshd key -
action add /home/marius/.ssh/id_rsa.pub
```

3. List available SSH keys



```
./wlsvetool -ov dbsrvg3309 -vm dbwlsq005 --reconfigure-service sshd keys  
-action get
```

B.2 Configuring a Shared Disk on NFS to Store Server Log Files

We can use the Network File System (NFS) to automount and transparently access a read-only shared disk on the network. This way, we can output the WebLogic Server logs to the shared disk to save disk space on the VMs.

In this example, the logs directory is created in the NFS Server location (/scratch/userdir/nfs/) for the mounting point in the VM /application/user_projects/domains/wlsve_domain/servers/WlsveAdmin/logs.

1. Create a /logs directory in the NFS location.
2. Create the mounting point in the VM

```
./wlsvetool -ov dbsrvg3309 -vm dbwlsq005 -r vm.cfg add mount nfs  
/application/user_projects/domains/wlsve_domain/servers/WlsveAdmin/logs/  
/scratch/userdir/nfs/logs uid=11735 gid=900
```

After starting the VM, the logs are created under the NFS Server shared location /scratch/userdir/nfs/logs.