
VML 0.3 COMMANDS (10/13/08)

NAME

vml – Virtual Machine Logbook

SYNOPSIS

vml.py [**vml_options**] **vml_command** [**cmd_options**] [**cmd_arguments**]

VML OPTIONS

-H [**--help**]

Display usage information about VML

--help-commands

Display usage information about the VML commands

VML COMMANDS

Here are brief descriptions of the VML commands:

consult [**entry_name**]

Display the content of the repository. If an **entry_name** is specified, display all the information about this entry.

```
example% vml.py consult
example% vml.py consult entry1
```

save [**options**] **source** **entry_name** [**repository_name**]

Save a source into the repository by creating a new entry. If **repository_name** is given, VML will save the entry in this repository, otherwise the default one will be used. The source can be the path to a virtual machine configuration file, a working area name or the name of an entry that's running into a working area. The entry created will be saved with the name 'entry_name'.

Options:

-l LOG, --log= LOG	Allow to save a log message with the entry
-i , --incremental	Specify the incremental backup mode (this is the default)
-d , --differential	Specify the differential backup mode
-f , --full	Specify the full backup mode
-e , --environment	Specify the environment backup mode (not implemented)

The options 'idfe' are mutually exclusive

```
example% vml.py save /vm/cernVM_07.vmx my_experiment
example% vml.py save -d entry1 my_exp2 -l "my log"
example% vml.py save -f wa1 entry1 -l "my log message"
```



open entry [working_area]

open the entry in the working area specified. If no working area is given, open the entry in the default working area.

```
example% vml.py open atlas_exp3
example% vml.py open atlas_exp2 wa1
```

delete entries...

Remove from the repository the specified entries.

```
example% vml.py delete experience_080808
example% vml.py delete exp_cernVM_07 exp_cernVM_08
```

export [-f,--full] entry destination_path

Export the entry as a tarball into the destination_path. If the option -f is given, VML will export the entry with all its dependency in order to make the entry self-contained.

```
example% vml.py export experience_080808 $HOME
example% vml.py export -f entry3 /tmp
```

import [options] source_file entry_name [repository_name]

Import the entry represented by the source_file into the repository given by repository_name. If no repository is specified, VML imports the entry into the default one. Source_file can be either in the vml exported format '.vml' or a tarball '.tar.gz' who contains a full virtual machine.

The options are:

```
-o WA, --open=WA    after importation, open the entry in working area WA
-l LOG, --log=LOG   add a log message to the imported entry
```

```
example% vml.py import $HOME/atlasentry0.9.vml my_exp1 -l "log msg"
example% vml.py import -o wa1 /tmp/atlasExp.tar.gz my_exp2
```

wadd [-D, --Default] working_area_name working_area_path

Add a new working area to VML. Use the flag -D to set it as default.

```
example% vml.py wadd wa1 /tmp/wa1
example% vml.py wadd -D wa2 /tmp/wa2
```

radd [-D, --Default] repository_name repository_path

Add a new repository to VML. Use the flag -D to set it as default.

```
example% vml.py radd rep1 /tmp/rep1
example% vml.py radd -D rep2 /tmp/rep2
```

wdelete working_area_name

Delete the folder of the working area 'working_area_name' from the file system.

```
example% vml.py wdelete wa1
```

rdelete repository_name



Delete the folder of the repository 'repository_name' from the file system.

```
example% vml.py rdelete rep1
```

wdefault working_area_name

Set the working area 'working_area_name' as the default working area.

```
example% vml.py wdefault wa1
```

rdefault repository_name

Set the repository 'repository_name' as the default the working area.

```
example% vml.py rdefault rep1
```

setup

Start the setup wizard of VML for the working area and repository configuration

```
example% vml.py setup
```

Next features (not yet implemented in VML 0.3):

get-cernvm [options] destination_path

Download the latest CernVM release to the destination_path. The options permitted with 'vml get-cernvm' are:

- i NAME , --import=NAME
when downloading is done, import the entry into the local repository with the given name.
- o NAME, --open=NAME, -io NAME
when downloading is done, import the entry into the local repository with the given name and then open it.
- v VERSION, --version=VERSION
VML will download the CernVM according to the version specified

```
example% vml.py get-cernvm /var/vmware/vm
```

```
example% vml.py get-cernvm -v 0.7 -o my_experience /var/vmware/vm
```

get [options] -r source_file destination_path

Download the entry from a remote repository. The options permitted with 'vml get' are:

- i NAME, --import= NAME
when downloading is done, import the entry into the local repository with the given name.
- o NAME, --open= NAME, -io NAME
when downloading is done, import the entry into the local repository with the given name and then open it.

```
example% vml.py get - electra.lbl.gov/exp_080808.tar.gz $HOME/VMs
```

```
example% vml.py get -o cern08 -r electra.lbl.gov/CernVM_08.tar.gz $HOME
```