

ctys-createConfVM

Creation and Configuration of VMs

for

QEMU/KVM, VBOX, VMW, and XEN

November 29, 2010

Contents

1 NAME	2
2 SYNTAX	2
3 DESCRIPTION	2
4 OPTIONS	6
5 ARGUMENTS	8
6 ENVIRONMENT	8
7 EXIT-VALUES	9
8 SEE ALSO	10
9 AUTHOR	10
10 COPYRIGHT	10

1 NAME

ctys-createConfVM - Creation and Configuration of VMs

2 SYNTAX

ctys-createConfVM

```
[--auto-all]
[--auto]
[-C]
[--create-image]
[-d <debug-level>]
[-D <directory>]
[--defaults-file=<file-name>]
[--defaults-file <file-name>]
[--expert]
[-h]
[-H <help-options>]
[--label=<label>]
[--label <label>]
[--list-env-var-options|--levo]
[--no-create-image]
[--no-save-para-kernel]
[--save-para-kernel]
[-t <session-type>]
[-V]
[-X]
```

3 DESCRIPTION

ctys-createConfVM provides the initial creation of complete VMs for various open source hypervisors. For present VMs including several commercial products the post-creation of configuration files and wrapper scripts is supported. The whole process could be automated by pre-configured specific configuration defaults for supported distributions. This provides for simplified installation and creation of VMs with seamless integration into the inventory management, automation facilities and interactive graphical frontends provided by the UnifiedSessionsManager framework.

The **ctys-createConfVM** could be either applied in the interactive dialogue mode, or by semi and full automatic batch processing based on pre-defined default values. Beneath the pure creation of the virtual machines the automation of the guest operating system is supported for various distributions. Therefore offline installation of final images as well as batch startups of guest systems by means of the installed operating system are supported. The emphasis for the current version is on opensource systems, but is going to be extended as a common generic wrapper including additional commercial products.

The current version provides the following support:

- Creation of configuration file and wrapper script for the integration of present VMs:
This is provided for all supported systems - **QEMU/KVM** , **VBOX** , **VMW** , and **XEN** , either open source, or commercial closed source products. The actual installation of the virtual machine and the included operating system has to be proceeded by the provided native facilities.
- Creation of integration files and additionally the raw VM:
This is provided for the hypervisors **QEMU/KVM** , and **XEN** . The support for VirtualBox(TM) and VMware(TM) products by native interfaces is planned. For this variant the complete installation and

configuration of the virtual machine is provided by the UnifiedSessionsManager, whereas the operating system has to be installed by the provided native means.

- Complete turn-key installation of VMs including the guest OS:
This is provided for the **QEMU/KVM**, and **XEN** in combination with appropriate operating systems. The installation procedure is divided into two stages.
 1. Creation and configuration of the raw VM
 2. Installation of the contained OS

The installation of the guest OS varies due to the provided means by the guest OS and distribution. The following cases are available in the current version:

1. Interactive installation
The interactive installation of the guest OS could be started by various provided types of install sources. This comprises attached bootable fixed media based installation such as CD/DVD including converted file-images, installation from USB devices and media cards, and network based installation by PXE.
2. Kickstart
A template with basic configuration for CentOS and similar OS which could be customized as required. This could be combined with the installation sources.
3. Debootstrap
Debian based offline installation pattern. Generated as a wrapper script.
The extension for the VirtualBox(TM) and VMware(TM) products by provided interfaces is planned.

The following table shows a sumup of available install automation for the current version, where **k** marks support by external kernel parameters - available for Linux - and **g** marks the required configuration of the GuestOS, e.g. by boot of installers.

Hypervisor/Emulator	Configuration	VM	GuestOS
QEMU/KVM	x	x	k,g
VBOX	x		g
VMW	x		g
VMWE - VMware ESX(TM)			
VMWEi - VMware ESXi(TM)			
XEN	x	x	k,g
XENS - XenServer(TM)			

Table 1: Installation Automation

Following install modes for automated install procedures could be utilized.

- **g**: guest OS based
- **k**: kernel based

The available call options are partly set by the commandline options, but could the majority of the values are used by pre-assigned environment variables. The actual available variables with their initial value could be listed by the option '-list-env-var-options' or for short '-levo', these are listed within the following chapter **ENVIRONMENT**. The Syntax for the call is

```
<VAR-NAME>=<VAR-VALUE> ctys-createConfVM ....
```

For example

```
ACCELERATOR=KVM \
DIST=Scientific \
RELEASE=5.4.1 \
OS=Linux \
```

```
OSVERSION=2.6.18 \
ctys-createConfVM -t qemu --label=tst213 --auto-all --no-create-image
```

The created files comprise a generic wrapper-script and a configuration file, where the wrapperscript presents the generic facilities for application on all provided hypervisors. These are partly adapted to the targeted hypervisor during creation and marked with a version string for later compatibility validation. The configuration script contains the specific variables and parameters for the wrapperscript. Both files are also part of the overall ctys-framework as they implement the final stage of the actual execution for the related plugin. The execution of **ctys-createConfVM** could be performed either **locally** or on **remote**. The resulting configuration could be executed by the **ctys-wrapper** script either by call of **ctys - locally** or **remote** - or **ctys-wrapper CLI - locally**, where both calls could be transformed to remote relays/hosts by ctys-beamer.

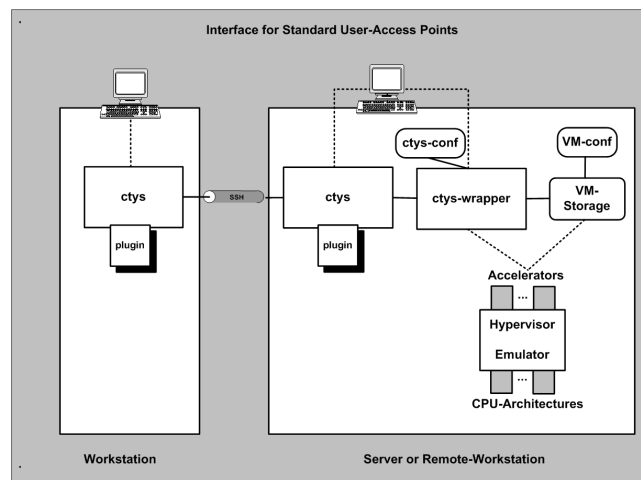


Figure 1: ctys distributed access

Both files are inline documented bash syntax files - with additional Python modules for Xen, implementing a medium level of configuration and workflow complexity and could be extended as required. The base call interface consisting of the call parameters

- bootmode
- check
- console
- print
- vncaccessdisplay

is mandatory and therefore required to be present for proper ctys operations. The amount of supported values could be adapted as required and varied by the user. The following values are required to be implemented mandatory:

```
--bootmode=HDD
--initonly
--instmode=(CD|PXE)
--console={VNC,CLI}
--vncaccessdisplay=<#nr>
--print
--check
```

The wrapper script is used for internal startup calls, but could be used particularly for tests from the command line. The following options support the initial runtime setup:

```
--listoptions
--print
--check
```

Particularly the debugging option '**-d printfinal**' provides the display of the final call assembly of each system call for debugging purposes.

The **WRAPPER-SCRIPT-OPTIONS** are:

-bootmode=<boot-modes>

The bootmode for now implements the following options

- **INSTALL**

This installs from media accessible within the file system. frequently this is a iso-image of s DVD.

- **PXE**

This installs from network and requires some ser setup procedures. The new versions of QEMU support this from the box, older versions might require specific BIOS related options.

- **VHDD**

The standard boot from a virtual HDD. By default only one boot media is foreseen, additonal data devices are prepared within the configuration file.

-check

This option forces a check only, thus simply the calloptions fare assembled and the final call is suppressed. In case of INSTALL and PXE, where intermediate steps for creation of the virtual HDDs are required these are actually suppressed too.

When called combined with **-print** option the final assembled call is displayed.

-console=<help-option>

This defined the CONSOLE. This is by default one of CLI0, CLI, EMACSM, EMACS, EMACSAM, EMACSA, GTERM, NONE, SDL, VNC, X11, or XTERM. The VNC type is implicitly created in any case by the standard call, allowing later decision.

-initmode

The virtual storage devices are created and initalized, else the configuration and wrapper files are created only.

-initmodeonly

The virtual storage devices are created and initalized only.

-instmode[=<src-mode>%<src-dev>%<target-mode>%<target-dev>%<free-custom-def>]

The installation mode actually creates the virtual bootable devices. This is in the most cases proceeded by booting the GuestOS into an install mode, where the provided mechanisms may be applied. The majority of OS provides CD/DVD-boot which could be performed by an so called ISO-image. Some OS has to be installed by specific variants only, such as PXE boot, when e.g. other variants are faulty.

Debian provides by **debootstrap** a method for offline installation particularly suitable for paravirtualization. This method is supported within ctys by a specific wrapper script, which is generated during creation and stored within the created directory of the VM.

The usage of the keyword **-instmode** without suboptions implies the default settings for all suboptions. When for specific subtions the preconfigured defaults should be applied the keyword 'default' could be used.

-listoptions

This call lsits the actual available options of the user QEMUBASE/QEMUKVM call.

-print

Prints some information and the final exec call with the whole set of assembled parameters.

-vncaccessdisplay=<#nr>

The display number for VNC access. The runtime system manages these by a dynamic algorithm in mixed and distributed environments. The value for the local commandline call should consider free ports, no interference to the ctys framework should appear.

4 OPTIONS

-auto

Forces all answers to be 'y', excluding mandatory parameters, which have to be answered by a manual confirmation.

-auto-all

Forces all answers to be 'y', including mandatory parameters, which have else to be answered by a manual confirmation.

-c

Forces the creation of a directory when **-D** is selected and the target directory is not yet present. For safety reasons when operating on remote hosts, only one level of subdirectories could be created, for multiple level creation use "mkdir -p".

-d <debug-level>

Same as ctys.

-D <directory>

Directory for storage of created configuration files.

-defaults-file <filename>

Filename for an alternative defaults file.

-expert

Expert mode, which reduces the displayed text, and shows some additional technical details.

-h|-help|-help

Print help, refer to "-H" for additional information.

-H <help-option>

The extended help option is based on system interfaces for display of manpages, PDF and HTML documents. This comprises the man pages and installed manuals.

For additional help refer to the documents or type **ctys -H help**.

-label=<label-for-vm>

The label - the name - for the VM to be created.

-list-env-var-options|-levo

List of current options to be set by environment variables. The values are the initially assigned static values, which partly are completed at runtime dynamically.

The actually resulting options, including a list of all valid environment variables is displayed as given in the following example figure. The '(h)' marked values sign the by default taken values from the host machine:

```
-bash-3.2$ ctys-createConfVM --label=testDisplay --levo
```

Not all values require to be set, some will be requested later by

dialogue.

Thus it is not necessary to have values assigned to the complete displayed set.

Actually used sources for default values:

- no-marker = Pre-Set value, either from defaults configuration, or by commandline.
- no-value = Either requested by dialog later, or the defaults of the finally called application are used.
- (g) = Dynamically generated.
- (c) = Read from actual configuration file, e.g. vmx-file.
- (h) = Used from current host as default.

Applicable modifications:

- blue = By call option, defines dependency for others.
- green = By environment, 'could be set almost independent' from other values.
- cyan = By miscellaneous facilities, but is dependent from others. E.g. LABEL defines by convention the network 'hostname', thus the TCP/IP params. This could ..., but should not be altered!

Most of the missing values will be fetched during actual execution of this tool by dynamic evaluation.

VAR name:Initial Value

```
C_SESSIONTYPE:QEMU
  LABEL:testDisplay
    MAC:
    IP:
  BRIDGE:
  DHCP:
  NETMASK:
    TCP:
  GATEWAY:

  EDITOR:acue

  UUID:f58abb0f-f63b-4c46-aaf1-4421e1e21b43 (h)

  DIST:CentOS (h)
  DISTREL:5.4 (h)
    OS:Linux (h)
  OSREL:2.6.32.6-app2-003-kvm (h)

  ARCH:x86_64 (h)
  ACCELERATOR:QEMU
    SMP:
  MEMSIZE:512
  KBD_LAYOUT:de

  STARTERCALL:/usr/libexec/qemu-kvm
  WRAPPERCALL:

  DEFAULTBOOTMODE:HDD
```

```

        DEFAULTINSTTARGET:/hda.img
        HDDBOOTIMAGE_INST_SIZE:8G
        HDDBOOTIMAGE_INST_BLOCKSIZE:256M
        HDDBOOTIMAGE_INST_BLOCKCOUNT:32
        HDDBOOTIMAGE_INST_BALLOON:y

        DEFAULTINSTMODE:CD
        INSTSRCCDROM:/dev/cdrom
        DEFAULTINSTSOURCE:/dev/cdrom
        INST_KERNEL:
        INST_INITRD:

        VMSTATE:ACTIVE

```

Remember that this is a draft pre-display of current defaults.
 No consistency-checks for provided values are performed at
 this stage.
 Some missing values are evaluated at a later stage dynamically.

-bash-3.2\$

-no-create-image

Suppresses the automatic creation of a virtual file image for installation.

-no-save-para-kernel

Suppresses the automatic storage of runtime kernels when these are selected. This is particularly frequently the case for paravirtualized DomU of Xen and in some cases of QEMU.

-no-save-para-kernel

Forces the automatic storage of runtime kernels when these are selected. This is particularly the case for HVM DomU of Xen, or for some OSs like CentOS, where this is not necessarily required.

-t <session-type>

The session type the VM has to be created for. Currently **QEMU** (for QEMU and KVM on x86), **VMW**, **VBOX**, and **XEN** are supported. Each defines a common set of parameters with additional specific variations.

-V

Version.

-X

Terse.

5 ARGUMENTS

None.

6 ENVIRONMENT

In addition to the commandline arguments the following environment variables provided. These could be listed by the **-levo** option and set by the call pattern:

```
IP=111.111.111.111 ctys-createconfVM -t Xen --label=myXenPattern
```

The current available variables are listed within the following table.

VAR name	Description	Default
ACCELERATOR	Available accelerator module.	Host OS
ARCH	Architecture for guest OS.	Current host.
BRIDGE	Host bridge.	(config)
C_SESSIONTYPE	As selected by -t.	-t
DEFAULTBOOTMODE	Default boot mode.	VHDD
DEFAULTCONSOLE	Default console.	config
DEFAULTHOSTS	Default HOSTs for GuestOS.	config
DEFAULTINSTMODE	Default install mode.	config
DEFAULTINSTSOURCE	Default install source.	config
DEFAULTINSTTARGET	Default install target.	config
DHCP	Use of DHCP for Guest OS.	(config)
DIST	Distribution of GuestOS.	Host OS.
DISTREL	Distribution release of GuestOS	Host OS.
EDITOR	Author of installation.	\$USER
GATEWAY	TCP gateway within GuestOS.	(config)
HDDBOOTIMAGE_INST_BALLOON	Expand on demand.	config
HDDBOOTIMAGE_INST_BLOCKCOUNT	Number of blocks.	config
HDDBOOTIMAGE_INST_BLOCKSIZE	Size of chunks.	config
HDDBOOTIMAGE_INST_SIZE	Sum of chunks.	config
INST_INITRD	Installation initrd image.	config
INST_KERNEL	Installation kernel.	config
INSTSRCCDROM	installation ISO image/media.	config
IP	IP address of GuestOS.	cacheDB
KBD_LAYOUT	Keaypoard laout.	config
LABEL	Name of VM, GuestOS hostname.	-label
MAC	MAC address of VM.	cacheDB
MEMSIZE	Assigned RAM.	config
NETMASK	Netmask of GuestOS TCP address.	(config)
OS	Operating system of GuestOS.	Host OS
OSREL	Release of GuestOS.	Host OS
SMP	Number of assigned CPU cores.	1
STARTERCALL	Hypervisor start-caller.	local hypervisor
TCP		cacheDB
UUID	UUID.	VM or dynamic
VMSTATE	Inventory state.	ACTIVE
WRAPPERCALL	System wrapper for hypervisor.	config

7 EXIT-VALUES

0: OK: Result is valid.

1: NOK: Erroneous parameters.

2: NOK: Missing an environment element like files or databases.

8 SEE ALSO

ctys executables

ctys plugins PMs

ctys-PM(1)

VMs

KVM(1) , *ctys-QEMU(1)* , *ctys-VBOX(1)* , *ctys-VMW(1)* , *ctys-XEN(1)*

HOSTS

ctys-CLI(1) , *ctys-RDP(1)* , *ctys-VNC(1)* , *ctys-X11(1)*

9 AUTHOR

Maintenance: <acue_sf1@sourceforge.net>
Homepage: <<http://www.UnifiedSessionsManager.org>>
Sourceforge.net: <<http://sourceforge.net/projects/ctys>>
Berlios.de: <<http://ctys.berlios.de>>
Commercial: <<http://www.i4p.com>>



10 COPYRIGHT

Copyright (C) 2008, 2009, 2010 Ingenieurbuero Arno-Can Uestuensoez

This is software and documentation from **BASE** package,

- for software see GPL3 for license conditions,
- for documents see GFDL-1.3 with invariant sections for license conditions.
The whole document - all sections - is/are defined as invariant.

For additional information refer to enclosed Releasenotes and License files.

