

# ctys-QEMU(1) QEMU/KVM Interface

March 10, 2011

## Contents

|          |                      |          |
|----------|----------------------|----------|
| <b>1</b> | <b>NAME</b>          | <b>2</b> |
| <b>2</b> | <b>SYNTAX</b>        | <b>2</b> |
| <b>3</b> | <b>DESCRIPTION</b>   | <b>2</b> |
| <b>4</b> | <b>OPTIONS</b>       | <b>2</b> |
| <b>5</b> | <b>PREREQUISITES</b> | <b>5</b> |
| <b>6</b> | <b>SEE ALSO</b>      | <b>6</b> |
| <b>7</b> | <b>AUTHOR</b>        | <b>6</b> |
| <b>8</b> | <b>COPYRIGHT</b>     | <b>6</b> |

## 1 NAME

ctys-QEMU - QEMU(TM) Interface - QEMU+KVM

## 2 SYNTAX

```
ctys -t QEMU -a action[=<suboptions>] ...
ctys -T QEMU -a action[=<suboptions>] ...
ctys -T ALL -a action[=<suboptions>] ...
```

## 3 DESCRIPTION

The **QEMU** plugin adds support for sessions to **VirtualMachines** of type **QEMU(TM)** with optional **KVM** and **KQEMU** based accelerators. The **KVM** accelerator is used implicitly when present and activated. Therefore multiple sets of executables are scanned during initialisation of each startup and set appropriately.

The management of the virtual machines includes the support of boot and shutdown for local and remote virtual machines. This is either executed by filesystem search of configuration files, or by the use of a pre-scanned inventory databases containing accessible machines. Various options control the name service and possible caching of information, particularly the option **-c** for the control of the location of the nameservice cache data. Virtual machines stored on network filesystems could be optionally executed on multiple worker-machines, providing a multipath-registration within the inventory. The inventory is populated automatically by scanning the local and remote filesystems on each participating worker-machine and collecting the information of detected VMs into the inventory database. The inventory database is currently an ASCII based file-database containing records separated by <CR>/lines, and semicolon separated fields.

The access to VMs is supported by the full scope of the <**machine-address**>, thus e.g. a user defined LABEL as a shortcut-alias could be used. The QEMU plugin is "**VM-STACK safe**" as it handles running embedded virtual machines and emulators by propagating state change events(e.g. SHUTDOWN) of the execution platform to contained hypervisors and emulators.

The supported versions are automatically detected, thus no further parameters for version distinction are required. KVM is supported within the QEMU plugin only. Thus KVM could be enabled within a native QEMU installation and as a separate installation. The various **KVM** versions and builds specific to distributions with their various available option sets are detected automatically. The actual option sets are identified and dynamically pre-set for the usage of appropriate values.

Additional information for installation is available from **ctys-configuration-QEMU**, information containing use-cases with application examples is available from **ctys-uc-QEMU**.

## 4 OPTIONS

**-a** <**action**>[=<**action-suboptions**>]

For the complete Syntax refer to the depicted generic superset within the call-framework ctys(1) .

**-a CANCEL**

Specific semantics for SPORT within QEMUMONSOCK as described before.

The CONSOLE type CLI0 requires specific handling for CANCEL.

**-a CREATE**

All standard parameters not listed here could be applied.

Additional arguments to be bypassed to the qemu executable. The application targets are:

- C: Client(not yet supported)
- S: Server, is applied to the wrapper script as the '-argsadd=<ARGSADD>' argument.
- B: Both(not yet supported)

**BOOTMODE:<MODE>[%<PATH>]**

For **BOOTMODE** multiple configuration variants are supported. In any case only one of them is supported to be the active **master** at any time. This is controlled by "MAGICID-QEMU" and "MAGICID-IGNORE" , which influences the inventory scanning process of **ENUMERATE**, and therefore the caching behaviour too. The following combinations of configuration file and Wrapper-Script are recognized by the QEMU plugin:

- **sh+ctys**  
This is the standard case, where a wrapper-script with the suffix 'ctys' and a configuration file with the suffix 'conf' exists. The configuration file is the 'almost' only place to configure the settings specific for the current VM. The contained configuration variables comprise the shell variable assignments and additional **ctys-keys** to be handled by the **ENUMERATE** scan. The configuration keys are "sourced" into from an external conf-file, which is required to be coallocated within the same directory.
- **ctys-conf**  
The same as ctys, but the configuration keys are **sourced** from an external conf-file, which is required to be coallocated within the same directory. These are created by the **ctys-cresteConfVM** utility.

The optional <PATH> parameter provides the temporary alteration of preconfigured boot source media.

**BOOTMODE:CD**

This boots an ISO image of a CDROM by setting the boot flag '-boot d'.

**BOOTMODE:DVD**

This boots an ISO image of a DVD by setting the boot flag '-boot d'.

**BOOTMODE:FDD**

This boots an file image. The file could be either a virtual HDD or a the tull path to a physical device.

**BOOTMODE:HDD**

This boots an image of a HDD by setting the boot flag '-boot c'.

**BOOTMODE:PXE**

This boots an image into PXE, currently only for x86 platforms supported.

**BOOTMODE:USB**

This boots an image of a USB-Stick which is actually the same as an HDD image by setting the boot flag '-boot c'.

**BOOTMODE:VHDD**

This boots an image of a HDD by setting the boot flag '-boot c'.

**CONSOLE**

The **CONSOLE** suboption defines the console to be used for the VM.

**CONSOLE:CLI**

The CLI mode is the backed by the plugin CLI, which is utilized in the same manner as the other X11-based plugins, thus attached by usage of the UNIX-Domain socket **QEMUMONSOCK** for a serial port. The CLI console is detachable and could be re-attached later. The VNC console access port is implicitly prepared additionally for later attachment.

**CONSOLE:CLI0**

This mode deviates from the common CLI mode, and is tightly coupled to the VM, thus could not be detached. When the console is detached, the VM will be terminated. When the GuestOS is

shutdown in CLI0 mode the console stays still occupied by the QEMU VM after the guest system is halted. In order to release the CONSOLE/Terminal, the monitor has to be used. Call **Ctrl-A-c-<RETURN>**, and - when the (**qemu**) monitor prompt occurs - execute **quit** within the monitor.

This mode sets particularly the following options and operational modifications.

- **trap: INT,TSTP, QUIT:** These signals are deactivated in the first instance of the called client, and in the entry instance of the remote client(s). Thus the signals, if activated, are transparently passed though to the target peer. The values could be configured by the variable and/or set by the option **-S**.
- **CTYS\_SIGIGNORESPEC.**
- **-b 0,2 -z 2:** This mode sets implicitly **-b 0,2 -z 2**, otherwise the the input stream might be disconnected. The background mode is generally not applicable to CLI0.

#### CONSOLE:EMACS

The same as CONSOLE:CLI, but utilizes for access the type EMACS.

#### CONSOLE:EMACSA

The same as CONSOLE:CLI, but utilizes for access the type EMACSA.

#### CONSOLE:EMACSAM

The same as CONSOLE:CLI, but utilizes for access the type EMACSAM.

#### CONSOLE:GTERM

The same as CONSOLE:CLI, but utilizes for access the type GTERM.

#### CONSOLE:SDL

This is the standard graphical console of QEMU.

#### CONSOLE:XTERM

The same as CONSOLE:CLI, but utilizes for access the type XTERM.

#### CONSOLE:VNC

The VNC console of QEMU. The VNC console access port is implicitly prepared additionally for the following CONSOLE types for later attachment: CLI, XTERM, GTERM, EMACSM, EMACS, EMACSAM, and EMACSA.

#### INSTMODE[:<params>)]

This boots an image as set by **INSTSRC** into a specific installation mode. The installation mode prepares the **INSTTARGET** device by pre-configured actions for usage as installation media for the GuestOS installer.

```
<params>:= <MODE-SRC>%<SRC-PATH>\
           %<MODE-TARGET>%<TARGET-PATH>\
           %(INIT|<custom>)
```

#### <MODE-SRC>

The <MODE> parameter provides the same modes as BOOTMODE of the INSTALLTARGET.

#### <SRC-PATH>

The optional <SRC-PATH> parameter provides the temporary alteration of preconfigured install target media.

#### <MODE-TARGET>

The <MODE-TARGET> parameter provides the same modes as BOOTMODE.

#### <TARGET-PATH>

The optional <TARGET-PATH> parameter provides the temporary alteration of preconfigured install target media.

#### (INIT|<custom>)

This sets the wrapper either to initial install mode for destructive first time actions - like replacement of filesystems, or to post-install mode, where e.g. just some post processing of basic system configuration is performed.

#### KERNEL:[<KERNEL-IMG>[,<INITRD>[,<APPEND>]]]

A specific kernel to boot, with an optional alteration of the initrd - which usually is required - and optional appended kernel arguments.

**-a LIST**

The LIST action displays information about the runtime state of active QEMU-VMs. This comprises QEMU/KVM managed VMs as well as any other, but some specific information like the TCP/IP-Address of the GuestOS are displayed for ctys managed VMs only.

The detection of ctys managed processes requires the bootimage to be the last commandline argument. The files for the bootimage, the wrapperscript, and the conf-file have to be coallocated within the same directory. In addition one of the following naming-conventions has to be fulfilled, the scan-order is as given.

```

        directoryname    ==  conffilename-prefix
    imagefilename-prefix ==  conffilename-prefix
        label            ==  conffilename-prefix

```

**-g <geometry>|<geometryExtended>**

The geometry could be set for the clients only, the resolution parameter **-r** is not applicable:

CLI

SDL

Limited applicable, not yet supported/tested, will follow soon.

XTERM|GTERM

The size Xsiz and Ysiz provide the UNIT of CHARACTERS only.

VNC

As expected.

**-r <resolution>**

Not supported.

## 5 PREREQUISITES

**Supported products:** The following product releases are verified to work.

| Product | Tested Versions              |
|---------|------------------------------|
| QEMU    | 0.9.0, 0.9.1, 0.11.0, 0.12.2 |
| KVM     | 72, 83                       |
| KQEMU   | ffs.                         |
| VDE     | vde2-2.2.0-pre1, vde2-2.2.3  |

### Supported products

**VDE2-Installation**

The VDE tools should be installed into the directory `"/opt/vde"`.

**QEMU-Installation**

"NON-KVM" QEMU should be installed either by standard distribution or into the directory `"/opt/qemu"`.

**KVM-Installation**

KVM should be installed by standard distribution.

## 6 SEE ALSO

*ctys(1)* , *ctys-createConfVM(1)* , *ctys-plugins(1)* , *ctys-QEMU(1)* , *ctys-configuration-QEMU(7)* , *ctys-uc-QEMU(7)* , *ctys-vhost(1)*

## 7 AUTHOR

Maintenance: <[acue\\_sf1@users.sourceforge.net](mailto:acue_sf1@users.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>>



## 8 COPYRIGHT

Copyright (C) 2008, 2009, 2010, 2011 Ingenieurbuero Arno-Can Uestuensoez  
For BASE package following licenses apply,

- for software see GPL3 for license conditions,
- for documents see GFDL-1.3 with invariant sections for license conditions,

This document is part of the **DOC package**,

- for documents and contents from DOC package see  
'**Creative-Common-Licence-3.0 - Attrib: Non-Commercial, Non-Deriv**'  
with optional extensions for license conditions.

For additional information refer to enclosed Releasenotes and License files.

