ctys-**XEN**(1)
XEN Interface

September 29, 2020

Contents

.

1   NAME

cts-XEN - XEN Interface

2   SYNTAX

```
cts -t XEN  -a action[=<suboptions>] ...
cts -T XEN  -a action[=<suboptions>] ...
cts -T ALL  -a action[=<suboptions>] ...
```

3   DESCRIPTION

The **XEN** plugin adds support for sessions to **VirtualMachines** of type **Xen(TM)**. The management of the virtual machines include particularly the support of boot and shutdown for local and remote virtual machines including an inventory with accessible machines. Stored machines on network devices could be accessed by multiple worker-machines, providing a multipath-registration within the inventory. The inventory is populated automatically by scanning the local and remote filesystems and collecting the information of detected VMs into a local ASCII file-database.

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 XEN 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 scope comprises for now the products XEN community edition, which are all TradeMarks of Citrix Inc. The supported product versions are automatically detected, thus no furter parameters for version distinction are required. Current version supports the server versions 3.x. Updates will be available soon.
Additional information for installation is available from **cts-configuration-XEN**, information containing use-cases with application examples is available from **cts-uc-XEN**.

4   OPTIONS

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

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

CANCEL

    The full <vm-address> range is supported. Particularly the wildcard-attribute **ALL** is provided for application on all sessions of selected and permissive user-scope.

CREATE

    All standard parameters not listed here could be applied.

    Dependent on the choosen parameter set some specific CONSOLE types can - whereas some cannot - be applied.

      CONSOLE

        The appropriate settings for the choosen console has to be prepared within the related config file.

        The default CONSOLE could be pre-set by the variable **XEN_CONSOLE_DEFAULT** in the xen.conf file. The original default is **XTERM**. The recommended text console for Gnome is GTERM, but any other could be set as default too.

		XTERM		
		GTERM		
		EMACS		
		EMACSM		
		EMACSA		
	CLI	EMACSAM	VNC	NONE
CONNECTIONFORWARDING	-	-	X	X
DISPLAYFORWARDING	X	X	X	X
SERVERONLY	-	X	-	X
CONNECT	X	X	X	X
RECONNECT	X	X	X	X
REUSE	X	X	X	X
RESUME	X	X	X	X
-b 0 - foreground	B+M	A	A	X
-b 1 - background	-	X	X	X
-z 2 - pseudotty	M	X	X	X

Applicable forwarding modes and call locations for XEN

X) Supported.
A.) Supported, but will block the call-terminal for the whole session,
so might not be used from a single-console environment.
B.) Blocks the console for other calls, thus allows for bulk targets
serial execution only.
M.) Mandatory, but could be suppressed with "-z NOPTY" when a terminal
with some drawbacks is sufficient. One of specific than is that the
password will be echoed for some systems in cleartext, anyhow as
a lonely night-rider it might not hurt you.

Types of CONSOLE to be applied depends on the **-b** parameter for background execution too.
The following behaviour applies:

"-b 0" - synchronous foreground execution
In this mode the current execution thread is performed synchronous in the foreground, this means
particularly a CLI based console cannot be detached, when multiple tasks are in the queue in
order to begin the next. Thus it would result to blocking the remaining sessions until the current
has been finished by the caller.
This parameter is allowed to be applied, but the caller has to be aware of the drawbacks, when
choosing multiple execution targets.
"-b 1" - synchronous background execution

In this mode the DomU will be started by different means for XTERM and VNC only.

CONSOLE:CLI

Will be generally rejected, because multiple execution targets cannot be handled by a single
physical console, and one target could be perfectly handled by *-b 0*.

CONSOLE:GTERM

The **gnome-terminal** which is currently simply mapped to XTERM.

CONSOLE:XTERM

Starts first an xterminal by using the X11 module and initiates the startup of the DomU
within the Xterminal session as a native and synchronous call to **xm -c ....** So it is basically
the asynchronous variant of a CLI call.
CONSOLE:EMACS
The **Emacs** is started in shell-mode, this supports the full scope of edit features on the out-

put buffer. The basic principle is similar to any X11 console with an embedded CLI interface.

CONSOLE:VNC
This case is somewhat different to the previous, in the way that two independent calls for the DomU itself are required.
- The DomU has to be started, which is performed by calling **xm <conf>**.
- The VNCviewer has to be attached to the offered port by the DomU. Therefore a timeout will be applied, which could be controlled by the environment variable **XEN_CONSOLE_DOMU_I** which is used for a sleep-call. Due to buffer handling some console messages could probably be lost.
The client call is an internal call of the VNC plugin, which is independent and could be applied separately.

CONSOLE:NONE
No console is started, any type could be connected later. "-z 2" - force allocation of a pty by ssh   Allocates a pty.

<callopts>

When <callopts> are given, these will be passed through to the call:

```
xm [-c] <conf-path> <callopts>
```

For additional information refer to Xen manual.

-g <geometry>|<geometryExtended>
The geometry has no effect on the server started within the DomU. Just the client will be set:

CLI
Not appliccable.
XTERM|GTERM
The size Xsiz and Ysiz provide the UNIT of CHARACTERS only.
VNC
As expected.

-L <execution-location>

```
<execution-location>=(
   (LOCALONLY|LO)
   | (CONNECTIONFORWARDING|CF)
   | (DISPLAYFORWARDING|DF)
   | (SERVERONLY|SO)
  )
```

-r <resolution>
Not supported.

<xopts>
Refer to common options parts description.

.

5 NOTES

The XEN plugin adds support for XEN sessions where the client types **CLI** , **VNC** , and **X11** are supported. Any non-listed standard option of ctys applies as defined. In cases of divergent behaviour for similar options, and options with specific suboptions, these are listed in this section.

There are some specifics to be recognized and/or applied specific to Xen. This is primarily due to it's nature of the hypervisor interface, where DomU-s are children of the one and only Dom0, which is not **visible** to **ps** as a normal process, but to the specific tools **xm** and **virsh**. Where virsh is part of **libvirt** but prerequired for ctys.

One main challange in combination of access to restricted system resources is the requirement of root permissions for some calls to manage DomU-s. This requires for user-level on demand **CREATE** and **CANCEL** the configuration of **sudo** or **ksu**.

Some drawbacks for the common applied tricks of ctys, using the **CLI** and **ps** as an dynamic storage and exchange interface for runtime information are not working in the altered runtime environment. Even though particularly the **virsh dumpxml** call offers a variety of information. One missing data, that really hurts is the missing information of the used configuration file for the list-ed or dumpxml-ed domain. The **source file** is available - which is the virtual boot-device. But this does not allow an back annotation to related configuration file - this could be just safely defined by additional naming convention, what is done within ctys for simplicity.

Another specifics is a legacy of ctys, which is the definition of ID as a static unique identifier for a VM and PM entity, which does not change, when the entity changes it's state to offline. Resulting of this, the **ID** is for **VMW**, **PM**, and **XEN** the fully qualified pathname of the configuration file, which is not necessarily unique, due to NFS mounted shares on multiple PMs and/or VMs. This is still not unique, when combining the PMs hostname and the pathname of the configuration file, because the contained IDs, e.g. TCP/IP address, MAC address, and UUID are now available within multiple entities, and thus will be listed as though when using administrative management utilities. Anyhow, it should be at least guaranteed by the user, that the entities are unique within the scope a single node. The toolset is prepared to handle various constellations, but it makes the selection by the user easier.

For this the following shortcuts and conventions apply.

- The **Domain-ID** as provided by Xen is for now ignored, the Domain-Name is required to be unique, so the LABEL, which is the Domain-Name, is sufficient as selection criteria. This is anyhow a static constant identifier, which is not true for the Xen-Domain-ID.

  The Domain-ID within ctys - **IDS** for ctys-vhost - is a holomorphic identifier, which is for machines - VMs and PMs - a configuration filepathname, for types of the category HOSTs a dynamic system generated ID such as a PID, DISPLAY, or port.

  Therefore the Domain-ID for Xen within ctys is the filepathname of the configuration file. This is particularly important due to stored information within the configuration file itself, or within the same directory. Due to the only available filepathname for the boot-image of the DomU instance by **virsh**, the fixed - maybe already widely applied - convention is defined, that the configuration file has to be coallocated within the same directory as the virtual boot device for the DomU and to be named the same as the name of the containing directory. This has not necessarily to be the LABEL which is the Domain-Name of the DomU, but could be. SO boot devices, which are physical, not virtual files, are not supported for now.

- **NO SPACES** within ANY entry are supported.

- When multiple LAN interfaces are configured, the MAC-addresses are indexed by their actual order-increment, beginning from 0. These are permuttated with any provided TCP address of the same index. E.g. MAC0 => eth0 => IP0=10.1.1.1, IP0=11.1.1.1.

- Due to the variety of consoles - CLIENTS - which could be attached and are not simply correlated, the LIST action only displays the SERVER components, which are Dom0/DomU, the clients has to be listed by an extra call to CLI, X11, and/or VNC.

- The execution of the creation by **xm** and some **virsh** access permissions has to be activated and required to be with root permissions. Therefore the configuration file **/etc/sudoers** and/or **/root/.k5users** has to be configured. The access privileges by **sudo** and **ksu -e** will be checked and set appropriately. The variable **XENCALL** and **VIRSHCALL** could be preconfigured.

- The execution of XEN requires in any case the VNC module.

- The version supported by XEN is the 3.x version. The tested and verified version is Xen-3.0.3 of the CentOS-5.0 distribution, even though any 3.x version might work. The version evaluation is done by usage of rpm or xm or virsh or xmtrace. The installation paths are evaluated by which call and should be prepared for execution by PATH.

- Due to the warning-output of some tools, this is fetched as ctys WARNING, which could be fully activated by "-d" option.

- Particularly the **D_SYS** debug-level, which traces all system calls, might be helpful for tracing permission settings.

  - The XEN plugin is stack-aware, though prepared to propagate CREATE and CANCEL actions, same for LIST.

- **XEN_CONSOLE_DOMU_INIT_WAIT**
  This variable contains the sleep value after **xm create ...** and before calling a **gnome-terminal** or **xterm**. Therefore in case of a machine which has difficulties due it's performance the value could be adjusted. The current value of 8seconds seems to be safe for initialization of created DomU.

6 SEE ALSO

ctys(1) , *ctys-CLI(1)* , *ctys-configuration-XEN(7)* , *ctys-createConfVM(1)* , *ctys-plugins(1)* , *ctys-uc-XEN(7)* , *ctys-vhost(1)* , *virsch(18)* , *ctys-VNC(1)* , *ctys-X11(1)* , *xm(1)*

7 AUTHOR

Arno-Can Uestuensoez   <https://arnocan.wordpress.com/>
<https://unifiedsessionsmanager.sourceforge.io/>
<https://github.com/unifiedsessionsmanager>

8 COPYRIGHT

Copyright (C) 2008, 2009, 2010, 2011, 2020 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.