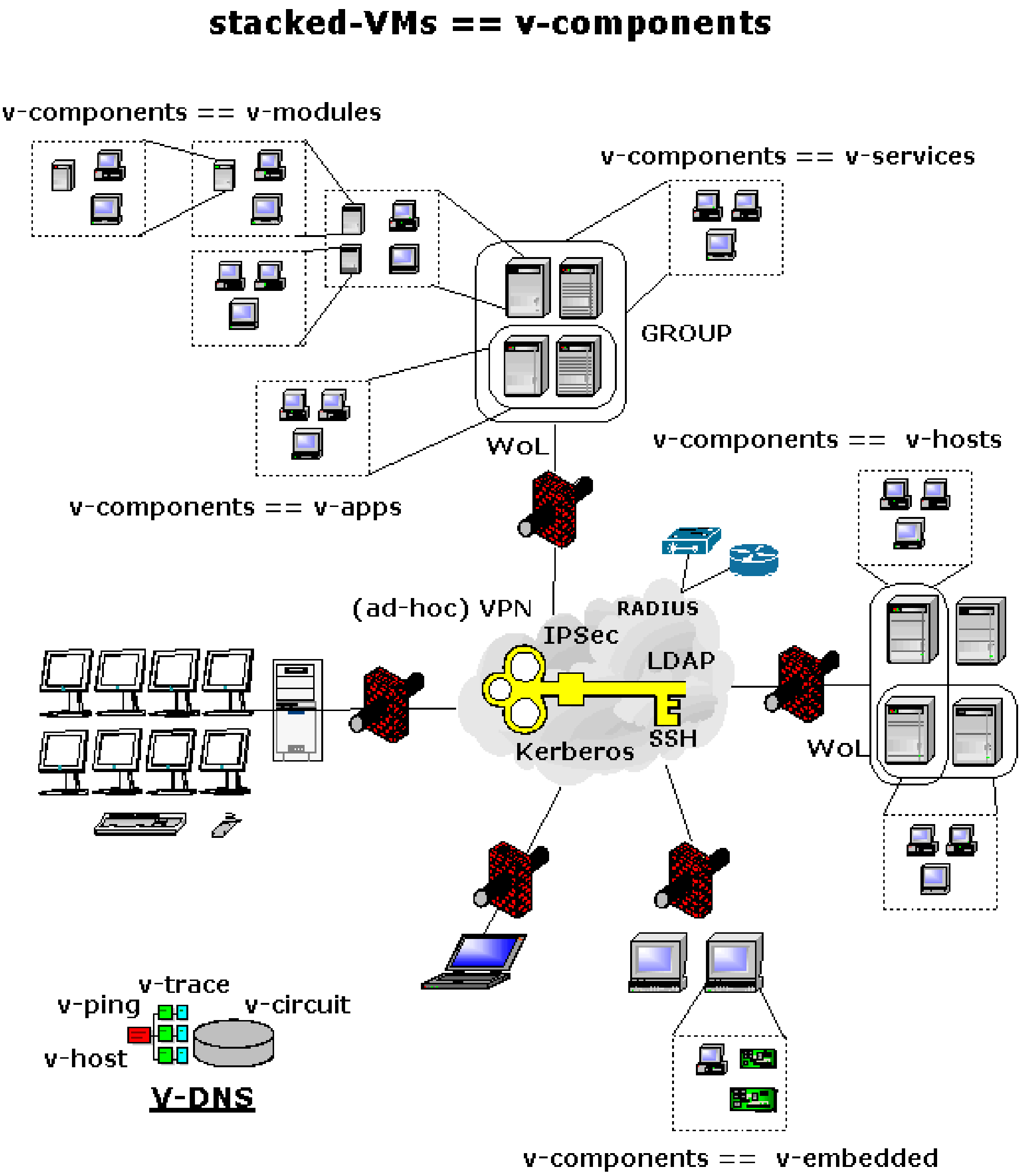

The UnifiedSessionsManager

Install-Guide(draft-pre-release)
Extract from the User-Manual

Version:01.07.001b05a - 2009.01.26

Contents

List of Tables

List of Figures

Licence - GPL3

```
---------------------------------------------------------------------------
PROJECT          = Unified Sessions Manager
---------------------------------------------------------------------------
CALLFULLNAME     = Commutate To Your Session
CALLSHORTCUT     = ctys

AUTHOR           = Arno-Can Uestuensoez - acue@UnifiedSessionsManager.org
MAINTAINER       = Arno-Can Uestuensoez - acue_sf1@sourceforge.net
VERSION          = 01_07_001b02
DATE             = 2008.08.06

COPYRIGHT        = Arno-Can Uestuensoez - acue@UnifiedSessionsManager.org
LICENCE          = GPL3
---------------------------------------------------------------------------
EXECUTING HOST   = ws2.soho
---------------------------------------------------------------------------
```

Chapter 1

Preface

This is just a draft extract from the User-Manual, in order to provide a shortcut for first installation.

Therefore some hyperlinks may be dengling or just reference a page referencing the user manual.

1.1 History

Version	Date	Author	Description
01.03.003.a01[144]	2008.02.11	Arno-Can Uestuensoez	Initial pre-release as embedded printable help
01.07.001.a01[145]	2008.08.03	Arno-Can Uestuensoez	First major update with numerous additions and partial review.
01.07.001.b02[146]	2008.08.11	Arno-Can Uestuensoez	Minor editorial updates. A lot of tests, some fixes.
01.07.001.b03[147]	2008.08.12	Arno-Can Uestuensoez	Minor editorial updates.
01.07.001.b04[148]	2008.08.16	Arno-Can Uestuensoez	Enhancement of documentation and Web-Site.
01.07.001.b05a[149]	2009.01.26	Arno-Can Uestuensoez	Minor editorials.

01.03.003.a01/2008.02.11

The first basic set of features including stack-aware recursive cancel for QEMU, XEN, and VMW. This release contains almost the whole set of first-time implementations(Section **??** '**??**' on page **??**) with personal copyright and offered license under GPL3.

01.07.001.a01/2008.08.03

First major update with numerous additions and partial review. Additional HTML versions and subsets are generated. Still classified as preview quality, but may be stable enough for public production use, as personally applied.

01.07.001.b02/2008.08.11

The first official stable release for flat-VM functions. The stacked VMs are present in first version, where the canonical syntax is implemented. Some more user friendly addresing with enhanced capability-recognition and validation will follow.

1.2 Contact

Public maintenance:	*acue@sf1_sourceforge.net*
Administrative contact:	*acue@UnifiedSessionsManager.org*
Commercial Services:	*i4p.com - Engineering Office Arno-Can Uestuensoez*

1.3 Legal

All mentioned Red Hat products and their registered names are Trademarks of the Company Red Hat, Inc.

All mentioned SuSE products and their registered names are Trademarks of the Company Novell, Inc.

All mentioned VMware products and their registered names are Trademarks of the Company VMware, Inc.

All mentioned Google products and their registered names are Trademarks of the Google, Inc.

All mentioned Sun products and their registered names are Trademarks of the Sun Microsystems, Inc.

All mentioned Microsoft products and their registered names are Trademarks of the Company Microsoft, Inc.

All mentioned Intel products and their registered names are Trademarks of the Company Intel, Inc.

All mentioned AMD products and their registered names are Trademarks of the Company Advanced Micro Devices, Inc.

Xen is a trademark of XenSource Inc.

QEMU is a trademark of Fabrice Bellard.

i4p is a trademark of Ingenieurbuero Arno-Can Uestuensoez.

If some is forgotten, it will be added immediately.

Chapter 2

ctys Setup

2.1 Installation

2.1.1 Basic Install

The current version of the UnifiedSessionsManager supports the install script ctys-install only.

The following steps has to be applied for installation:

1. Download
   The package is provided as a simple gzipped-tar file[150, UnifiedSessionsManager]. It has to be downloaded and unpacked. and unpacked to local filesystem.

2. ctys-install
   The install script in the binary directory of the unpacked package

   "ctys.<version>/bin/ ctys-install "

   has to be called. Several arguments could be applied, a list is displayed by "-h" option.

   An already present set of user specific configuration files is not removed by default, but could be forced to be replaced.

3. Check of required system-call permissions
   A number of required system-calls needs specific root call permissions, which has to be provided by one of the following means:

   - Execution as user "root"
     When using ctys as user root any required permission should be available.
   - ksu
     "ksu" is the sudo like call of kerberos [125, MIT-Kerberos][126, Heimdal] , therefore ".k5users" and/or ".k5login" in the home directory of the permission-offering user(root) has to be stored.

- sudo
  Sudo[127, sudo] could be used aternatively or in combination with "ksu". Sudo provides a more finegrained access administration, but is not(yet?) integrated into kerberos authentication. The file "/etc/sudoers" has to be edited appropriately by visudo.

  The potential pitfall when using SUDO is the requirement of a PTY by default, which could be configured within sudoers, but should be kept due to additional constraints by usage of OpenSSH. The options "-Z" and "-z" handle this issue.

Additionally the required tools such as bridge-utils has to be installed, as checked by the following calls.

The ctys-plugins tool supports a means to check (almost) any internal system call for it's presence and available call permissions. Some limit occurs on "inherent destructive" calls, which have no option to be used for check purposes only. Typicall for this are the shutdown-commands, which on some platforms could not be analysed for access permission without actually shutting down the machine.

The following calls could be used to validate the functionality of current installation:

- Check of Client Functions
  This checks the client set of required function calls.

  ```
  ctys-plugins -d 64,p -T all
  ```

- Check of Server Functions
  This checks the server set of required function calls

  ```
  ctys-plugins -d 64,p -T all -E
  ```

The previous calls utilize the internall common wrapper function

"checkedSetSUaccess"

by setting a specific debugging flag with the value "64=D_SYS". This activates the trace of system calls only. The options are described within the User-Manual, for addtional technical help on this library function call the online help for call interfaces could be used:

"ctys -H funchead=checkedSUaccess"

Any missing but required component as listed by the previous call should be installed and the required access granted by an appropriate entry for sudo/ksu.

4. VMW - Some specifics
   The installation of VMware is quite straight forward for supported Host operations systems by VMware.

   When installed on a non-listed OS, like CentOS, the required build of kernel modules during vmware configuration frequently causes some trouble. Simple warnings could be ignored in almost any case, but when errors occur the "vmware-any-any-update" patch from Petr Vandrovec[108, vmwareAny] should be applied. Some caution is required for usage of the appropriate version for matching the VMware version.

5. QEMU - Setup VDE
   When QEMU is going to be used, the VDE - Virtual Distributed Ethernet from VirtualSquare is required to be installed, therefore should be downloaded and installed by "make install". VDE is used for network setup of TAP devices to be used by QEMU with the utility "ctys-setupVDE" .

   The contained wrappers "vdeq/vdeqemu" are also required. For additional information on ctys wrappers for QEMU refer to Section **??** '**??**' on page **??** .

   A description howto proceed is given in Section **??** '**??**' on page **??** .

6. XEN - Required packages and permissions
   Due to the CREATE and CANCEL actions, which start and stop a DomU, privileged access is required for the caller for permissions on functionality within the Dom0. Particularly to "xm create" and to "virsh". Therefore the virsh of libvirt is required additionally to Xen.

   The current tested versions are Xen-3.0.x and Xen-3.1.x, but newer versions should work too. The main test platform is CentOS/RHEL, for this version only PVM tests are performed, but HVM might work too.

   The access permissions has to be set for sudo and/or ksu.

7. ctys configuration
   The next step should be the adaptation of the provided default configuration to local machine. Therefore the config file in the home directory of the user and/or the installed default files has to be edited. The provided configuration files contain the required description and additional examples.

8. Configure PMs
   The involved physical machines - PMs - should be configured by calling the tool ctys-genmconf in order to generate the PM configuration data. The result is stored by default within the directory "/etc/ctys.d".

   This step requires to be repeated when all hypervisors are installed and marked as operable by the check with ctys-plugins utility. This call updates the stored registration of capabilities available on the local machine, including the later required STACKCAP variable for verification of possible stacking capabilities, when a VM-Stack is going to be started. The update has to be performed before the creation of a cacheDB with the tool ctys-vdbgen .

9. Generate DHCP and/or MAC cache
   The access performance for plugins with stored configuration data - PMs and VMs - will be dramatically enhanced when generating a prefetched cache database.

   The first step required for the creation of the database is the creation of the mapping table generated by ctys-extractMAClst and/or ctys-extractARPlst , either from a valid "dhcpd.conf" of from simply "pinging" a list of hosts.

   The result consists of the data required for mapping of MAC addresses to IP/DNS addresses as provided by ctys-macmap . The format of stored data is documented and could be edited manually. It is an extended "/etc/ethers" format, the "/etc/ethers" database could be generated by the previous tools.

10. Generate cache of configurations for VMs and PMs
    The access to stored static configuration is performed by a two step approach controlled by the option "-c". The first attempt is to read from the generated cache, the second is to scan the filesystem on the target entity in accordance to the provided call options.

    The prefetch of the configuration data in a local file database enhances the call performance dramatically, a factor of at least 10, but almost in any case of 100 and much more is common.

    The build of the cache database is handeled by the tool ctys-vdbgen for collecting the data and by the tool ctys-vhost for preprocessing required correlations and intermediate preprocessings.

11. ctys
    Once this point is reached, ctys should be operational as required. The first tests should be executed by simple dynamic plugins such as CLI, X11, and VNC. Call examples are given in Section **??** ‘**??**’ on page **??** .

    In case of errors the option "-d". provides a scalable degree of debugging information.

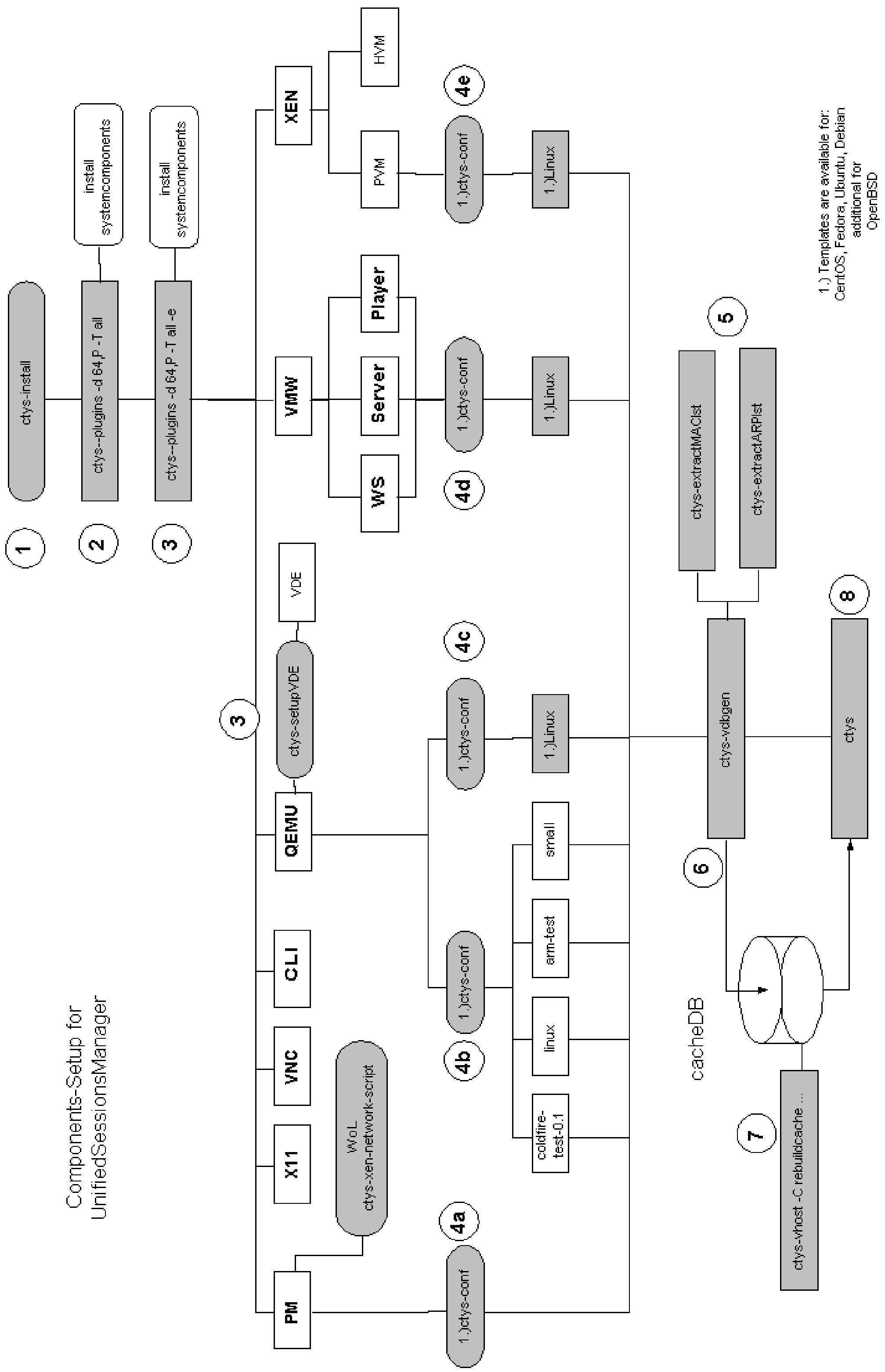

Figure 2.1: Install Steps

2.1.2 Security Environment

Some basic hints are given here only for the wide field of security and general access issues, additional support is available as commercial services only([?], [153], [152]).

General Remarks

Network Accounts

When using UnifiedSessionsManager a network account with SSO is absolutly recommended. There are two exceptions, where the account should be a local account:

- The root account sould be a local-only account, of course.
- In case of required WoL for a machine running a virtual bridge, such as a Xen based machine, a local account is required. A "simple" shutdown may work without any difficulty. This is due to a currently required workaround particularly in case of Xen for setting of WoL on the physical NIC for offline operations of the NIC itself.

NFS

Using NFS has some security risks, even though it is particularly a real benefit for development issues. Thus NFS versions <4 should not be used when crossing insecure network segments without additional provisions.

root Access

Should be configured local only.

Router Configuration

Router have to be configured for WoL, when the target NIC is outside the local segment. This is beyond the scope of this document.

Quick Setup for ssh

The only and one facility supported for communications between any type of entities is the usage of OpenSSH. Particularly the Handling of DISPLAY is deeply embedded into the UnifiedSessionsManager by usage of OpenSSH. Therefore the most benefit results from setting up SSH for SSO. The recommended setup is the usage of SSH in combination with Kerberos.

Quick Setup for ksu

The preferred authentication is the usage of Kerberos, which offers the access configuration facilities by ".k5login" and ".k5users". The usage is quite straightforward, even though for some aspects not as flexible as "sudoers" is. Which is compensated by it's advance for networking purposes.

The only hard-wired restriction for usage of ksu applies, when network users are configured for CANCEL action on bridged NICs. When the WoL feature has to be applied, the NIC needs to be disconnected during the CANCEL procedure with continued access to restricted system resources. Thus this requires "sudoers" and a local user.

Quick Setup for sudo

Using sudo for authentication offers a perfect and straight forward setup for local users. For environments with NIS/NIS+ or any "distribution" utility a networked configuration update is available too. But anyhow, the initial access to a machine, and the "relay-user" is out of the scope of "sudoers".

The network access, particularly the SSO, is crucial for the applicability of ctys in a lager environment, thus sudo has to be used in companion with any network accounting facility only.

Quick Setup for LDAP

LADP is recommended for any distributed directory system. The user information should be handeled by LDAP, which has it's particular advantage when using a SMB based OS.

The setup in a heterogeneous environment is in details somewhat tricky, and requires some more description, though outside the scope of this document.

Quick Setup for autofs

Autofs is particularly the choice for a network login, when data has to be available within a secure segment only this could be simply based on NFS(version<4). The setup could particularly be based on LDAP in combination with Kerberos and SSH, with a lean centralized configuration. The description as provided by the project should suffice for the first steps.

Create a Local User

Even though almost anything could be configured to be accessed by networked users, the CANCEL action presets some additional requirements.

A local user is required for any CANCEL action on machines, where the network has to be disconnected as an intermediary step during the shutdown of the machine. Obviously, the process hangs, when some non-cached-modules has to be loaded via the disconnected connection, e.g. by NFS. This is required for now only for the Xen-3.0.x vesion, when setting WoL on a NIC, which is part of the virtual bridge, and may change for later versions.

2.1.3 Setup Access Permissions

When the decision for the facilities to be used for authentication and authorization is made, and the configuration is finished, the specific required access permisssions for system calls by ctys could be established.

To check the actually present permissions for system calls the embedded debugging interface could be used. There are two basic approaches available:

- ctys-plugins
  This is the specific validation utility, which is a framework including some additional statistical display for the current states of plugins.

```
ctys-plugins -d 64,P -T all -E
```

A second variant validates the local client functionality.

```
ctys-plugins -d 64,P -T all
```

- ctys

  ```
  ctys-plugins -d 64,P -T all
  ```

  This lists all used system-calls with their actual checked execution states.

The check calls are actually the verified system calls encapsulated by a wrapper and performed with a more or less harmless option. The result is verified and weighted dependent of the actual call environment.

2.2 Configuration

2.2.1 Plugins

The following listed configuration file defines the actually used plugins. For example the VMW plugin is active for Linux OSs only, but neither when running on Solaris nor on OpenBSD.

For conceptual information refer to the chapter on design of the User-Manual, particularly to the chapters on modular design and operational states.

```
#When set, the bootstrap-loader ignores the given
# <plugin-type>.
#This should be in case of dependencies such as of XEN from
#VNC utilized carefully. But on machines with OpenBSD
#e.g. the plugins VMW and XEN could be set to ignore safely.
#
#export PM_IGNORE=1
#export CLI_IGNORE=1
#export X11_IGNORE=1
#export VNC_IGNORE=1
#export XEN_IGNORE=1
#export VMW_IGNORE=1
#export QEMU_IGNORE=1
#export OVZ_IGNORE=1
#
#
# This could be configured conditionally for each of the
# following variables, and any other valid shell variable.
# This is particularly helpful in case of NFS mounted home
# directories sharing the identical user-configuration for
# multiple machines within a cluster:
```

```
#
#   MYHOST   : actual host
#   MYOS     : actual OS
#   MYOSREL  : release of actual OS
#   MYDIST   : actual distribution
#   MYREL    : release of actual distribution
#

#
# Some examples for ignoring of XEN-plugin on specific
# sets of nodes:
#
#   ->host01 only
#     [ "$MYHOST" == "host01" ]&&export XEN_IGNORE=1;
#
#   ->Any node NOT on "clust0*"
#     [ "${MYHOST#clust0*}" == "${MYHOST}" ]\
#           &&export XEN_IGNORE=1;
#
#   ->Any node IS on "clust0*"
#     [ "${MYHOST#clust0*}" != "${MYHOST}" ]\
#           &&export XEN_IGNORE=1;
#
#   ->Any node in domain "exe1"
#     [ "${MYHOST##*.exe1}" == "${MYHOST}" ]\
#           &&export XEN_IGNORE=1;
#
#   ->Any node NOT running OpenBSD
#     [ "${MYOS}" != "OpenBSD" ]\
#           &&export XEN_IGNORE=1;
#
#   ->Any node IS running Linux
#     [ "${MYOS}" == "Linux" ]\
#           &&export XEN_IGNORE=1;
#
#   ->Any node IS running CentOS
#     [ "${MYDIST}" == "CentOS" ]\
#           &&export XEN_IGNORE=1;
#
#   ->Any node which IS NOT final execution target
#     REMARK: for now patched: [ -z "$CTRL_EXECLOCAL" ]\
#     [ -z "`echo $*|sed ....`" ]\
#           &&export XEN_IGNORE=1;
#
#   ->....
#
#
# Almost any combination and any additional constraint
# could be added by means of bash.
#
```

```
# When using internal state variables the user is
# responsible for any resulting side effect.
#


#################
#Configuration of supported contexts for standard
#plugins.
#

#PM supports currently Linux and OpenBSD
[ "${MYOS}" != "OpenBSD" -a "${MYOS}" != "Linux" ]\
&&export PM_IGNORE=1;

#XEN is supported on Linux only as server, else as
#client only.
#VNC check will be done by plugin
#Check for "-e", because CTRL_EXECLOCAL is not yet
#initialized.
[ "${MYOS}" != "Linux" -a -n "`echo " $* "|sed -n '/ -e /p'`" ]\
&&export XEN_IGNORE=1;

#VMW is supported on Linux only, else as client only.
#Native local client and VNC access for WS6 will be
#checked by plugin.

[ "${MYOS}" != "Linux" ]\
&&export VMW_IGNORE=1;

##########################################################
#                                                        #
#ATTENTION:                                              #
#  The plugins CLI+X11+VNC could be called MANDATORY for #
#  others, so their "IGNORE-ance" might force unforseen  #
#  side-effects, think twice!!!                          #
#                                                        #
#  Same is true for PM when you require WoL and          #
#  controlled PM shutdown, what might be obvious!        #
#                                                        #
##########################################################
```

Chapter 3

Tool References

3.1 ctys-extractAPRlst

Usage:

```
ctys-extractAPRlst [options] \
   <group/machine-address-list-in-same-segment>
```

Description:

ctys-extractAPRlst generates a sorted list of 3-column table containing:

> <nodename>
> <IP-Address>
> <MAC-Address>

Alternatively output to stdout could be generated for usage in "/etc/ethers".

> <MAC-Address> <nodename>

or

> <MAC-Address> <IP-Address>

Therefore this tool uses a list of targets which has to be in the same segment as the executing machine. The list could contain any valid ctys <machine-address> or any <ctys-group> definition.

The entities will be resolved and accessed as selected by options. The two methods are "ping" which assures the basic TCP/IP-stack access, and "arp" which lists the mapping as required.

So the usage of this tool could be said providing "real-in-time-data", without the limitation to static configured MAC-IP mapping, so recognizing pool-addresses too. But it's application is limited by the area of a single segment.

For additional segments the "ctys-extractMAClst" tool should be used, which relies on configuration data for DHCP servers.

ctys-tools generally expect for input generated files with default column-order, which is defined by "-n".

Options:

-E

Generates output for "/etc/ethers". This is required when using ether-wake for WoL, what is no longer supported by ctys.

The options "-n" and "-i" become in combination with "-E" a slightly different semantic:

-n output is "<MAC-addr> <DNS name>"

-i output is "<MAC-addr> <dotted IP address>"

The option is not supported with "-E".

-n|-i|-m

-n Print sorted records: <name>;<MAC>;<IP>

-i Print sorted records: <IP>;<name>;<MAC>

-m Print sorted records: <MAC>;<name>;<IP>

-h

Print help.

-p <db-dir-path>

Directory for data to be stored. This is - and has to be in this version - the same directory as used by ctys-vdbgen and ctys-vhost.

So each file-based ctys-DB requires it's own mapping file for now. This is going to be extended when the LDAP based addressing is introduced. Anyhow, this feature will remain, because it has some advantages for daily business when setting up small to medium networks or multiple test-environments.

The hard-coded filename is "macmap.cfdb"

-P

Almost same as "-p", but takes default ctys-file-DB, provided by DEFAULT_DBPATHLST.

Default output is stdout.

-V

Version.

-X

Terse output format, effects "-V" when set left-of.

Arguments:

<ping-hostlist-in-same-segment>

Any host within the same segment, will be ping-ed and arp-ed.

Exit Values:

- 0: OK
  Result is valid.
- 1: NOK:
  Erroneous parameters.
- 2: NOK:
  Missing an environment element like files or databases.

3.2 ctys-extractMAClst

Usage:

```
ctys-extractMAClst [options] <dhcp.conf-filename>
```

Description:
ctys-extractMAClst generates a sorted list of 3-column table containing:

<nodename>
<IP-Address>
<MAC-Address>

Alternatively output to stdout could be generated for usage in "/etc/ethers".

<MAC-Address> <nodename>

or

<MAC-Address> <IP-Address>

Therefore this tool requires as source a valid "dhcpd.conf"-Syntax as defined by ISC.

Static configured address mappings are supported only. Dynamic allocated leases of address ranges are not supported.

ctys-tools generally expect for input generated files with default column-order, which is defined by "-n".

Options:

-E

Generates output for "/etc/ethers". This is required when using ether-wake for WoL, what is no longer supported by ctys.

The options "-n" and "-i" become in combination with "-E" a slightly different semantic:

-n output is "<MAC-addr> <DNS name>"

-i output is "<MAC-addr> <dotted IP address>"

The option is not supported with "-E".

-n|-i|-m

-n Print sorted records: <name>;<MAC>;<IP>

-i Print sorted records: <IP>;<name>;<MAC>

-m Print sorted records: <MAC>;<name>;<IP>

-h

Print help.

-p <db-dir-path>

Directory for data to be stored. This is - and has to be in this version - the same directory as used by ctys-vdbgen and ctys-vhost.

So each file-based ctys-DB requires it's own mapping file for now. This is going to be extended when the LDAP based addressing is introduced. Anyhow, this feature will remain, because it has some advantages for daily business when setting up small to medium networks or multiple test-environments.

The hard-coded filename is "macmap.cfdb"

-P

Almost same as "-p", but takes default ctys-file-DB, provided by DEFAULT_DBPATHLST.

Default output is stdout.

-V

Version.

-X

Terse output format, effects "-V" when set left-of.

Arguments:

<dhcp.conf-filename>
This tool requires a valid "dhcpd.conf"-Syntax.

Exit Values:

- 0: OK
  Result is valid.
- 1: NOK:
  Erroneous parameters.
- 2: NOK:
  Missing an environment element like files or databases.

3.3 ctys-genmconf

Usage:

```
ctys-genmconf <options> [<execution-target-list>]
```

Description:
ctys-genmconf generates the initial configuration entry for a PM or VM, which is stored as

> "/etc/ctys.d/[pv]m.conf".

The output is either written to stdout only or duplicated into the previously mentioned file.

The content of this file is required by the VMSTACK feature (refer to Section **??** '**??**' on page **??**) during validation of static consistency and dynamic applicability of the stack members. This is particularly required for hidden files in case of nested VMs to be started, where this data is required to be cached for pre-access within cacheDB. Several additional approaches are available, but this is the most versatile concept, even though it requires some pre-caching efforts.

The current version supports one context only, thus for each change of the booted kernel the differences could require a new generation of the configuration file, including the update of the cacheDB. Future versions are going to support multiple boot-contexts for each PM and VM, which will include the decision for booting the appropriate kernel as decided by the foreseen load-balancing mechanism.

The utility could be performed locally or remotely by full support of remote ctys-addressing.

The generation of data requires root access for some tools. Namely the utility "dmiencode", which is used to evaluate the UUID of the machine requires for execution root-permissions. Anyhow, some older machines may not have a readable UUID at all.

The utility should therefore be executed once on each participating PM and VM by the administrator during installation, and should be completed manually with additional data.

Preferably the rpm package should be installed during initial installation phase, which will generate the appropriate configuration entries.

The data from the pm.conf will be required for the ctys-vhost utility and is therefore fetched by the ctys-vdbgen utility and is evaluated by ctys-dnsutil for dynamic decision of node type.

Options:

-h

Show help.

-I <TCP-address>
Preset value.

-k <ctys-vhost-search-key> The search key to be used as filter for match-only results from the "macmap.fdb" avoiding the usage of local parameters for "host, TCP/IP, and OS". Any valid argument for "ctys-vhost" is applicable.

-M <MAC-address>
Preset value.

-P

Generates standard file path '/etc/ctys.d/[pv]m.conf'.

Default output is stdout.

-r

The range of interfaces to be included, current version supports ethernet interfaces only with the following ranges:

WITHIP
Requires an IP address, which is the "inet" line, currently for IPv4 only.

This is current default.

WITHMAC
Requires a MAC address. An IP address is not neccessary, but could be present.

ALL
Enumerates all ethernet interfaces which is "WITHIP||WITHMAC", localhost is excluded.

-u
Generate a UUID.

-U <UUID>
Preset value.

-V
Show version.

-x <category>
Configuration files for the various machine categories.

PM
Generates the file "/etc/ctys.d/pm.conf".

VM
Generates the file "/etc/ctys.d/vm.conf".

The output is slightly different.

Arguments:

<execution-target-list>
An optional list of <execution-target>. When the "-P" option is choosen, the remote files will be updated, when suffitient permissions are available, else the output is collected locally. The call is simply mapped to a call of the CLI plugin with the option CMD , thus works syncronuous and sequential.

Exit Values:

- 0: OK
  Result is valid.
- 1: NOK:
  Erroneous parameters.
- 2: NOK:
  Missing an environment element like files or databases.

3.4 ctys-macmap

Usage:

```
ctys-macmap [options] <seach-string>
```

Description:
ctys-macmap searches within the given list of "macmap.fdb" for matching lines, and shows the requested fields of result. Simple awk-regexpr as for "ctys-vhost" are supported.

The result is pre-sorted, though in case of multiple results these will be presented sorted.

The contents of "macmap.fdb" are given as one record on each line with the following fields.
<nodename>;<IP-Address>;<MAC-Address>
For the generation of macmap.fdb from standard dhcp.conf and the limitation to static assignment data refer to "ctys-extractMAClst".

Options:

-n|-i|-m
    -n Print: <name>
    -i Print: <IP>
    -m Print: <MAC>

-h
    Print help.

-p <db-dir-path-list>
    Two syntaxes are supported for <db-dir-path-list>:

PATH-like with FS=':':
    <db-dir-path-list>=<db-dir-path>[:<db-dir-path>[:...]]
ctys-conformant with FS='%', due to leveled scanner-hierarchies:
    <db-dir-path-list>=<db-dir-path>[%<db-dir-path>[%...]]

Both supported field separators are reserved values.

Directory paths for databases, containing a file with hard-coded name "macmap.fdb".

When missing, the default DEFAULT_DBPATHLST will be used.

ATTENTION: Currently no spaces are supported within pathnames!

Arguments:

<simple awk-regexpr>
A "simple" regexpr for awk, used as in given extract:

```
awk -v s="\${argLst}"
  '\$0~s\&\&n==1  '{cache=cache \$1;  mx=1;} ...'
```

Therefore e.g.

```
ctys-macmap  -n -m '00:50:56:.3:..:.9'
```

generates the following matches:

```
tst009;00:50:56:13:11:39
tst108;00:50:56:13:11:49
```

EXAMPLES:
The following examples describe some variants of starting a remote

machine by Wake-On-LAN. This is utilized within ctys when using the PM plugin "LINUX".

The evaluation of the MAC-address could be done by several keys from the "macmap.fdb" database. This is the common source when using

```
"ctys-macmap -m <execution-instance>"
"ether-wake <mac-address-from-ctys-macmap>"
```

or

```
"ctys -t LINUX -a create=<-WoL-target> <execution-instance>"
```

ctys-macmap is particularly useful, when the map-database is not yet generated and e.g. local MAC-address for Wake-On-LAN is required for a known name of a PM. Than just call:

```
"ctys-macmap -m <pm-name>"
```

getting the MAC-address

> "00:50:56:13:11:39"

For Wake-On-LAN a.k.a. WoL use the following e.g. in combination with ksu of Kerberos:

```
"ksu -e ether-wake ‘ctys-macmap -m ws1‘"
```

For using "ksu" as a sudo replacement edit "/root/.k5users" only, "/root/.k5users" is not required.

For further information on "ksu" as a replacement for su/sudo refer to Kerberos manuals.

For further information on activating "wol" refer to "http://ahh.sourceforge.net".

Todo:

Introduce "vendor.fdb" for MAC-Vendor-replacement.

Exit Values:

- 0: OK
  Result is valid.
- 1: NOK:
  Erroneous parameters.
- 2: NOK:
  Missing an environment element like files or databases.

3.5 ctys-plugins

Usage:

```
ctys-plugins [<options>] [<execution-target-list>]
```

Description:
"ctys-plugins" is a versatile test-tool for verification of current status of installation. Due to some priority based restrictions it operates partly on a slightly basic level, involving some specific pre-defined trace levels for output control. Anyhow, the required options for quick-usage are documented in the following sections. Extended trace-levels neccessarily require some deeper familiarity with internal design quickly, which is normally nit required for top-level problem handling.

The utility could be performed locally or remotely by full support of remote ctys-addressing without context options on the command line. If for whatever reasons remote context options are reuqired, than could be applied to macro definitions only.

For now the following main tasks are performed by ctys-plugin:

- display of operational states and capabilities of plugins
- distinguish the check-scope of verified features by "-E" for client-site call-RELAY and server-site final-EXECUTION
- display of called system tools with the actual resolved path-prefix
- display of the actual access permissions as configured by "/etc/sudoers" and/or "$HOME/.k5users"/"$HOME/.k5login".
- display of any level of system initialization traces during call bootstrap, refer to common ctys-standard "-d" option

The output is splitted into 3 sections:

1. The first section is the bootstarp of the tool itself, this includes the basic intialization of the framework, where only framework specific options are evaluated. Typical for this is the "-d" options, which is prefetched by the library itself.
2. The second section contains the trace output during initialisation of the plugins. The content dependes on the choosen debug level. Error messages are displayed in any case.

```
Checking PLUGINS-STATEs now...
------------------------------
<trace-output>
------------------------------
...results to:
```

3. The third section contains the processed results of the evaluated raw component states. Therefore an basic overview of the hosting system is given, followed by a sum-up of the variuos sets of plugings related to the different operational states. The final list contains the operational details for each inidividual plugin.

Options:

-d <common-debug-option>

This option is the common analysis and debugging facility of the UnifiedSessionsManager. Due to the wide scope it is maybe somewhat "like a developer interface".

Anyhow, the most important application for a user during installation and first time systems configuration is the analysis of the called system utilities. This frequently leads to some trouble, which might be obvious, once the following call is executed:

```
"ctys-plugins -d 64,P -T all -E"
```

This call checks all system calls for the current node as a final execution location. The availability as well as (some) access permissions are evaluated. When things don't work, this call is the first instance to "ask".

The "-d" option activates a bit-pattern style debug-level by "P" suboption. Than the bit "64" is set, which is the predefined variable "D_SYS", tracing the internal call-wrapper, almost exclusively used for system calls. When not appropriate, the workaround is implemented in a neatless style.

The "-T" option sets simply "all" plugings to be loaded and initialized.

The "-E" option executes a final destination call, instead of an initial or intermediary RELAY-call.

The following call performs almost the same, but as a client or RELAY.

```
"ctys-plugins -d 64,P -T all"
```

-E

Check for local host as final execution target, this forces full verification. If "-E" is not set, only required functionality for a client role is validated, which could be for some packages almost the same and though treated as.

- This is e.g. the case for PM.
- This is e.g. not the case for XEN, which obviously requires a completly different runtime environment for it's clients than within the server as Dom0/DomU
- This e.g. could be or not the case for VMW, depends on type of product + version + requested type of client.

-t <plugin-type>

The type of plugin to be set to BUSY(4), this is any SINGLE plugin as applicable by "ctys -t ..." call.

-T <plugin-preload-typelist>

The prefetch list of plugins to be set to IDLE(2), before performing, this is any comma seperated LIST of plugins as applicable by "ctys -T ..." call.

Arguments:

<execution-target-list>
An optional list of <execution-target>. When the "-P" option is choosen, the remote files will be updated, when suffitient permissions are available, else the output is collected locally. The call is simply mapped to a call of the CLI plugin with the option CMD , thus works syncronuous and sequential.

Exit Values:

- 0: OK
  Result is valid.
- 1: NOK:
  Erroneous parameters.
- 2: NOK:
  Missing an environment element like files or databases.

3.6 ctys-setupVDE

Usage:

```
ctys-setupVDE [options] [<action-arguments>]  \
       [<execution-target-list-arguments>]
```

Description:
ctys-setupVDE encapsulates and combines a subset of functionality for required tools supporting the creation of devices for network access by QEMU.

The utility could be performed locally or remotely by full support of remote ctys-addressing, including context specific target-options, MACROS and GROUPS. E.g. the required system permissions could be preconfigured for specific users by "ksu" and/or "sudo", for additional information refer to Section **??** '**??**' on page **??**

ATTENTION:

> The remote-execution includes some inherent pit-falls to be considered thoroughly!
>
> This is the case, when this utility has to be executed on a remote site, where not yet a bridge (the only supported networking device for now) exists. During the creation of the required bridge - the so called "main virtual-bridge" (see figure: **??**), the network will be disconnected for a short time, so any access to NFS or any other networked file system will be interrupted temporarily, which leads to eventual missing of additional tools to be called, e.g. for reconnecting. The same is true for authentication, when kerberos based "ksu" or "sudo", or any other network centric authentication is used in a non-cached environment, so for non-root users the access to system resources is frequently rejected. Particularly the reconnection of the network device.
>
> Thus remote execution is not approved for users with a mounted remote-home, even though it might work under specific condi-

> tions. Local-only users with "sudo" control by complete locally configured environments are verified to work stable.
>
> A specific behaviour of the current version is applied to the created main-bridges. These will get the same IP and MAC addresses as the logical interface, anyhow it works perfectly, as long as you can cope with multiple interfaces with same address information within applied tools. For the functionality of the UnifiedSessionsManager this is handled by a "sort -u" on resulting enumeration IF-lists.
>
> One reason for "doing" the bridge allocation this way is the minimized risk of detaching the remotely handeled VMs for too long from the network services, which might make them unusable from than on. This concept will be probably modified in future versions.

Anyhow, the remote usage of "ctys-setupVDE", once the authentication is configured properly and security facilities are setup thoroughly, offers a tremendous functionality to centralized setup of VM stacks. This is particuarly true in combination of remote usage of ctys-genmconf ans ctys-plugins .

The usage of ctys-setupVDE assures the appropriate environment for the used of the wrappers "vdeq" and "vdeqemu" of the package VDE[130, sourceforgeVde], which is the recommended tool when TAP devices for Linux has to be created. Anyhow, this utility could be used in any comparable case too, but fit particularly for QEMU setup.

The same configuration as for starting QEMU is used. Therefore neatless communications by usage of QEMUSOCK is guaranteed. The variable QEMUSOCK is based on the variable CTYS_SOCKBASE, which is the default base directory, where UNIX domain sockets are created. This should be used for eventual additional UNIX domain sockets, such as tcp based serial ports or monitoring devices, too. For additional applicability refer to the user manual of QEMU and to the templates provided by UnifiedSessionsManager.

The following tools are bound together within a script:

- vde_tunctl
- vde_switch
- unixterm
- nc
- brctl
- ifconfig
- /etc/init.d/network

Two types of virtual bridges/switches(see figure: **??**) are managed by ctys-setupVDE

- "main virtual-bridge"
  The switch to be used for interconnecting the "external" interface, which is in case of the hosting machine itself a physical NIC.
  This switch is created if not yet present, but has to be deleted manually by the user.

- "vde_switch"
  The switch to be used for attachment of VMs. This switch is completely managed by ctys-setupVDE.

ctys-setupVDE prepares a TAP device and with an attached new bridge, therefore it requires the *"Virtual Distributed Ethernet" - VDE(vde2)*[130, sourceforgeVde] package. Additional information with a WiKi containing some very helpful tutorials [132, Basic Networking] could be found at *"VirtualSquare"*[131, VirtualSquare].

In current implementation some assumptions are made in order to ease design and implementation. Anyhow, for practical application these constraints might not be an important matter.

- one TAP for each vde_switch
- each user has one switch which communicates by default via "/var/tmp/vde_switch0.$USER".

- the management interface for each switch is by default "/var/tmp/vde_mgmt0.$USER".
- appropriate access permissions are provided by sudo or ksu, for automatic detection the ctys framework is used

The following steps are performed by ctys-setupVDE:

1. Creation of a TAP device.

   > "vde_tunctl -u <user-without-root-permission>"

   e.g.

   > vde_tunctl -u acue

   Returns a line like:

   "Set 'tap3' persistent and owned by uid 4711"

2. Use the returned 'tapX' for networking.

   ```
   ifconfig $1 0.0.0.0 up
   brctl addif $2 $1
   ```

   Does the same as:

   > /etc/xen/qemu-ifup tap3 xenbr0

   Which brings up the newly created interface 'tap3' and adds an interface to the virtual Xen bridge connecting it to the world outside.

   The results could be verified with:

   - ifconfig tap3
     should list an interface 'tap3'

- brctl show
  should contain an interface 'tap3'

3. Connect the device.
   Now this interface will be connected to another virtual switch, the vde_switch in order to provide an internal multiplexer for multiple QEMU instances to be connected to the external interfaces e.g. via a present Xen-bridge.

   ```
   QEMUSOCK=/var/tmp/vde_switch0.$USER
   QEMUMGMT=/var/tmp/vde_mgmt0.$USER

   vde_switch -d \
              -tap tap3 \
              -s ${QEMUSOCK} \
              -M ${QEMUMGMT}


   chown -R <userX.groupX> ${QEMUSOCK}
   chown -R <userX.groupX> ${QEMUMGMT}
   ```

   The state could be veriefied with:

   ```
   QEMUMGMT=/var/tmp/vde_mgmt0.$USER

   unixterm ${QEMUMGMT}
   ```

For additional information refer to "??" for the Section ?? '??' on page ?? .

Options:
All options are "optional".

-b <virtual-bridge>
: The virtual bridge connected to the external network to be attached by TAP device. Default is to use the first bridge detected by brctl. If none is present, tha by default a new one is created with the name "ctysbr0", and the first found interface is added to the bridge.

When an interface is provided by "-i" option and a new bridge has to be created, this will be used instead of the first valid.

-d <level>
Sets debug.

-f

Forces execution even when processing seems to be critical.

1. Forces call of "kill <PID>", when here-script with "unixterm ... shutdown" fails.

   For current version this seems to be frequently the case on i386 architecture, whereas x86_64 works.
2. Creates a new bridge, even when connected via a network session. This could interrupt the current calling session permanently, even lead to it's hang-up due to a required short-time disconnect. So this should preferrably proceeded from within a local session.

-g <sbit-group>
Sets the s-bit for the group, this has to be the same as the resulting owner's group.
If not set, the resulting permissions for QEMUSOCK are "rwx——",
else "rwx–S—".

-i <interface>
The interface to be added to a newly created bridge, see "-b" option.

-s <ALTERNATE-QEMUSOCK>
A file-socket to be used for communications peer via virtual switch. Default is set by common QEMUSOCK configuration.

-S <ALTERNATE-QEMUMGMT>
A file-socket to be used for management console of virtual switch. Default is set by common QEMUMGMT configuration.

Could be used with "unixterm $QEMUMGMT" of *VDE(vde2)*[130, sourceforgeVde].

-u <non-privileged-user>[.<group>]
Owner of the created TAP device. Default is current user.

-h
Print help.

-V
Version.

-X
See ctys, terse for machine output.

Arguments:

cancel
Removes the switch and it's attached TAP device. In case of partial present resources these will be cleared as present, thus remaining parts of partly execution could be reset.

check
Performs basic check for the accesibility of the virtual switch setup for selected USER. Therefore a simple "ctys-setupVDE PORTS" call is analysed for the occurance of at least one "tap" device and one UNIX-Domain socket, which are verified by their existence. In case of erroneous state basic information for further analysis is displayed. Anyhow, still malfunction could occur, but if check fails, it will definetly.

create
Creates a new virtual switch, this comprises a new TAP device and an attached virtual switch. When no bridge is present a virtual bridge is created too, and the tap-device is attached.

The CREATE call just checks whether a functional switch is already present, if not it just creates a new one. Therefore the current defined socket for the management interface is utilized.

Thus a new call on a present, but erroneous switch leads to reuse of the sockets, but creates a new tap-device and starts a new instance of a vde-switch-process. Present tap-devices are not reused, and just kept untouched.

info
Shows vde_switch information.

This is the default behaviour.

ports
Lists ports of vde_switch.

list
Lists present vde_switch-es. The base-switch entries are displayed only.

listall
Lists present vde_switch-es. Any entry is displayed, this includes the dynamic created port specific sockets.

Due to some minor difficulties for now these are not removed, when the client disappears, thus "listall" could be used to check the dengling entries from time to time.

<execution-target-list>
Execution targets to be listed.

Exit Values:

- 0: OK
  Result is valid.
- 1: NOK:
  Erroneous parameters.
- 2: NOK:
  Missing an environment element like files or databases.

3.7 ctys-vdbgen

Usage:

```
ctys-vdbgen [options] <target-list>
```

Description:

ctys-vdbgen generates databases of static mapping from configuration files for usage by ctys-gettarget. This tool itself is a wrapper for

```
"ctys -a ENUMERATE=machine...."
```

The output of the ENUMERATE action for a given list is stored into one file contained in the given DB-PATH, or default respectively.

The mapping contains available VMs for a given list of PMs.

Several files could be generated as specific access-groups which could be selected by parameter for ctys call.

The content will be used as runtime-decision base for additional selective evaluation by "ctys -a LIST=MACHINE" call.

REMARKS:

1. Multiple-Entries of IP, and/or MAC addresses
   All entries are evaluated and checked for matching indexes. This is for an "IP0" an "eth0" is searched, if missing a warning is generated. Whereas "eth0" without an IP address will be accepted.

   The second specific is the generation of one seperate entry for each of the resolved IP-MAC pair.

2. Other Multiple-Entries
   The common behaviour in case of multiple entries is to use the FIRST-ONLY. Even though some tools might present more than one, it must not be relied on!

3. Whitespaces
   Whitespaces are generally no supported. When required within suboptions the '%' sign has to be used for padding field-separators.

   Anyhow, please avoid them, at least for now!

Due to the defaults described in the following subsection for options, the call

```
ctys-vdbgen <host1> <host2> ...
```

leads to the default call

```
${HOME}/bin/ctys -a enumerate=machine,b:\$HOME >\
  ${HOME}/.ctys/db/default/enum.fdb <host1> <host2> ...
```

Where the PM plugin by default additionaly checks for the PM configuration which is located in

```
/etc/ctys.d/pm.conf
```

For additional information refer to "ctys-genmconf" .

A special append-mode is supported for addition of data. This mode does a pure concatenation only, not redundancy of added data is tested. Therefore the user should be aware, when calling append mode, whether he already updated the data of that node.

The deletion of data has to be performed manually for now, a simple ASC-II editor, MS-Excel, or the spreadsheet application of OpenOffice could be used for this task.

The data format is described in the appendix of the ctys manual.

Options:

The following options are case sensitive, thus are required literally. If not printed properly, word-options have 2 leading hyphens ’-’+’-’="–".

<same as "ctys -a enumerate=machine,b:<srcpathlst>"

With specific environment variables for modification of the default search path DEFAULT_ENUM_BASE.

```
<srcpathlst>=
   "\${BASEPATHLST:-\$DEFAULT\_ENUM\_BASE}\
\${BASEAPPENDLST:+\%\$BASEAPPENDLST}"
```

which expands by default to

```
"ctys -a enumerate=machine,b:\$DEFAULT\_ENUM\_BASE ..."
```

All context options and suboptions will be passed through, so apply same as for ENUMERATE action.

–append

This activates the append mode, just appending data to present fdb or creating a new if missing. The modes append, replace, and stdio are exclusive and could not be combined.

This is foreseen for addition of minor changes and avoiding a long running recollection of each data set from each target.

-b

Background mode, refer to common options "-b" .

–base=<base-path-list>

A <base-path-list> to be scanned exclusively. This can be combined with multiple targets and the "–append" mode for quick addition of specific new VMs. The path could be the contianing directory for a single VM only.

-c

Nameservice caching mode, refer to common options "-c" .

-C

Result data caching mode, refer to common options "-C" .

–cacheDB=<output-db-directory-path>

Pathname for directory containing DB file to be created. This file stores the mapping records generated from ctys-ENUMERATE literally.

The evaluation order priority and predefined default values for the directory is defined as follows:

1. "–cacheDB=<output-db-directory-path>"
2. DEFAULT_VDBGEN_DB=$HOME/.ctys/db/default
3. DEFAULT_DBPATHLST=$HOME/.ctys/db/default

or

1. "–cacheDB=<output-db-directory-path>"
2. DEFAULT_VDBGEN_DB=<install-path>/conf/db/default
3. DEFAULT_DBPATHLST=<install-path>/conf/db/default

The name for the filedb itself is hard-coded as "enum.fdb". Additional files are be stored within the cacheDB directory.

REMARK:

Only ONE path could be provided here for the PATHLST.

–filecontext

This switches off several defaults and assumes that a configuration file with a complete CONTEXT is provided.

The explicit suppression of defaults is required, due to the addition of chained context options for dialogue based entries from right to left. The evaluation of superposing options is proceeded from-left-to-right. Thus the last wins, so the last dialogue entries have the highest priority.

–progress

This activates the formatted table output, therefore some traces are filtered and displaye in a compact overview, indicating the progress and required processing time.

For the application of the "–progress" option additionally the appropriate trace switches for the target has to be set. This is due to the required activation of the appropriate trace output for post-filtering on local machine. In case of usage of MACROS or GROUPS these could be stored permanenetly as required.

E.g. the following debugging("-d") context options for the target of enumeration required to be present.

```
ctys-vdbgen \
  --cacheDB=/home/tstusr/.ctys/db/tmp \
  --append \
  --base=qemu \
  lab00'(-d 2,s:16,w:0,p)'
```

When "progress" choosen the data for each displayed instance is a post-result, calculated as a sumup after finished processing. For continuous display refer to "progressall".

–progressall

This activates the formatted table output with continuous display, where the data is filtered as well, but any relevant entity with an intermidiate result is displayed. The final sum-up data is additionally displayed as soon as the somplete set of results for the current entity is available.

–replace

This activates the explicit replace mode, which is required when a "enum.fdb" alread exists. Choosing this will delete the present "enum.fdb", when the data has to be kept, make a backup before activating this option. The modes append, replace, and stdio are exclusive and could not be combined.

–stdio

Writes it's output to STDOUT only. The modes append, replace, and stdio are exclusive and could not be combined.

-t

Session type, refer to common options "-t" .

-T

Preload of session type plugins, refer to common options "-T" .

Arguments:

<target-list>
Any target to be enumerated.

Exit Values:

- 0: OK
  Result is valid.
- 1: NOK:
  Erroneous parameters.
- 2: NOK:
  Missing an environment element like files or databases.

3.8 ctys-vhost

Usage:

```
ctys-vhost [options] <awk-regexpr>[ <awk-regexpr>[ <...>]]


ATTENTION: Default behaviour is no output, thus
you need to select one, e.g. "-o ...".
```

Description:

ctys-vhost is the basic address resolution interface for runtime execution of commands based on ctys addressing.

The similarity of UNIX "host" function is expanded with several features, which take into account the roaming of VMs and thus changing their actual execution path within a so called "execution stack" assembled by PMs, VMs, and HOSTs.

Due to technical reasons the locator functionality required for building complete fully qualified ctys-addresses of execution stacks, which could be seen as a UNIX "ping" similarity, is included.

Particularly a basic load-balancing is included, which is very basic of course, but could be extended easily. The current version requires at least one of the potential ExecTargets to be active, which could be a PM for execution of a VM to perform a command, or a VM for execution of a command only, no automatic start of deactivated sessions are performed.

The main task of this tool is supporting a scripting-IF as a link with an convenient name-binding scheme between a GuestOS and it's containing VM and PM by an open and GuestOS-Native interface. The whole access and security facility of VMs and PMs including HOSTs(e.g. VNC) is handled by encrypted connections only. It is designed and implemented as a seamless SSH-based authorization and authentication system. The authors environment utilizes Kerberos, LDAP, and SSH with automount for SSO. The ctys-vhost supports mainly the glue for seamless binding of roaming

VMs on a homogeneous UNIX platform.

Simplicity and common extendibility by widely used bash is another present feature.

Due to targeted simplicity and efficiency the address resolution is based on simple unstructured pattern matching by regular expressions onyl whenever possible(which is almost for each call). Even though the line-record is structured by fields as given by "-a LIST=MACHINE" and "-a ENUMERATE=MACHINE", the match itself will be performed as a simple regexpr by "awk-match". Whereas the output is performed on the level of fields. Multiple regexprs are supported and will be iterative applied on the intermediary results. The reduction of the output as requested by the "-o" option is performed on the final set of resulting records only.

This fits perfectly, as long as the given IDs are kept unique, when ambiguity occurs, the match will be taken as defined by "-M" option.

REMARK:

> This flat-and-simple approach works as good or better to say much better than any "Attribute-Value-Assertion" or any other kind of an Object-Tree based access. And the best is, you can implement it within some days, without any framework to be used!!!
>
> But yes, some selection scope could be tricky, anyhow, for almost any practical relevant query it works fine. As could be seen.

Ambiguity occurs frequently when using a VM with NFS(or an other some more secure network file system) from multiple machines and using the nodes as processing-capacity-only. Accessing the same files and enumerating them for selection of the appropriate execution machine could be a mandatory requirement for load-distribution policies. In this case a basic COST option "-C" supports the very basic "-M" option for some quite usable load distribution within execution-groups.

In most other cases uniqueness should be given, e.g. the key UUID is defined to be unique, but could be tampered e.g. by co-allocated backups. For avoiding of backup-access the "-M first" option might be helpful.

Anyhow, the management of up to some hundred VMs might not be a challange by the current file-DB and not really performance-optimized toolset. One of the next versions will additionally support LDAP based nameservices, targeting an enterprise environment with "unlimited" and "volatile" distributed services to be managed.

When a MAC-IPAddress-HostName mapping table a.k.a. "macmap-DB" is present this will be used for open mappings which are not configured within the VM configuration files a.k.a. enum-DB. Particularly any IP or PM/Hostname address for given MAC-Address will be resolved when not present within the enum-DB.

The address resolution will be performed by the following steps:

1. Check the static list of given ExecGroup for possible candidates.

2. List the active sessions on the given ExecGroup.

3. Take the appropriate PM/VM by utilising "-C" and/or "-M" option.

For additional help refer to online help within "ctys" to the section "NAMESERVICE-BASE".

Now some real benefits when using ctys-vhost as interactive tool instead of using id from scripts:

- Given partial strings, e.g. "192.168.1" lists all machines of that subnet. When the "-M active" option is choosen, all currently active sessions within that subnet are listed.

- Any string could be used as partial pattern, e.g. parts of MAC-Ids of fragments of UUIDs. The given string will be matched against complete record, mostly an awk-regexpr, thus any part, even spanning multiple FIELDS could be used. But currently not regexpr, just literal characters are supported.

- The database founding the mapping information of ctys-vhost could be altered by "-p" option for handling of multiple sets, e.g. for test-purposes.

- The databases enum-DB and macmap-DB are populated just with the native information provided by their main sources, dhcpd.conf and the config-files of supported VMs. Therefore not any information might be present in each of them, e.g. the IP-Address of the GuestOS might be present within the macmap-DB, but not within the VM-config. The "-S" option allows the combined usage of multiple sources, e.g. by values "all", "macmap", or "enum".
- ctys specific configurations-extensions as described for the "-a ENUMERATE" option are fully supported. This includes particularly the storage of GuestOS information within the VM-config by specific ctys-Prefixes(#@#) and some helpful keywords.

The format of the generated data records is literally the same as the MACHINE output of the ENUMERATE action.

Additionally to the flat-matching by simple regular expressions some additional keywords are defined. The are AND , OR , NOT , E: ,and F: , which are described within the section related to the arguments.

Options:

-c <spent cost on execution environment>

Cost as for load distribution when selecting a target.
<spent cost on execution environment>=MINCNT|MAXCNT

MINCNT Gives minimum loaded target, number of given types are simply counted.

MAXCNT Gives maximum loaded target, number of given types are simply counted.

CNT Lists each target with it's TYPE-COUNT.

Companion options apply to resulting set of equal cost.

-C <DB sources>

Limits the generation of the cache DB to the for mapping-resolution to the listed sources. Default is to use all. Only available databases will be used, missing are silently ignored.

Due to some performance issues when repetitively accessing same temporary runtime data, some internal caches are defined. These can be controlled, and reused or cleared by usage of some of the following keywords. But additionally some automatic checks apply.

> For data from static information, which has to be preprocessed a local cache-DB is created. This cache-DB will be checked for modification time of it's sources before each access and updated when outdated.
>
> The modification time of the cache files will be checked additionally for their age. When these exceeds the value defined by CACHECLEARPERIOD, which is by default 3600seconds, the caches are forced-cleared and rebuild silently by next call.

The following data sources are utilized:

ENUM
: Enumaration results only, as supplied by cached local"enum.fdb".

MACMAP
: DHCP information for MAC resolution, the macmap-DB should be available, but is otherwise simply ignored.

  This will be utilized in conjunction with an enumeation result, e.g. ENUM.

GROUPS
: Adds caching of GROUPS for all group files from the current CTYS_GROUPS_PATH variable. Therefore each group file will be completely expanded by nested evaluation and replacement of "#include" statements and stored by replacing each resulting entry with it's MACHINE format entry from the staticCacheDB.

  Each group will be cached within an file by it's own, thus the access could be performed by just one file-selection for the complete nested resolution of it's entities.

```
<DB sources>=
OFF|
CLEARTMP|
CLEARALL|
GROUPS|
KEEPALL|
LIST|
LISTARGETS|
LISTGROUPS|
MEMBERSDB|
MACMAPONLY|
MACMAP|
REBUILDCACHE
```

This group of keywords controls the runtime behaviour, which has an impact to the overall performance.

OFF
: Bypasses the usage of caches.

MACMAPONLY
: Uses the macmap.fdb only for mapping, this is just senseful for mappings between DNS, MAC, and TCP. The request will be rejected, when "-o" option contains any other input.

  For matching entities within MACMAP this might be the fastest approach. It is the only applicable approach, when the target is not yet populated in standard DB, for example due to pre-initial conditions.

MACMAP
: Activates the raw usage of macmap.fdb for DNS, MAC, and TCP as preferred source of resolution.

  This has two flavours, depending from selected output attributes:

  1. Only one of, or all: TCP|MAC|DNS
     In this case the MACMAP DB will be utilized within

the "bigger awk", due complete probable containment of information thus first a raw access to MACMAP will be tried. When no result was found, the general script with DNS/Bind access will be performed. In standard manner(due to SW architecture, ignoring previous trial).

2. Additional output requested:
   In this case particularly the field positions of the resulting output can not be handled in a smart manner for an independent pre-filter, though the standard execution path is performed.

When the macmap.fdb is properly maintained and contains the complete scope of mapping information, this enhances the performance, else it could have an negative impact, even though it will not be dramatic, or for small amounts almost not recognizable.

A second aspect to be aware of is, that the two different databases might diverge. Particularly the order of the stored records could not be relied on to be the same. When using the option "-M all" the order might not be relevant, but for "-M first"(default) and for "-M last" the results might frequently be different.

The basic difference of the contents is the fact, that the macmap.fdb (let us say!) contains any networked host, whereas the standard enum.fdb the registered VMs only, so might be a subset of macmap.fdb.

The correlation of both will be performed, when a cache is build and addressing references are resolved for faster access.

GROUPS

Activates the usage of GROUPS and it's related cache data which is due to performance issues deactivated for now by default.

The following additional keywords control and support the management of internal caches.

LISTCACHE

Lists all current caches.
This call terminates immediately after performing, so any remaining options are ignored.

LISTTARGETS

Lists all current cached targets.

This call terminates immediately after performing, so any remaining options are ignored.

LISTGROUPS

Lists all current cached groups.
This call terminates immediately after performing, so any remaining options are ignored.

MEMBERSDB

Displays a list of all current staticCacheDB members in ctys-stacked-address notation.

CLEARTMP

Clear it's internal temporary caches first and rebuild on demand.

CLEARALL

Clear all it's internal caches first and rebuild on demand.

This includes a directory-wildcard-clear, which includes probably the caches of other tasks, so use it considerably.

This call terminates immediately after performing, so any remaining options are ignored.

REBUILDCACHE

The static data to be concatenated from static assembly databases, for now the enum-DB and the macmap-DB is cached within a static database and concatenated with the volatile RT data into the RTCACHE.

The requirement of rebuild for the static data is checked by modification time of it's components, and when required updated silently.

When setting this flag, the data is rebuild in any case.

Additional information is available from description of:

- NAMESERVICES of "ctys -H"
- "ctys -a ENUMERATE...."
- "ctys-extractMAClst"
- "ctys-vdbgen"

-d <debug-level>
Same as ctys.

-h
help

-i <input-list>
Options controlling input content for specific cases.

```
<input-list>=[CTYSADDRESS|CTYS]
```

CTYSADDRESS|CTYS
A fully qualified address is supported for mapping of one of the given output attributes.

-I <0-9>
Interactive, gives summarised display of progress for main values. The degree of display depends on the choosen level:

- 0
  For completeness only, switches the display OFF, same as omitting the option at all.
- 1
  Activates a moderate level with display of basic benchmark data.

- 2

  Activates a more informative level with intermediate QUERY data reduction pattern. This particularly supports the design of multi-key selection queries for perfomance optimization.

  ```
  ctys-vhost <in-out-options> <arg1> <arg2> <agr3>
  ```

For the display of the actual contents of a specific intermediate step in addition to it's draft performance-overview, just drop all following filters/arguments from the call, what will display the requested result as final. This result is identical to the covered intermediate result when using it within a chained set of filters.

-l <USER>
Remote user to be used for SSH-access-checks, when the "-R" option is activated.

```
DEFAULT=CTYS\_NETACCOUNT(DEFAULT->USER)
```

-M <result-set-output-reduction>
Restricts a set of multiple results with additional constrains for output:

```
<result-set-output-reduction>=FIRST|LAST|ALL
```

FIRST First matching entity.

LAST Last matching entity.

ALL All matching entities.

COMPLEMENT
All entities NOT matching.

SORT Final result is sorted by "sort".

USORT Final result is sorted by "sort -u". Only full matches are reduced.

UNIQUE Final result is sorted by "sort -u" but only displayed when actually one record only results. When multiple records are matched, an empty string is returned and the exit value is set to "1".

-o <output-list>

Options controlling output content. Values of all given options are listed as one RECORD per line for each matched entity. The keywords are not case sensitive and could be used as a comma-seperated list. Shortcuts are applicable mostly as one-character alternatives as listed.

The default output when this option is not provided is to display a pre-configured table stored as a MACRO in the default-macros file with the name

TAB_CTYS_VHOST_DEFAULT

This table could be customized as required, but should be handeled carefully.

```
<output-list>=
(
  (
    (
      [ARCH][,]
      [CATEGORY|CAT][,]
      [CONTEXTSTRING|CSTRG][,]
      [CPORT|VNCPORT][,]
      [CTYSADDRESS|CTYS][,]
      [CTYSRELEASE][,]
      [DIST][,]
      [DISTREL][,]
      [EXECLOCATION][,]
      [GATEWAY][,]
      [HWCAP][,]
      [HWREQ][,]
      [HYPERREL|HYREL][,]
      [IDS|ID][,]
      [IFNAME][,]
      [LABEL|L][,]
      [MAC|M][,]
      [NETMASK][,]
      [TYPE|STYPE|ST][,]
      [OS][,]
      [OSREL][,]
      [PLATFORM|PFORM][,]
```

```
          [PM|HOST][,]
          [PNAME|P][,]
          [RELAY][,]
          [RELOCCAP][,]
          [SERIALNUMBER|SERNO][,]
          [SERVERACCESS|SPORT|S][,]
          [SSHPORT][,]
          [STACKCAP|SCAP][,]
          [STACKREQ|SREQ][,]
          [TCP|T][,]
          [USERSTRING|USTRG][,]
          [UUID|U][,]
          [VCPU][,]
          [VERSION|VERNO|VER][,]
          [VMSTATE|VSTAT][,]
          [VNCBASE][,]
          [VNCDISPLAY|DISP][,]
          [VRAM][,]
        )
        [TITLE|TITLEIDX|TITLEIDXASC][,]
        [MACHINE|MAXKEY][,]
      )
      | TAB_GEN[:<tab-args>]
    )
    [IP|DNS][,]
    [,SORT[:<sort-args>]]
```

The previous keywords for specific fields set the related bit for output. These will be OR-ed to the resulting output. Thus the MACHINE keyword includes all fields, whether individually set or not.

The format keys IP and DNS change the representation of the IP field.

ARCH
The architecture presented by the hypervisor to the GuestOS.

CATEGORY|CAT
The category of the plugin, which could be for now one of: HOSTs, PMs VMs.

CONTEXTSTRING|CSTRG
A string stored for the use by responsible the plugin.

CTYSADDRESS|CTYS
A fully qualified address to be used within ctys. This includes the complete address for the whole execution-stack of the destination instance, beginning with hosting PM.

Whereas almost any other output is just a subset of the generated static database, this value is the result of the assembly of multiple items to a complete address for an unambiguous execution path. The namespace could be the private network or even the global network, when globally unique PM addresses as FQDN are used.

CTYSRELEASE
The release of ctys used for creation of the VM.

DIST
Output of distribution installed within VMs guest.

DISTREL
Release of distribution.

DNS
Output of TCP/IP address (any valid for the VM). This option supports the name representation as reported by DNS, for the numerical representation refer to IP.

ATTENTION: Only the first match will be listed when multiple addresses are present for the same entity.

EXECLOCATION
The location of execution for the VM. Either a keyword, or a list of hosts/groups.

GATEWAY
The TCP gateway to be used for the current interface, which is for the standard case the one for the whole multihomed node.

HWCAP
The offered hardware capacity by the VM to the GuestOS.

HWREQ
The required hardware capacity of the VM from the PM, which could be a lower peer VM within a stack.

HYPERREL|HYREL
The release of the hypervisor the current VM is created with. E.g. "Xen-3.0-x86_64".

IDS|ID|I

Output of static ID, which is a pathname for VMs, and a runtime ID for HOSTs. The IDs are (foreseen to be!?) unique within the namespace of their PM or VM. This should be considered when roaming VMs between PMs.

Following current exceptions apply:

XEN

The value is the configuration path statically unique on local host, common to IDs of other VMs.

The domain-ID is handled - due to hypervisor architecture and structural and dynamic means of accessibility - similar to an ordinary "UNIX-pid", but not considered within ctys.

HOST

For plugins of type HOST, which are more or less simple processes offering specific services, the "UNIX-ID" is utilized.

The "UNIX-ID" could consist of several kinds of entries. A common example is VNC, where the entries semantic could be one of:

- DISPLAY = VNC-port-offset
- DISPLAY = VNC-port

- Any of above could be context-specific, and utilized more or less correlated by any other FBP-aware application too. E.g. vncviewer for XEN and WMWare-Workstation 6.

In addition, for a plugin a ctys specific ID might be defined, e.g. based on "UNIX-PID".

So, ... it is just an abstract ID, no generic overall-algorithm applicable.

IP

Output of TCP/IP address. This option supports the numerical representation, for the DNS name representation refer to DNS.

LABEL|L

Output of LABEL.

MAC|M
: Output of MAC address.

  ATTENTION: Only the first match will be listed when multiple addresses are present for the same entity.

MACHINE
: Complete records matching the <regexpr-list> in terse format for postprocessing.

MAXKEY
: The maximum common set of attributes for LIST and ENUMERATE.

NETMASK
: The TCP netmask of current interface.

OS|O
: Output of OS as configured.

OSREL
: Release of OS.

PLATFORM|PFORM
: The HW platform provided for the GuestOS.

PM|HOST|H
: Output of TCP/IP address of the PM-Physical Machine, which is the hosting machine.

PNAME|P
: The same as <ID|I>, this is due to the usage of filepathname of the configuration as an unique ID at least within the namespace of a sigle hosts filesystem.

RELAY
: The relay interface, device, virtual bridge, virtual switch, or virtual hub, the VM is interconnected too witin it's PM/lower-stack-peer.

RELOCCAP
: The available capacity for relocation of the VM, either to another compatible virtual PM as a stack-entity, or an actual physical PM. The destination container has to provide the required HWREQ and STACKREQ of the VM, which has to be compatible with the HWCAP and STACKCAP ot the target.

SERIALNUMBER|SERNO
: An arbitrary serial number for the VM stored in the configuration file. This number should be unambigiuos.

SERVERACCESS|SPORT|S

Server access port for execution of a TCP/IP connect. This is the raw port to be used for server specific admin tools, which is different from user's client access. This port is currently rarely supported, namely not utilized due to security reasons, e.g. in case of XEN.

The main intention of ctys is to avoid propriatery interfaces as much as possible, and support "bare support tools" only. This interface could only be propriatery. So being honest, 'do not really like that!

SSHPORT

A list of provided SSH ports on this interface. Currently supported for OpenSSH only.

SORT

Enables the post-sort filter.

```
<sort-args>=[ALL|EACH][%UNIQUE][%<sort-key>]
```

UNIQUE

Activates a pre-final filter for call of "sort -u".

<sort-key>

Defines a sort key as "-k" option for "sort -k <sort-key>".

STACKCAP|SCAP

The capacity offered by the hypervisor to nested VMs.

STACKREQ|SREQ

The capacity required by the hypervisor as a nested VM itself.

STYPE|ST

Output of the session type, either of category VM, PM, or a HOST by it's plugin name.

TAB_GEN:<tab-args>

Refer to common format for additional information.

TCP|T

The ip address of the VM in stored format.

ATTENTION: In case of multiple interfaces and/or addresses for each address of a so called "multi-homed" machine a sepereate entry is generated, thus it is listed as a seperate host entry.

TITLE
: The title for any selected field within the output.

TITLEIDX
: The title with the related indexes as required and enumerated for input into the generic table.

TITLEIDXASC
: The title with the related indexes as required and enumerated for input into the generic table. In addition the ASC-II values of column indexes for common spreadsheet forms are displayed.

USERSTRING|USTRG
: A free editablecustomizable string from the user.

UUID|U
: Output of UUID.

VCPU
: The number of pre-assigned VCPUs.

VERSION|VERNO|VER
: Version of config.

VMSTATE|VSTAT
: The configured state of the VM. Current aupported values are: ACTIVE, BACKUP.

VNCBASE
: Base port for calculations of ports from display and vice versa. The default is 5900.

VNCDISPLAY|DISP
: DISPLAY to be used by XClients, which in case of VNC is already calculated by usage of context-specific PortOffset.

VNCPORT|CPORT
: Client access port for execution of a TCP/IP connect. This is the raw port to be used for vncviewer or proprietary clients with their own MuxDemux-dispatcher.

All configured VNC access ports for any VM could be listed as:

```
ctys-vhost -o cport,l -M all '59\[0-9\]\[0-9\]'
```

Where a standard baseport of 5900 is assumed.

VRAM
: The amount of pre-assigned VRAM.

-p <db-directory-path-list>

Comma seperated path list to directories containing the name-resolution DBs, same for each <db-directory-path> as for ctys-vdbgen.

ctys-vhost could handle multiple mapping-DBs for virtual concatenation. The advantage of this is the ability of substructuring VMs and PMs into access-groups by ctys-vdbgen and using them in combinations as required during runtime. This offers particularly advantages when performing ctys-vhost for loadbalancing by usage of cost-option "-C".

-r

Activates the common usage of dynamic runtime data. Without this option only some distinct functions like load-distribution utilize selective calls of runtime-data-evaluation for further restricting their intermediate results. This is e.g. obviously the count of actual executed instances on a PM for the case of cost evaluation on a potential distribution target.

When runtime data evaluation is activated in general, the "-R" option alplies to any result as a further constraint.

The usage of runtime data evaluation cost performance of course. This could become dominant, when huge clusters are evaluated, thus should be considered whether really required, and applied to reasonable sets only. But anyhow, when some bigger sets are required by definition, caching of data with different strategies could be applied.

-R <runtime states>

Restricts a set of multiple results with additional constrains for output.

Only the possible targets which are actually operational are listed. This includes the actual running VM with it's hosting PM, and in addition all other operational machines, where the current VM is available too. This case is the most common for NFS based exec-pools, where a single VM could be accessed remotely by a number of PMs. This particularly offers the advantage of copyless-migration of online and offline VMs.

Very handy, and in companion with others probably one of the most important internal top-level-calls for GuestOS-Command-Execution.

```
<runtime states>=
[MARK|(REVERSE|R|-),]PING|SSH[,PM|VM]
```

MARK
: A match for any of the following keywords is simply made with a prefix as running by "R;", instead of just showing the resulting set.

  The remaining will be formatted with "-;" as prefix for alignment.

REVERSE|R|-
: This reverses the resulting set, thus the "not matching" states only will be shown.

PING
: A RUNNING session is simply "ping-ed".

  Resulting statement of operational mode may result on applied security policies, but not exclusively on the state of the running OS's IP-stack.

SSH
: A RUNNING session is additionally checked for SSH-connect by a dummy-interconnect.

  This might be really senseful, once SSO is established.

  "ssh" is the only and one state, which is a viable confirmation for the ability of establishing ctys connections.

PM
: Checks only PM for accesibility, which is the default behaviour.

  PM accessibility is defined as the accessibility of the running OS on PM.

VM
: Checks VM for accesibility, this is particularly related to the SSH key.

VM accessibility is defined as the accessibility of the running OS on VM.

-s

Set when ctys-vhost is used as an internal subcall for another master-tool. In this case some automatic triggered tasks such as the time-driven rebuild of caches are suppressed. Instead a hint for required re-sync is printed as warning.

Urgent tasks will be worked out, even if they might take some minutes. This is the case when no cache is present, of the caches differ in their age.

All tools using this as an internal system call should set this flag.

-S <BasicDataManagementSupport>

The "-W" option represents some basic management interfaces for the additional entity class GROUPS and the entity characteristics CONTAINMENT. Where the containment is applied to the whole set of stored entities.

These interfaces allow some smart listing and display of current supported data, the handling of data as deletion and creation is handeled by the ctys-vhost command as appropriate.

```
<BasicManagementSupport>=
LISTALL|
LIST|
LISTDB|
MEMBERSDB|
LISTGROUP|
MEMBERSGROUP[23]
```

The following keywords may be applied.

LISTALL

Displays a list of all current available data sources.

LIST

Displays a list of all current data sources, the same as

LIST = LISTDB + LISTGROUP

LISTDB
Displays a list of current file-databases.

MEMBERSDB
Displays a list of all current staticCacheDB members in ctys-stacked-address notation.

LISTGROUP
List all current groups from the CTYS_GROUPS_PATH.

The output format is as follows:

" <size> <#lrec>/<#incs> <#srec> <group>"

<size> Size n kBytes.

<#lrecs> The overall number of target entities without resolution of nesting, so just the current file is evaluated.

<#incs> The overall number of include-statements contained within current file.

<#srecs> The overall number of target entities with resolution of all nested includes.

<group> The name of current group, which is the filename too. When "-X" option is set (LEFT of this option), than the basename is shown only, else the full filepathname.

MEMBERSGROUP[:<group-list>]
Lists members of scanned groups. When no <group-list> is provided, the variable CTYS_GROUPS_PATH is decomposed and similar to the PATH variable, any resulting directory is scanned for all existing group files. The members of found groups are displayed.

The nested containment hierarchy by "include" is expanded before output.

In case of provided <group-list> the listed groups are displayed only. The format of <group-list> is:

```
<group-list> =: <group-name>[%<group-list>]
```

Two types of storage are shown:

- Raw group files, which may contain target entities, include-statements and comment lines.
- Cache group files, which contain the whole resolved set of containment tree as flat target entity recorded from the statCacheDB.

(MEMBERSGROUP2|MEMBERSGROUP3)[:<group-list>]

Same as MEMBERSGROUP, but with slightly different output format.

Additional information is available from description of:

- NAMESERVICES of "ctys -H"
- "ctys -a ENUMERATE...."
- "ctys-extractMAClst"
- "ctys-vdbgen"

-T <type-list>
Types to be recognized when calculating target. For additional information refer to "-T" option of ctys.

-V
Shows version.

-X
See ctys, terse for machine output.

REMARK: Due to order dependency of options evaluation, set this as first/leftmost option.

Arguments:
It is recommended to try the "-I 2" option for some performance analysis of order dependency for multiple-selection queries.

<awk-regexpr>[<awk-regexpr>[<...>]]
A list of simple awk regular expression, for matching based on $0. This is called here "flat-matching", though no structural information like in case of attribute-value assertion, is recognized for the pattern match.

The given lists are matched each on the resulting set of complete records from the previous pattern-matching. The last filter applied will be accompanied by reduction of fields of final matching records as selected by "-o" option.

The main advantage of this approach is the simplicity of data structures and the utilization of common tools and data structures. Some performance gain is another advantage.

The drawback is, that in some cases the regexpr has to be choosen thoroughly.

Some Examples:

. (a single dot)
All items within the database.

inst
All items which contain any string "inst"

'înst'
All items, which start with "inst", where the first field in a record is the hostname.

'*inst'
All items, which end with "inst".

'xen|qemu'
All items containing 'xen' or 'qemu'.

AND

The AND operator is the the same as a simple space-operator(" "), which causes the keyword to be applied as selective filter on the previous intermediate result. The result is matched based on the internal MACHINE format, which might lead to different results than the requested final output format only.

E:<#field0>:<#field1>
Compares two fields given by their canonical numbers. The most important application might be the quer for a specific PM record, where the "netname" has to be matched by "PM", which is the "uname -n". Be aware, that only substrings and equal strings match, for local networks using DNS, the "netname" has an additional point "." at the end, thus order of

numbers are significant for a match.

The "$<#field0>" is the canonical number as presented by TITLEIDX .

F:<#field0>:<content-match>
Queries for a specific FIELD with provided number to be compared by awk-function "match($<#field0>,<content-match>)". Be aware, that only substrings and equal strings match.

The "$<#field0>" is the canonical number as presented by TITLEIDX .

NOT
The NOT operator replaces the current composite state for the next argument only, operators are skipped. It should be recognized, that the NOT operator replaces only the current state, thus no chained evaluation of previous operators is applied. Anyhow, different operators, which are independant, such as NOT and AND, are superposed.

OR

The OR operator adds to the previous intermediate result a filtered subset of the last "AND-result". This sounds maybe a little strange, but simply said, a number of grouped OR operators just imply a parentheses/brace around all OR-ed elements. The overall operations is simple from-left-to-right.

The reason for omitting group-operators is just simplicity of implementation and grant of a resonable overall performance. When more operators are required, a full set of syntax might be implemented.

Exit Values:

- 0: OK
  Result is valid.
- 1: NOK:
  Erroneous parameters.
- 2: NOK:
  Missing an environment element like files or databases.

- 7: NOK:
  Missing cacheDB directory.
- 8: NOK:
  Missing stat cache.
- 9: NOK:
  Missing groups cache.
- 10: NOK:
  Missing "macmap.fdb"
- 11: NOK:
  Unambiguity was requested by "-M unique", but query result is ambiguous.

3.9 ctys-utilities

3.9.1 ctys-install

Usage:

```
ctys-install [<key>[=<value>]] ...
```

Description:
"ctys-install" is the base interface for installation and update. Therefore several call options are provided.

- initial install - common paths
  The "initial install" has no specific options, just the generic options for redefinition of install paths are available.
- update
  The options force, foceall, and forceclean are particularly available for protection or deletion of present user specific configuration files.
- common sources - symbolic links
  The "linkonly" option forces the installation of symbolic links pointing to the actual location of the called "ctys-install" directories and executables instead of performing a full scale install by copy.

  The usage of symbolic links for each executable has particularly advantages for "quick-installs" and test purposes when used in combination with a commmon directory structure on all machines by NFS. Thus multiple user accounts on multiple virtual and physical machines could be used for access to modified versions without requiring an update.

  The required configuration and template directories are still copied to the user's home for further modification. Several configuration files are initially present only as defaults within the resulting install directory and has to be copied by the user when required to be altered.

The following combinations of call options for install cover the common install procedures.

1. Individual Install
   Installs by copy of configurations and local lib-subdirectory containing the copy of installed version.

   - "ctys-install"
     The basic and initial call, when a complete local copy is required.
   - "ctys-install force"
     Suppresses the version check, which by default allows for updates with "higher" version numbers only. Thus the current version will be installed independetly of the previous.
   - "ctys-install forceall"
     When previous configuration or templates are present, by default the installation is canceled with a notification. When FORCEALL is set, the present directories are moved to a backup directoy by appending the current date to the directory name.

     The caution in handling this directories is due to the fact, that the configuration directories contain the individually edited configuratiob, GROUP and MACRO files, and the generated cacheDB files.
   - "ctys-install forceclean"
     The option FORCECLEAN forces the previous deletion of installed componnents, no backups are created.

2. Centralized Install by Local Symbolic Links
   Installs a copy of configurations and creates symbolic links to the install sources.

   - "ctys-install linkonly"
     The basic and initial call, where locall symbolic links and a minimal copy of configuration and template files is created. The actual runtime system is just called by symbolic links, where for each executable an individual link is created by default in "$HOME/bin".
   - "ctys-install linkonly force"
     See "ctys-install force"

- "ctys-install linkonly forceall"
  See "ctys-install forceall"
- "ctys-install linkonly forceclean"
  See "ctys-install forceclean"

3. Centralized Install by Search PATH
   Just install to any target, either by copy or by linkonly, and set the PATH as required.

Options

[libdir=<default=$HOME/lib>]

DEFAULT=$HOME/lib
The root path for the physical install target. The actual physical install is handled by a name including the current version in a similar manner as shared libraries naming convention.

[bindir=<default=$HOME/bin>]

The path to the starter directory, which is contained in the PATH variable.
Symbolic link: $bindir/ctys -> $libdir/ctys.<version>/bin/ctys
Bootstrap file: $bindir/bootstrap/bootstrap.<version>

[templatedir=<default=$HOME/ctys>]

Directory to templates and test-data matching current version. When an emtpy string is set, the install is suppressed, and the templates are contained within the libdir only.

[remove<default=unset>]

Removes previous configuration and templates.

[noconf<default=unset>]

Suppresses the install of initial configuration files. Could be somewhat dangerous, because some essential parameters are stored within the configuration and should match the executed version.

[force<default=unset>]

Checks "ctys -V -X" alphabetically/literally, normally only updates are allowed, but force installs in any case.

Current users configuration director $HOME/.ctys will be left unchanged.

[forceall<default=unset>]

Checks "ctys -V -X" alphhebetically/literally, normally only updates are allowed, but forceall installs in any case.

Anything, else than users configuration directory $HOME/.ctys, will be removed and installed again. The users current configuration will be moved to $HOME/.ctys.bak.$DATETIME.

[forceclean<default=unset>]

Checks "ctys -V -X" alphhebetically/literally, normally only updates are allowed, but force installs in any case.

Anything, including current users configuration directory $HOME/.ctys, will be removed and installed again.

[linkonly<default=unset>]

Suppress the local copy to $HOME/lib directory, just appropriate symbolic links are set to the given source directory determined by this call.

Anyhow, the configuration is copied and/or preserved as usual.

[version|–version|-V]

[-X]

[help|–help|-help|-h]

3.9.2 ctys-install1

Internal call for ctys-install.

Bibliography

Books

UNIX

[1] Juergen Gulbins: *UNIX Version 7, bis System V.3,* Springer-Verlag Berlin Heidelberg; 1988; ISBN: 3-540-19248-4

[2] Maurice J. Bach: *The Design Og The UNIX Operating System,* Prentice Hall, Inc.; 1986; ISBN: 0-13-201757-1

[3] Sebastian Hetze, Dirk Hohndel, Martin Mueller, Olaf Kirch: *LINUX Anwender Handbuch,* LunetIX Softair; 1994; ISBN: 3-929764-03-2

[4] Rob Flickenger: *LINUX SERVER HACKS,* O'Reilly&Associates, Inc.; 2003; ISBN: 0-596-00461-3

[5] Olaf Kirch: *LINUX Network Administrator's Guide,* O'Reilly&Associates, Inc.; 1995; ISBN: 0-56592-087-2

[6] Daniel P. Bovet, Marco Cesati: *Understanding the LINUX KERNEL,* O'Reilly&Associates, Inc.; 2003; ISBN: 0-596-00002-2

[7] Daniel P. Bovet, Marco Cesati: *Understanding the LINUX KERNEL(2nd. Ed.),* O'Reilly&Associates, Inc.; 2003; ISBN: 0-596-00213-0

[8] Wolfgang Mauerer: *Linux Kernelarchitektur - Konzepte, Strukturen und Algorithmen von Kernel 2.6,* Carl Hanser Verlag Muenchen-Wien; 2004; ISBN: 3-446-22566-8

[9] Alessandro Rubini: *LINUX Device Drivers,* O'Reilly&Associates, Inc.; 1998; ISBN: 1-565592-292-1

[10] Alessandro Rubini, Jonathan Corbet: *LINUX Device Drivers(2nd. Ed.),* O'Reilly&Associates, Inc.; 2001; ISBN: 0-596-00008-1

[11] Jonathan Corbet, Alessandro Rubini, Greg Kroah-Hartman: *LINUX Device Drivers(3nd. Ed.),* O'Reilly&Associates, Inc.; 2005; ISBN: 0-596-00590-3

[12] Juergen Quade, Eve-Katharina Kunst: *Linux-Treiber entwickeln,* dpunkt.verlag GmbH; 2004; ISBN: 3-89864-238-0

[13] Wehrle, Paehlke, Ritter, Mueller, Bechler: *Linux Netzwerkarchitektur - Design und Implementerung von Netzwerkprotokollen im Linux-Kern,* Addison Wesley, Inc.; 2002; ISBN: 3-8273-1509-3

[14] Brandon Palmer, Jose Nazario: *Secure Architectures with OpenBSD,* Addison Wesley, Inc.; 2004; ISBN: 0-321-19366-0

[15] Michael W. Lucas: *Absolute OpenBSD - UNIX for the Practical Paranoid,* No Starch Press, Inc.; 2003; ISBN: 1-886411-99-9

[16] Chris Tyler: *FEDORA LINUX,* O'Reilly&Associates, Inc.; 2006; ISBN: 1-596-52682-2

[17] Benjamin Mako Hill, Jono Bacon, Corey Burger, Jonathan Jesse, Ivan Krstic: *The Official ubuntu Book,* Pearson Education, Inc.; 2007; ISBN: 0-13-243594-2

[18] Jonathan Oxer, Kyle Rankin, Bill Childers: *UBUNTU Hacks - Tips&Tools for Exploring, Using, and Tuning Linux,* O'Reilly&Associates, Inc.; 2006; ISBN: 1-596-52720-9

[19] Tim SchÃ¼rmann: *(K)Ubuntu - Installieren - Einrichten - Anwenden,* Open Source Press; 2006; ISBN: 3-937514-30-9

[20] Michael Urban, Brian Thiemann: *FreeBSD 6 Unleashed,* Sams Publishing, Inc.; 2006; ISBN: 0-672-32875-5

[21] Marshall Kirk McKusick, George V. Neville-Neil: *The Design and Implementation of the FreeBSD 6 Operating System,* Addison Wesley, Inc.; 2005; ISBN: 0-201-70245-2

Security

[22] Guenter Schaefer: *Netzsicherheit - Algorithmische Grundlagen und Protokolle,* dpunkt.verlag GmbH; 2003; ISBN: 3-89864-212-7

[23] Alexande Geschonneck: *Computer Forensik - Systemeinbrueche erkennen, ermitteln, aufklaeren,* dpunkt.verlag GmbH; 2004; ISBN: 3-89864-253-4

[24] Jonathan Hassel: *RADIUS - Securing Public Access to Privaze Resources,* O'Reilly&Associates, Inc.; 2002; ISBN: 0-596-00322-6

[25] Jason Garman: *Kerberos - The Definitive Guide,* O'Reilly&Associates, Inc.; 2003; ISBN: 1-596-00403-6

[26] Daniel J. Barret, Richard E. Silverman & Robert G.Byrnes: *SSH The Secure Shell - The Definitive Guide,* O'Reilly&Associates, Inc.; 2005; ISBN: 1-596-00895-3

[27] John Viega, Matt Messier & Pravir Chandra: *Network Security with OpenSSL,* O'Reilly&Associates, Inc.; 2002; ISBN: 0-596-00270-X

[28] Jonathan Hassel: *RADIUS,* O'Reilly&Associates, Inc.; 2002; ISBN: 0-596-00322-6

[29] Gerald Carter: *LDAP - System Administration,* O'Reilly&Associates, Inc.; 2003; ISBN: 1-596592-491-6

[30] Matthias Reinwarth, Klaus Schmidt: *Verzeichnisdienste - Telekommunikation Aktuell,* VDE Verlag GmbH; 1999; ISBN: 3-8007-2373-5

[31] Dieter Kluenter, Jochen Laser: *LDAP verstehen, OpenLDAP einsetzen,* dpunkt.verlag GmbH; 2003; ISBN: 3-89864-217-8

[32] Daniel J. Barret, Richard E. Silverman & Robert G.Byrnes: *LINUX Security Cookbook,* O'Reilly&Associates, Inc.; 2003; ISBN: 1-565-00391-9

[33] Charlie Scott, Paul Wolfe & Mike Erwin: *Virtual Private Networks,* O'Reilly&Associates, Inc.; 1999; ISBN: 1-565592-529-7

[34] Bill McCarty: *SELINUX - NSA's Open Source Security Enhanced Linux,* O'Reilly&Associates, Inc.; 2004; ISBN: 1-565-007161-7

[35] Rober L. Ziegler: *Linux Firewalls,* New Riders Publishing, Inc.; 2000; ISBN: 0-7357-0900-9

[36] D. Brent Chapman and Elisabeth D. Zwicky: *Einrichten von Internet Firewalls,* O'Reilly&Associates, Inc.; 1996; ISBN: 3-930673-31-2

[37] Wolfgang Barth: *Das Firewall-Buch,* Nicolaus Millin Verlag GmbH; 2004; ISBN: 3-89990-128-2

[38] Joseph Kong: *Designing BSD Rootkits - An Introduction to Kernel Hacking,* No Starch Press, Inc.; 2007; ISBN: 978-1-593271-142-8

[39] Cyrus Peikari, Anton Chuvakin (Peter Klicman, Andreas Bildstein, Gerald Richter): *Kenne deinen Feind - Fortgeschrittene Sicherheitstechniken,* O'Reilly&Associates, Inc.; 2004; ISBN: 3-89721-376-1

[40] Ryan Russel et al.: *Die mitp-Hacker-Bibel,* mitp-Verlag/Bonn; 2002; ISBN: 3-8266-0826-3

[41] Wallace Wang: *Steal This Computer Book 4.0 - What They Won't Tell You About The Internet,* No Starch Press, Inc.; 2007; ISBN: 1-59327-105-0

[42] Johnny Long: *Google Hacking - For Penetration Testers,* Syngress Publishing, Inc.; 2005; ISBN: 1-931836-36-1

[43] Thomas Bechtold, Peter Heinlein: *Snort, Acid & Co. - Einbruchserkennung mit Linux,* Open Source Press; 2004; ISBN: 3-937514-01-3

[44] Syngress Autorenteam: *Snort 2.0 Intrusion Detection,* mitp-Verlag/Bonn; 2003; ISBN: 3-6266-1304-X

Networks

[45] Charles E. Spurgeon: *Ethernet - The Definitive Guide,* O'Reilly&Associates, Inc.; 2000; ISBN: 1-56592-660-9

[46] Olaf Kirch: *LINUX - Network Administrators Guide,* O'Reilly&Associates, Inc.; 1995; ISBN: 1-56592-087-2

[47] Hal Stern, Mike Eisler & Ricardo Labiaga: *Managing NFS and NIS,* O'Reilly&Associates, Inc.; 2001; ISBN: 1-56592-510-6

[48] Paul Albitz & Cricket Liu: *DNS and Bind,* O'Reilly&Associates, Inc.; 2001; ISBN: 1-596-00158-4

[49] James M. Kretchmar: *Open Source Network Administration,* Pearson Education, Inc.; 2004; ISBN: 0-13-046210-1

[50] Douglas R. Mauro, Kevin J. Schmidt: *Essential SNMP(1.nd Ed.,* O'Reilly&Associates, Inc.; 2001; ISBN: 0-596-00020-0

[51] Douglas R. Mauro, Kevin J. Schmidt: *Essential SNMP(2.nd Ed.,* O'Reilly&Associates, Inc.; 2005; ISBN: 0-596-00840-6

[52] James D. Murray: *Wimdows NT SNMP,* O'Reilly&Associates, Inc.; 1998; ISBN: 1-56592-338-3

[53] Marshall T. Rose: *The Simple Book(2nd. Ed.),* Prentice Hall, Inc.; 1996; ISBN: 0-13-451659-1

[54] David Perkins, Evan McGinnis: *Understanding SNMP MIBs,* Prentice Hall, Inc.; 1997; ISBN: 0-13-437708-7

[55] Mathias Hein, David Griffiths: *SNMP Simple Network Management Protocol Version 2,* International Thomson Publishing GmbH; 1994; ISBN: 3-929821-51-6

[56] Peter Erik Mellquist: *SNMP++ An Object-Oriented Approach to Developing Network Management Applications,* Hewlett-Packard(TM) Professional Books; 1997; ISBN: 0-13-264607-2

[57] Marshall T. Rose: *The Open Book - A Practical Perspective on OSI,* Prentice Hall, Inc.; 1990; ISBN: 0-13-643016-3

[58] Uyless Black: *Network Management Standards(2nd.Ed.),* McGraw-Hill, Inc.; 1994; ISBN: 0-07-005570-X

[59] Wolfgang Barth: *NAGIOS - System and Network Monitoring,* No Starch Press, Inc.; 2006; ISBN: 1-59327-070-4

[60] Heinz-Gerd Hegering, Sebastian Abeck, Bernhard Neumair: *Integriertes Management vernetzter Systeme,* dpunkt.verlag; 1999; ISBN: 3-932588-16-9

[61] Walter Gora, Reinhard Speyerer: *ASN.1 Abstract Syntax Notation One(2nd.Ed.),* DATACOM-Verlag; 1990; ISBN: 3-89238-023-6

[62] Klaus H. Stoettinger: *Das OSI-Referenzmodell,* DATACOM-Verlag; 1989; ISBN: 3-89238-021-X

Embedded Systems

[63] Derek J. Hatley,Imtiaz A. Pirbhai: *Strategien fuer die Echtzeit-Programmierung,* Carl Hanser Verlag Muenchen-Wien; 1988; ISBN: 3-446-16288-7

[64] Edmund Jordan: *Embedded Systeme mit Linux programmieren - GNU-Softwaretools zur Programmierung ARM-basierender Systeme,* Franzis Verlag GmbH; 2004; ISBN: 3-7723-5599-4

[65] Michael Barr, Anthony Massa: *Programming Embedded Systems(2nd.Ed.),* O'Reilly&Associates, Inc.; 2006; ISBN: 1-596-00983-6

[66] John Lombardo: *Embedded Linux,* New Riders Publishing; 2001; ISBN: 0-7357-0998-X

[67] Craig Hollabaugh, Ph.D.: *Embedded Linux - Hardware, Software, and Interfacing,* Addison Wesley, Inc.; 2002; ISBN: 0-672-32226-9

[68] Bob Smith, John Hardin, Graham Phillips, and Bill Pierce: *LINUX APPLIANCE DESIGN - A Hands-On Guide to Building Linux Appliances,* No Starch Press, Inc.; 2007; ISBN: 978-1-59327-140-4

[69] David E. Simon: *An Embedded Software Primer,* Addison Wesley, Inc.; 1999; ISBN: 0-201-61569-X

[70] John Waldron: *Introduction to RISC Assembly Language Programming,* Addison Wesley, Inc.; 1999; ISBN: 0-201-39828-1

[71] Steve Furber: *ARM System Architecture,* Addison Wesley, Inc.; 1996; ISBN: 0-201-40352-8

Online References

Obvious, but to be written due to German-Law:
"It should be recognized, that the given links within this and following sections are solely owned by and are within the exclusive responsibility of their owners. The intention to reference to that sites is neither to take ownership of their work, nor to state commitment to any given statement on their sites. It is much more to honour the work, and thank to the help, the author advanced from by himself. Last but not

least, the intention is to support a short cut for the users of UnifiedSessionsManager to sources of required help."

OSs

[72] RedHat(TM) Enterprise Linux: *http://www.redhat.com*

[73] RedHat(TM) Enterprise Linux-Doc: *http://www.redhat.com/docs*

[74] CentOS: *http://www.centos.org*

[75] CentOS-Doc: *http://www.centos.org/docs*

[76] Scientific Linux: *http://www.scientificlinux.org*

[77] Debian: *http://www.debian.org*

[78] OpenSuSE: *http://www.opensuse.org*

[79] OpenBSD: *http://www.openbsd.org*

[80] OpenBSD-FAQ: *http://www.openbsd.org/faq/index.html*

[81] OpenBSD-PF: *http://www.openbsd.org/faq/pf/index.html*

[82] FreeBSD: *http://www.freebsd.org*

[83] NetBSD: *http://www.netbsd.org*

[84] uCLinux: *http://www.uclinux.org*

[85] Linux on ARM: *http://www.linux-arm.com*

[86] eCos: *http://ecos.sourceware.org*

Hypervisors/Emulators

kvm

[87] KVM: *http://de.wikipadia.org/wiki/Kernel_based_Virtual_Machine*

QEMU

[88] QEMU(TM): *http://www.qemu.org*

[89] QEMU-CPU support: *http://fabrice.bellard.free.fr/qemu/status.html*

[90] QEMU-User Manual: *http://fabrice.bellard.free.fr/qemu/qemu-doc.html*

[91] QEMU-OS support: *http://www.claunia.com/qemu*

[92] QEMU-Debian: *http://909ers.apl.washington.edu/ dushaw/ARM*

[93] QEMU-Debian on an emulated ARM machine; Aurelien Jarno: *http://www.aurel32.net*

[94] QEMU-Debian kernel and initrd for qemu-arm-versatile; Aurelien Jarno: *http://people.debian.org/ aurel32*

[95] QEMU-Debian ARM Linux on QEMU; Brian Dushaw: *http://909ers.apl.washington.edu/ dushaw/ARM*

[96] QEMU-Ubuntu-Installation/QemuEmulator: *http://help.ubuntu.com/community/Installation/QemuEmulator*

[97] QEMU-Ubuntu-VMwarePlayerAndQemu: *http://wiki.ubuntu.com/VMwarePlayerAndQemu*

[98] QEMU-OpenBSD: *http://141.48.37.144/openbsd/qemu.html - Dag Leine Institut fuer Physikalische Chemie, Martin Luther Universitaet Halle*

[99] QEMU-NetBSD - Running NetBSD on emulated hardware: *http://www.netbsd.org/ports/emulators.html*

[100] QEMU-OpenSolaris: *http://www.opensolaris.org/os/project/qemu/host*

[101] QEMU-OpenSolaris Networking: *http://www.opensolaris.org/os/project/qemu/Qemu_Networking*

[102] QEMUlator: *http://qemulator.createweb.de*

SkyEye

[103] SkyEye: *http://skyeye.sourceforge.net*

VMware

[104] VMware(TM) Inc.: *http://www.vmware.com*

[105] VMware-OpenBSD - HowTo Install VMWare-Tools: *http://openbsd-wiki.org*

[106] VMware-Communities: *http://communities.vmware.com*

[107] VMware-Forum: *http://vmware-forum.de*

[108] VMware-Any-Any-Patch from Petr Vandrovec: *http://kinikovny.cvnt.cz/ftp/pub/vmware*

[109] Open Virtual Machine Tools: *http://open-vm-tools.sourceforge.net*

Xen

[110] RED HAT(TM) ENTERPRISE LINUX 5.1 - Virtualization: *http://www.redhat.com/rhel/virtualization*

[111] Virtual Machine Manager - VMM: *http://virtual-manager.et.redhat.com*

[112] libvirt: *http://www.libvirt.org*

[113] Xen(TM): *http://www.xen.org*

[114] XenWiki - Xen-Networking: *http://wiki.xensource.com/xenwiki/XenNetworking*

[115] Xen-CentOS: *Creating and Installing a CentOS5 domU instance*

[116] Xen-Fedora: *http://fedoraproject.org/wiki/FedoraXenQuickstartFC6*

[117] Xen-OpenSolaris: *http://www.opensolaris.org/os/community/xen*

[118] Gerd Hoffmann: *"Install SuSE as Xen guest."*

Security

[119] OpenSSH: *http://www.openssh.org*

[120] SSH Communications Security(TM): *http://www.ssh.com*

[121] SSH Tectia(TM): *SSH Tectia*

[122] OpenSSH-FAQ: *http://www.openbsd.org/openssh/faq.html*

[123] OpenSSL: *http://www.openssl.org*

[124] OpenLDAP: *http://www.openldap.org*

[125] MIT-Kerberos: *http://web.mit.edu/kerberos/www*

[126] Heimdal-Kerberos: *http://www.h5l.org*

[127] sudo - "superuser do"(check google for actual link): *http://www.sudo.ws*

Specials

Dynagen/Dynamips

[128] Dynagen, by Greg Anuzelli: *Dynagen*

[129] Dynamips, Cisco(TM) 7200 Simulator: *Dynamips, Cisco(TM) 7200 Simulator*

QEMU-Networking with VDE

[130] VDE-Virtual Distributed Ethernet: *http://sourceforge.net/projects/vde*

[131] VirtualSquare: *http://virtualsquare.org*

[132] VirtualSquare: *Basic Networking*

PXE

[133] By H. Peter Anvin: *SYSLINUX - PXELINUX -ISOLINUX*

[134] PXE-ROM-Images: *Etherboot*

Routing

[135] "Linux Advanced Routing&Traffic Control", by Bert Hubert: *LARTC-HOWTO*

Scratchbox

[136] *Scratchbox*

Serial-Console

[137] By van Emery: *"Linux Serial Console HOWTO"*

[138] By David S. Lawyer / Greg Hawkins: *"Serial-HOWTO"*

[139] By David S. Lawyer: *"Text-Terminal-HOWTO"*

Miscellaneous

[140] IEEE: *http://www.ieee.org*

[141] The GNU Netcat: *Netcat*

[142] Netcat Wikipedia: *Netcat Wikipedia*

[143] By van Emery: *"Linux Gouge"*

UnfiedSessionsManager Versions

[144] The first public version of 2008.02.11, by Arno-Can Uestuensoez. Available with online ASC-II help only by "ctys -H print": *"UnifiedSesionsManager"* *http://sourceforge.net/projects/ctys*

[145] The second public version of 2008.07.10, by Arno-Can Uestuensoez: *"UnifiedSesionsManager"* *http://sourceforge.net/projects/ctys*

[146] The second public version with minor updates of 2008.08.6, by Arno-Can Uestuensoez: *"UnifiedSesionsManager"* *http://sourceforge.net/projects/ctys*

[147] Minor editorial updates of 2008.08.12, by Arno-Can Uestuensoez: *"UnifiedSesionsManager"* *http://sourceforge.net/projects/ctys*

[148] Enhanced documentation, 2008.08.16, by Arno-Can Uestuensoez: *"UnifiedSesionsManager"* *http://sourceforge.net/projects/ctys*

[149] Minor editorial changes, 2009.01.26, by Arno-Can Uestuensoez: *"UnifiedSesionsManager"* *http://sourceforge.net/projects/ctys*

Sponsored OpenSource Projects

[150] UnifiedSessionsManager:
http://www.UnifiedSessionsManager.org
http://sourceforge.net/projects/ctys
http://ctys.berlios.de

[151] zamklibant:
http://sourceforge.net/projects/zamklibant

Commercial Support

Commercial support and additional services are available exclusively from:

[152] Arno-Can Uestuensoez:
http://www.uestuensoez.com

[153] Ingenieurbuero Arno-Can Uestuensoez:
http://www.i4p.com