ctys(1)
Commutate To Your Sessions

UnifiedSessionsManager
Core Interface

November 28, 2010

Contents

List of Tables

List of Figures

.

1   NAME

ctys - Commutate To Your Session

2   SYNTAX

```
[-t <session-type>]
[-a <action>]
[<generic options>]
[--]
<arguments>
```

3   DESCRIPTION

The utility **ctys** is the core interface to be used within the UnifiedSessionsManager. **ctys** encapsulates the user interfaces of the whole set of supported hypervisors and native sessions into a common syntax with a single call interface.

3.1   Basics

The most important aspects for the application of ctys are:

3.2   Installation and Setup

The installation and configuration is descibed within the HowTo manual and the specific configuration guides. The basic installation should work almost from the box, once the prerequisites are present. The most valuable tool for verifying the setup of ctys is **ctys-plugins** which performs a validation of local and remote installations and displays a detailed report.

3.3   Configuration

The majority of the preset default values could be changed by configuration files and environment variables. The configuration files are installed by default within **$HOME/.ctys** directory tree.

3.4   VMs and PMs - Sessions Namebinding

The most important aspect for the introduction of a common interface is the unique namenbinding for addressing the user sessions to physical machines, supported hypervisors and guest OSs. The session in this case is defined as a representation of the lifetime for the active state of the runtime entity.
Each session could span multiple logins from multiple users, dependent from the individual setup. Thus the namebinding of the sessions and their related attributes becomes the most important feature for the management of the distributed user environments.
The specific actions required for the evaluation of the sessions namebinding are distinguished by their types:

- **HOSTs**:
  HOSTs are pure dynamic sessions, without persistent data. Thus for the namebinding the calculation of dynamic values, or in case of present instances the mapping of values and instances is required.

- **VMs**:
  VMs are based on persistent configuration data. Thus stored data from configuration files is evaluated.

- **PMs**:
  PMs are accessed by a two folded approach. First the external access to the box is performed by the addressing schema of the provided access protocol. Second, when configuration data of physical host and/or the performed guest OS is required, a single data set positioned at specific locations is evaluated.

Two basic access-approaches are provided for VMs:

- **scanning**: Scans for each call the filesystem for required configurations.

- **caching**: Queries a pre-populated cache database for required information.

The scanning approach does not require additional preparation, just some basic setup like the root-directory to start. The caching approach requires the previous scan of a list of targets and the storage of their resulting inventory data into a so called local cacheDB ctys-vhost .
The supported core utilities are:

- **ctys-vdbgen** - for automated scans

- **ctys-vhost** - as a database front-end for the cacheDB

These provide particularly the dynamic network related data for caching and offer an enhanced query interface.

3.5   Option Groups

The options are grouped into two sections:

ACTIONS
These are options directly related to specific plugins. ACTIONS are suboptions to the **-a** option and are preset by the actual loaded plugin controlled by **-t** and **-T** options.

Generic Attributes
These are the options which are mostly generic within the whole set of utilities. They influence the ACTIONS, but are related to the runtime environment.

.

4   OPTIONS

.

4.1   ACTIONS

ctys supports the following actions:

Handling of specific sessions:

CREATE: Start and/or connect to sessions

CANCEL: Finish, suspend and kill sessions

Retrieval of overall generic information:

ENUMERATE: static information about stored VMs

INFO: static details about ENABLED VMs

LIST: dynamic information about active VMs

SHOW: dynamic details about ENABLED VMs

Internal helper-methods using the official interface:

GETCLIENTPORT

4.1.1   CANCEL

```
CANCEL=(<machine-address>){1}|ALL
  (
    [FORCE|STACK][,]
    [SELF][,]
    [
      RESET
      |REBOOT
      |(INIT:<init-state>)
      |(PAUSE|S3)
      |(SUSPEND|S4)
      |((POWEROFF|S5)[:<timeoutBeforeKillVM>])
    ][,]
    [(CLIENT|SERVER|BOTH)][,]
    [TIMEOUT:<timeout-value>]
    [,USER:<user>[%<credentials>]]
  )
```

CANCEL terminates running sessions of present plugins. This includes HOSTs, VMs, and PMs. Thus beneath the handling of virtual and login sessions the management of physical machines is included. The CANCEL method includes therefore the handling of state dependencies in a hierarchical manner too. This is e.g. when a physical machine is going to be canceled - shutdown, than the contained remaining sessions are going to be canceled too. Therefore a vertical dependency is defined, which recognizes and handles the nested execution of contained sub-sessions too.

CANCEL works by default asynchronously, thus is usable in parallel for emergency shutdowns by typing quickly a short syntax.

<machine-address>
> Refer to common options parts description.

ALL
> Cancels all instances of actually loaded plugins collected by a list call.

CLIENT|SERVER|BOTH
> Selection of the VM parts to be CANCELed.

FORCE
> Cancels the stack by call to hypervisor without forward propagation. If supported, a kill call is performed after a timeout.

STACK
> Cancels a stack by forward propagation of CANCEL requests. If supported, a kill call is performed after a timeout.
>
> When handling stacks of nested PMs and VMs, the application of CANCEL action on a lower level will force contained instances to be terminated too. Thus the behaviour has to be pre-selected, whether a 'top-down' soft shutdown has to be performed, or a 'bottom-up' behaviour of instances, by killing the assigned level without recognition of contained instances. This might be appropriate e.g. in emergency cases. Two basic directions are defined:

FORCE
> As described above.

STACK
> Uses a chained approach for shutting down by a top-down behaviour. Therefore the VM-stack will be first walked up and marked by repetitive sub-calls with defined specific CANCEL suboptions.

Due to implementation specifics some remapping of inner states is performed. The mode of operation is controlled by following parameters. When not given, the default CANCEL mode is applied, which is a SHUTDOWN/STOP for servers and an UNIX-kill for client processes.

- **INIT:<init-state>**
  Mapped to UNIX init.

- **PAUSE**
  The VM will be paused immediately, remaining clients will be "UNIX-killed", not so if SERVER selected.

- **SUSPEND**
  The VM will be suspended immediately, remaining clients will be "UNIX-killed", not so if SERVER selected.

- **RESET**
  A reset is performed on any instance immediately, remaining clients will be "UNIX-killed", not so if SERVER selected.

- **REBOOT**
  Performs a "soft reset", where the instances will be given a timeout before forcing them to terminate.

- **POWEROFF[:<timeoutBeforeKillVM>]**
  A multilevel-delayed SHUTDOWN of VMs is performed.

SELF
   Includes the execution target as final system to be canceled. Else the upper stack beginning with the <machine-address> is canceled only. If no upper stack is present nothing will be done.

RESET
   Resets the target with forward propagation on stacks. RESET utilizes the native call of the hypervisor as a second step when FORCE is not set.

REBOOT
   Reboots the target, the behaviour is similar to RESET, but for the second step the hypervisor is not used. Thus a REBOOT call with FORCE flag set and SELF flag unset will not perform any CANCEL action, whereas RESET calls the hypervisor as the only action.

   A REBOOT call without FORCE flag set and SELF flag unset will perform a stack propagation, and thus in case of missing native GuestOS support for ctys the hypervisor will be called by the propagation function.

INIT:<init-state>
   Performs an native INIT on UNIX systems with forward propagation by remapping of the INIT request .

PAUSE|S3
   Calls PAUSE when supported, which is an ACPI state S3.

SUSPEND|S4
   SUSPEND has almost the same behaviour as PAUSE, the only difference is the state S4.

POWEROFF|S5[:<timeoutBeforeKillVM>])
   POWEROFF or ACPI state S5, switches into offline mode, which is actually a stand-by mode. Within UnifiedSessionsManager the configuration of WoL is pre-required in order to activate systems from state S5.

TIMEOUT:<timeout-value>
   Timeout after forward propagation of a CANCEL request, this is used for each of contained levels.

USER:<user>%[<credentials>]
   The user to be used for native access to the <action-target>.

4.1.2 CREATE

```
CREATE=<machine-address>{1}
[
   CONNECT
   |REUSE
   |RECONNECT
   |RESUME
][,]
[USER:<user>[%<credentials>]][,]
[CONSOLE:<console-type>][,]
[BOOTMODE:<boot-mode>[%<boot-img-pathname>]][,]
[(CHDIR|CD):<working-directory-change-to>][,]
[INSTMODE:
      <boot-mode>%<inst-boot-source>%<insttargetmode>%<insttarget>%<inststage>]
      [,]
[PING:(OFF|<#repetition>%<sleep>)][,]
[SSHPING:(OFF|<#repetition>%<sleep>)][,]
[STACKCHECK:<stack-check>][,]
[(STUBMODE|STUB)[:ON]][,]
[(VNCDESKIDLIST|VDIL):<list-of-custom-ids>][,]
[WAITC:<timer>][,]
[WAITS:<timer>][,]
[<callopts>][,]
[<xopts>]
```

The CREATE action starts local and remote sessions for supported plugins. The desribed part within this document is the generic superset for all plugins, where specific plugins may support less options than depicted here. This is due to specific restrictions. For example the CLI session may not require a configuration file due it's solely dynamic nature. In some exceptional cases the specific plugin may provide an additional attribute, what should be avoided when ever possible.

The CREATE has to basic modes, the operational creation of session by starting an entity and the initial creation of a session, which could be the installation of a VM. Therefore the optional parameters BOOTMODE for variation of the default boot for CREATE of a VM and the INSTMODE for the initial boot mode from a specific installation medium are provided. This is the case for the current version for some 'complete open source' solutions, where as the more commercial products are for now foreseen to handle the installation of guest systems with their own utilities.

The CREATE action is additionally supported by several utilities, for the virtual machines in particular by the guest system installer and configurator **ctys-createConfVM(1)** . This provides currently for some plugins the plug-and-play installation of virtual machines including the guest system, while for some commercial products for now the creation of additional configuration information is provided only. This temporary restriction is due to the actual amount of proprietary interfaces to be adapted, which is going to be provided. The particular support for the CREATE session is given by persistently provided default values, which omit long commandline strings and represent a distributed database for example for the mapping of user defined strings as execution shortcuts.

The main advance of the common seamless syntax for CREATE ist the unique interface syntax for the start of plugins with various frontends. This particularly provides for the various application fields with either user desktops, or server based backend setups. Therefore following in addition to the various BOOTMODEs and INSTALLMODEs, the available CONSOLEs are listed.

Another point to be mentioned here is the advanced addressing feature, which is the generic superset for all provided pluging. This so called <**machine-address**> provides particularly a common set of attributes to be used either in combination or solely - when unambitious - is provided. Again some attributes may be omitted case by case, but some in general. So the UUID is not applicable by definition for a CLI session, which is just

a remote login. Whereas it might not be available in some cases only for a VM or a PM, e.g. due to access permissions.

The CREATE action also provides options, which are listed as generic options but may effect the frame of the created session only. These are particularly related to sharing and restarting of sessions as well as for their presentation on graphical desktops. These include sizes as well as screen and workspace position.

The following given sub-options are not order dependent, the keywords are case-insensitive. For call details refer to the specific package.

<machine-address>
    Refer to common options parts description.

BOOTMODE

```
BOOTMODE:
  (
    KERNEL
     |PXE
     |FDD
     |CD
     |vHDD|HDD
     |ISO
     |INSTALL
  )
```

The BOOTMODE parameter supports the alteration between pre-configured boot setups, which is specific to the boot media and the applied hypervisors. The usage of this parameters requires a wrapper script specific to the UnifiedSessiosnManager or a specific configuration variant to be in place. For additional information refer to the test-cases and pattern within the installed directory tree in **$HOME/ctys**. The following table lists the various support options for current base-plugins.

Mode	KVM	OVZ	QEMU	VBOX	VMW	XEN
CD/DVD	OK	ffs	OK	(BIOS)	(BIOS)	OK
FDD	OK	ffs	OK	(BIOS)	(BIOS)	X
USB	OK	ffs	OK	(BIOS)	(BIOS)	X
ISO	OK	ffs	OK	(BIOS)	(BIOS)	OK
KERNEL	X	ffs	X	-	-	OK
PXE	OK	ffs	OK	(BIOS)	(BIOS)	OK
VHDD/HDD	OK	ffs	OK	OK	OK	OK

Table 1: Supported Boot/Install-Modes

Mode	KVM	OVZ	QEMU	VBOX	VMW	XEN
CD	ffs	ffs	ffs	-	-	-
DVD	ffs	ffs	ffs	-	-	-
FDD	OK	ffs	OK	-	(BIOS)	ffs
ISO	ffs	ffs	ffs	-	-	-
USB	OK	ffs	OK	(BIOS)	(BIOS)	ffs
VHDD/HDD	OK	ffs	OK	(BIOS)	(BIOS)	OK

Table 2: Supported Install-Targets

<boot-img-pathname>
    Optional pathname to an alternaternative boot image.

(CHDIR|CD):<working-directory-change-to>

Change current working directory on remote site before execution of the remote access. This is currently applicable for the X11 and CLI plugins only.

CONNECT

Connects to an existing session, else an exit with error state is performed. Therefore a new client is started.

CONSOLE

The CONSOLE supports a common user access facility, which could be an ordinary CLI interface to be used within a shell, as well as a VNC based remote desktop accesible from a X11 based local desktop. The main distinction between a CONSOLE and the HOSTs access is the direct and native access of a CONSOLE to a hypervisor facility, whereas a HOSTs session is an ordinary full-scale login into the running GuestOS.

In addition to the currently supported explicit choices, almost any type of local or remote frontend could be provided by combining DISPLAYFORWARDING, CONNECTIONFORWARDING and an explicit call by CMD suboption of CLI or X11 plugin. For examples refer to CLI-Examples and X11-Examples. Anyhow, for now the following console types are supported for the various plugins as preconfigured enumerations.

CONSOLE	CLI	KVM	OVZ	PM	QEMU	VBOX
CLI	X	X	ffs	X	X	ffs
EASYECLIPSE	ffs	ffs	ffs	ffs	ffs	ffs
ECLIPSE	ffs	ffs	ffs	ffs	ffs	ffs
EMACS	-	X	ffs	X	X	ffs
EMACSM	-	X	ffs	X	X	ffs
EMACSA	-	X	ffs	X	X	ffs
EMACSAM	-	X	ffs	X	X	ffs
FIREFOX	-	ffs	ffs	ffs	ffs	ffs
GTERM	-	X	ffs	X	X	ffs
NONE	-	X	ffs	X	X	X
RDP	-	-	ffs	-	-	X
SDL	-	X	ffs	-	X	X
VBOX	-	-	-	-	-	X
VNC	-	X	ffs	X	X	ffs
VMW	-	-	-	-	-	-
VMWRC	-	-	-	-	-	-
XTERM	.	X	ffs	X	X	ffs

Table 3: Supported Console-Types - Table 1 of 2

CONSOLE	VMW	VNC	X11	XEN
CLI	-	-	-	X
EASYECLIPSE	ffs	ffs	ffs	ffs
ECLIPSE	ffs	ffs	ffs	ffs
EMACS	-	-	X	X
EMACSM	-	-	(X)	X
EMACSA	-	-	X	X
EMACSAM	-	-	X	X
FIREFOX	X	-	-	-
GTERM	-	-	X	X
NONE	(*)	X	-	X
RDP	-	-	-	(*)
SDL	-	-	-	(*)
VBOX	-	-	-	-
VNC	(X)	X	-	X
VMW	X	-	-	-
VMWRC	X	-	-	-
XTERM	-	.	X	X

Table 4: Supported Console-Types - Table 2 of 2

INSTMODE:<boot-mode>%<inst-boot-source>%<insttargetmode>%
   <insttarget>%<inststage>

The instmode parameter manages the second stage of an installation process, once the configuration files are prepared. All parameters are mandatory, but could partly be replaced by the keyword **default**(current version requires NOT UPPER CASE). Current version supports this option for QEMU/KVM only.

- **<boot-mode>**
  The mode to boot, available modes are: CD, DVD, FDD, HDD, USB, PXE, VHDD. Where VHDD is equivalent to HDD.

- **<inst-boot-source>|default**
  The path to the install medium. This could be preconfigured within the configuration files. The fully qualified pathname is required. When **default**, the preconfigured values from the conf-file are used.
  REMARK: For now **lowercase** for **default** is required.

- **<insttargetmode>**
  The mode for the target to be installed. Available modes are: HDD, USB, VHDD.

- **<insttarget>|default**
  The path to the medium for the installed system. This could be preconfigured within the configuration files. The fully qualified pathname is required. When **default**, the preconfigured values from the conf-file are used.
  REMARK: For now **lowercase** for **default** is required.

- **<inststage>**
  The stage of installation.
    - **INIT:** The 'INIT' stage prepares the install devices by erasing them, thus has to be suppressed in case of a required reboot durcing the installation process.
    - **default:** Avoids the reset of the device contents.

PING:(ON|OFF|<repetition>%<sleep>)
   Controls whether access to the CREATE target should be verified before with the option of polling until success.

   PING is used as first to verify accesibility of the TCP stack, before the actual access permission is checked by SSHPING.

REUSE
   Basically the same as CONNECT, but creates new sessions if required.

RECONNECT

Basically the same as REUSE, with the difference, that any client session will be terminated before a new ONE is established.

When RECONNECT sub-option is given, any previously running client(and only the clients!) will be canceled before starting the new client. This could be restricted by assigning access-rights to any of current clients, which has to be handled by underlying security layer.

Therefore the hypervisor has to be configured properly for the server behaviour.

- **DISPLAYFORWARDING**
  Client an server processes run on the server machine, the display is forwarded by means of the GUI only.

- **CONNECTIONFORWARDING**
  The client process is executed on the caller's machine and connected to the server by a seperate SSH tunnel.

- **SERVERONLY**
  Not applicable.

- **LOCALONLY**
  FFS.

RESUME

Resumes a previously suspended session. This can differ between the various plugins.

SSHPING:(ON|OFF|<repetition>%<sleep>)

Controls whether actual access to the CREATE target should be verified before termination of current task.

STACKCHECK

The STACKCHECK attribute defines and/or deactivates specific pre-checks for the current VM when used in a VMSTACK-REF context.

```
STACKCHECK:
  OFF
  |
  (
   [(CONTEXT|NOCONTEXT)][%]
   [(HWCAP|NOHWCAP)][%]
   [(STACKCAP|NOSTACKCAP)]
  )
```

for additional information refer to STACKCHECK-REF and to STACKEDSESSIONS-REF.

STUBMODE

The STUBMODE suppresses the remote execution of the full ctys set, instead just a remote shell by a simple SSH call is executed. This could be utilized particularly for sessions to machines without an installed ctys.

USER:<user>%[<credentials>]

The account to be used for native access to the <action-target>.

WAITC:<timer>

Initial "sleep <timer>" after execution of client, once performed before an eventually PING and/or SSHPING.

The effect is visible for the user when operating in SYNCHRONOUS mode, else may have internal influence only.

WAITS:<timer>

Initial "sleep <timer>" after execution of server, once performed before an eventually PING and/or SSHPING.

The effect is visible for the user when operating in SYNCHRONOUS mode only, else may have internal influence only.

4.1.3 ENUMERATE

```
ENUMERATE
[=
  (
    (
      (
        [ACCELERATOR|ACCEL][,]
        [ARCH][,]
        [CATEGORY|CAT][,]
        [CONTEXTSTRING|CSTRG][,]
        [CTYSRELEASE][,]
        [DIST][,]
        [DISTREL][,]
        [EXECLOCATION][,]
        [EXEPATH|EXEP][,]
        [GATEWAY][,]
        [HWCAP][,]
        [HWREQ][,]
        [HYPERREL|HYREL][,]
        [HYPERRELRUN|HRELRUN|HRELX|HRX][,]
        [IDS|ID][,]
        [IFNAME|IF][,]
        [LABEL|L][,]
        [MAC|M][,]
        [NETMASK][,]
        [NETNAME][,]
        [OS][,]
        [OSREL][,]
        [PLATFORM|PFORM][,]
        [PM|HOST][,]
        [PNAME|P][,]
        [RELAY][,]
        [RELOCCAP][,]
        [SERIALNUMBER|SERNO][,]
        [SERVERACCESS|SPORT|S][,]
        [SSHPORT][,]
        [STACKCAP|SCAP][,]
        [STACKREQ|SREQ][,]
        [STYPE|ST|TYPE][,]
        [TCP|T][,]
        [USERSTRING|USTRG][,]
        [UUID|U][,]
        [VCPU][,]
        [VERSION|VERNO|VER][,]
        [VMSTATE|VSTAT][,]
        [VNCBASE][,]
        [VNCDISPLAY|DISP][,]
        [VNCPORT|CPORT][,]
        [VRAM][,]
      )
      [TITLE|TITLEIDX|TITLEIDXASC][,]
      [MACHINE|MAXKEY][,]
    )
  )
  [
    (REC_GEN|REC):<tab-args>
    | (SPEC_GEN|SPEC):<tab-args>
    | (TAB_GEN|TAB):<tab-args>
```

```
     | (XML_GEN|XML):<tab-args>
  ]
  [IP|DNS][,]

  [,TERSE]
  [,PKG:<pkglist>]
  [,SORT[:[ALL|EACH][%UNIQUE][%<sort-key>]]]
  [,(BASEPATH|BASE|B):<base-path>[%<base-path>]{0,n}]

  [,MATCHVSTAT:<vstat-list>]
]


vstat-list:=<vstat-enum>[%<vstat-list>]

vstat-enum:=(
     ACTIVE | DISABLED | BACKUP
     | CUSTOM | TEMPLATE | TESTDUMMY
     | EMPTY | PRESENT
     | ALL | ENABLED
     | IGNORE
     )
```

ENUMERATE collects the stored static configuration data of all VMs. Therefore the filesystem is scanned for all known types of configuration files with specific matching filters. These could be displayed in various formats and content-sets as selected by suboptions. The display could be either formatted for human display, e.g. by table output, or for machine processing in database conformant ASC-II formats. The set of content is controlled by the provided constraints for specific attributes. The enumeration is applied for local and remote sessions, where lists of start-directories for filesystem scans could be individually defined.

The data is collected for each loaded plugin with stored configuration such as VMs and PMs, dynamic plugins such as of type HOSTs have for now no individual static configuration data, thus are not included in the enumeration. These are displayed by the dynamic method LIST.

The enumeration includes beneath the filtering for specific VM attributes in addition the filtering for some dynamic operational states. This comprises additionally the actual availability of the hypervisor, which depends of the present runtime configuration. In case of Xen for example the appropriate kernel has to be running, or in case of QEMU on x86 the KVM or KQEMU kernel modules define whether QEMU is used in emulation mode only, or as KVM with kernel based acceleration. The same for KQEMU.

The scan could be either proceeded for collection of the actually executable hypervisors on each node, or for the presence of any configuration of potentially executable stored VMs. In case of multiple installed hypervisors, which could be used e.g. by the boot with a different kernel, the latter has some advantages when a test environment is to be setup. Another application is to control the exclusion of templattes and backups, which may cause some ambiguity within the database else.

The scope of the scanned states is controlled by the **vstat-enum** with the commandline attribute by the suboption **MATCHVSTAT**, which controls the output dependent of the dynamic and/or static state. Currently the following enumeration attributes are defined.

- **ALL**
  Ignores the actual availability of a specific hypervisor. This is convenient for test environments, where the hypervisor on a machine is exchanged frequently.

- **ACTIVE**
  Active, available for execution. This is the configuration file is present, and staticly enabled by the VMSTATE attribute.

- **BACKUP**
  Backup, not available for execution.

- **CUSTOM**
  Custom state, available for execution.

- **DISABLED**
  Disabled, not available for execution.

- **EMPTY**
  Empty, available for execution.

- **ENABLED**
  Active and executable by actually available hypervisor. This is the present ACTIVE state and the operational and ENABLED presence of the hypervisor in SERVER mode.

- **IGNORE**
  Ignored, not available for execution.

- **PRESENT**
  Present, not available for execution.

- **TEMPLATE**
  Template, not available for execution.

- **TESTDUMMY**
  Test, available for execution.

The output is as listed in the following record description. Some exceptions occur, when multiple interfaces are configured within a VM. Each interface is assigned with each of it's IP address to a seperate output record, containing a single MAC address and a single assigned TCP address. Thus the number of output records is increased for multihomed VMs and PMs resulting in one entry for each interface address.

Some additional values are supported for basic management of VMs by simply adding masked keywords to present configuration files and/or directories. When "-X" option is set, the output is prepared as ";" semicolon seperated list for post-processing. The same is true, when setting TERSE. The MACHINE keyword for full data set as a canonical record implies TERSE.

Processing-Key	Short Description
ALL	Sets the output to a superset of valid fields.
DNS	Transforms TCP addresses to numeric format.
IP	Transforms TCP addresses to numeric format.
MACHINE	Sets the output to the canonical full-set.
VSTAT	Alters the VMSTATE attribute, to be semantically matched.
MAXKEY	Sets the output to common subset.
PKG:<pkg-list>	Constrains on output to defined list.
REC_GEN	Activates proprietary record filter.
SORT	Activates sort filter.
SPEC_GEN	Activates record filter for visual checks.
TAB_GEN	Activates table filter.
TERSE	Output for post processing.
TITLE	Output of field names.
TITLEIDX	Output of field names with indexes.
TITLEIDXASC	Field names with additional spreadsheet column-refs.
XML_GEN	Activates XML record export filter.

Table 5: Processing Suboptions

The following table depicts the complete set of fields for current data record.

Nr	FieldKey	Fieldname	Common
1	PM or HOST	ContainingMachine	X
2	TYPE	SessionType	X
3	LABEL or L	Label	X
4	ID	ID	X
5	UUID	UUID	X
6	MAC	MAC	X
7	TCP	TCP	X
8	DISPLAY	DISPLAY	-
9	CPORT	ClientAccessPort	-
10	SPORT	ServerAccessPort	-
11	VNCBASE	VncBasePort	-
12	DIST	Guest-Distro	-
13	DISTREL	The release of the distribution.	-
14	OS	Guest-OS	-
15	OSREL	OS-Release	-
16	VERNO	VM-Config version number	-
17	SERNO	VM-SerialNo	-
18	CATEGORY	Category	-
19	VMSTATE	The state of the VM	X
20	HYPERREL	Hypervisor used for installing the VM.	X
21	STACKCAP	The capabilites supported.	-
22	STACKREQ	The list of capabilites required.	-
23	HWCAP	Offered virtual HW.	-
24	HWREQ	Required HW, either virtual or physical.	-
25	EXECLOCATION	Defines the possible execution locations.	-
26	RELOCCAP	Defines LOCATION behaviour.	-
27	SSHPORT	Alternative port for p option of SSH.	-
28	NETNAME	DNS name of current interface.	-
29	HYPERRELRUN	Release of present hypervisor.	-
30	ACCELERATOR	Present accelerator.	-
31	EXEPATH	Pathname for execution frontend.	-
32	RESERVED10	For future use.	-
33	IFNAME	Interface within the GuestOS.	-
34	CTYSRELEASE	MAGICID of the originator for each record.	-
35	NETMASK	Internet NETMASK.	-
36	GATEWAY	Internet Gateway.	-
37	RELAY	The interconnection interface.	-
38	ARCH	Architecture presented to the GuetsOS.	-
39	PLATFORM	Virtual device.	-
40	VRAM	The pre-configured amount of RAM.	-
41	VCPU	The pre-configured number of V-CPUs.	-
42	CONTEXTSTRG	A private context storage for the plugin	-
43	USERSTRING	A string to be customized by the user.	-

Table 6: Output Record-Format for MACHINE suboption

4.1.4 GETCLIENTPORT

Returns the port for attaching the front end client services to the server component.

```
GETCLIENTPORT=<label>|<id>{1}
```

This will be used internally only, or within plugins and macros. Security is based on SSH for ctys execution and the appropriate options of the current VM for restricting to local access only.
The output is presented as follows:

```
"CLIENTPORT(<type,<FQDN-host>,<vm-label>)=<client-access-port>"
```

Which could be for example:

```
"CLIENTPORT(VMW,host01.fantasy,linuxBox)=904"
```

4.1.5 INFO

Displays miscellaneous static information for the given hosts. This action is under development and is planned to be extended. Currently some OS and Machine information is displayed. Particularly the present HW-Virtualization registers of CPUs the are shown. For now the Display is given as:

```
bash-3.1$ ctys -a info -W delphi
#############################
Node:delphi.soho
System      :Linux
OS          :GNU/Linux
RELEASE     :2.6.21.6-delphi-005
MACHINE     :i686
KERNEL#CPU  :SMP-KERNEL
CPU-INFO
processor:0
vendor_id           :GenuineIntel
cpu family          :6
model               :11
model name          :Intel(R) Pentium(R) III CPU ...
stepping            :4
cpu MHz             :1266.131
cache size          :512 KB
processor:1
vendor_id           :GenuineIntel
cpu family          :6
model               :11
model name          :Intel(R) Pentium(R) III CPU ...
stepping            :4
cpu MHz             :1266.131
cache size          :512 KB

Flags assumed equal for all processors on same machine:
flags
vmx(VT-x - Pacifica)    = 0
svm(AMD-V - Vanderpool) = 0
PAE                     = 1

MEM-INFO
MemTotal            :  4018 G
SwapTotal           : 24579 G

VNC         :VNC Viewer Free Edition 4.1.2 for X - ...
wmctrl      :wmctrl is on this machine not available


------------
ctys:       :01_02_003a10
Plugings:   :   VNC
```

Figure 1: INFO Display

4.1.6 LIST

```
LIST[=
  (
    (
      (
        [ACCELERATOR|ACCEL][,]
        [ARCH][,]
        [CONTEXTSTRG|CSTRG][,]
        [CPORT][,]
        [DISPLAY][,]
        [EXECPATH|EXEP][,]
        [GROUP|GID][,]
        [HYPERRELRUN|HRELRUN|HRELX|HRX][,]
        [ID|PATHNAME|PNAME|P][,]
        [IFNAME|IF][,]
        [JOBID|JID][,]
        [LABEL][,]
        [MAC][,]
        [PID][,]
        [PM|HOST][,]
        [PNAME|P]
        [SITE][,]
        [SPORT][,]
        [TCP][,]
        [TUNNEL | (CLIENTS|C) | (SERVER|S) | (BOTH|B)][,]
        [TYPE|ST|STYPE][,]
        [USER|UID][,]
        [UUID][,]
      )
      [TITLE|TITLEIDX|TITLEIDXASC][,]
      [MACHINE|MAXKEY][,]
    )
  )
  [
    (REC_GEN|REC):<tab-args>
    | (SPEC_GEN|SPEC):<tab-args>
    | (TAB_GEN|TAB):<tab-args>
    | (XML_GEN|XML):<tab-args>
  ][,]
  [IP|DNS][,]
  [,SORT[:[ALL|EACH][%UNIQUE][%<sort-key>]]][,]
  [PKG:<pkg-list>][,]
  [TERSE][,]
  [USER:<user>[%<credentials>]][,]
]
```

LIST disyplays the realtime runtime data related to actually running local and remote sessions. Therefore the LIST action is a dynamic method in difference to the ENUMERATE action displaying static data. Even though the LIST action is focussing on dynamic data some additional static data is required for mapping purposes and completion of the human readable information. Therefore the dynamic runtime information of the plugins may contain at least some identifiers in order to access the persistently stored data within the configuration files. Also some dynamic data related to the control of jobs and their forked and parallel executed background subjobs is stored in semi-persistend caches. This is particularly required for some nested distribution of subjobs.

All loaded types are listed by filtering according to provided suboptions. The base set to be filtered is defined by the options "-t" and/or "-T". If "-t" is not present, the default "-t ALL" will be applied to all pre loaded plugins. For changing the selection scope of listed users refer to "-s" option.

The LIST action is deeply influenced by the setting of the option "-b" concerning the performance, and the option "-C" concerning the way the output data is displayed. The basic influence on the display is described in "Parallel And Background Operations" the perfomance repercussion is presented in "Performance Measures".

LIST supports various display modes, where the displayed subset of fields could be configured by switching on with the assigned keyword.

TABLES
The tables mode supports semi-fixed and generic tables. The tables are stored as macros and could be listed by call of **ctys-macros(1)** .

- **semi-fixed tables**:
These tables have a fixed number of displayed columns with limited content variation, but fully resizable column-widths.

The most important view is the TAB_TCP, which shows the following columns:

Field-Name	Content
TCP-Container	PM
TCP-guest	(MAC|TCP|DNS)
Label	<label>
ID	<ids>|<id>
Sesstype	(PM|CLI|X11|VNC|VMW|XEN|QEMU)
C	(C|S)
User	$USER
Group	<group>

—

The sizes could be defined by providing an integer width for each column seperated by "%". Contents will be truncated righthand when they extend the size of the column.

```
"TAB_TCP:7%%6%%3%%"
```

This defines the following sizes:

Field-Name(index)	Size	Default
TCP-Container(1)	7	17
TCP-guest(7)	default	17
Label(3)	6	20
Sesstype(2)	default	8
C(14)	3	1
User(12)	default	10
Group(13)	default	10

- **generic tables**
Generic tables are fully customizable. Each table has to contain at least one mandatory column. The number, order, and repetition is free to be defined. But no semantics check or display format validation is performed on the results. MACRO files are supported for storage and repetitive call.

```
TAB_GEN:<idx>_<colname>_<width>[%<idx>_<colname>_<width>]{0,n}
```

The underscore "_" is here reserved as field seperator, thus could not be used within regular values. Enclosed empty fields are not allowed.

For each column the triple of data is required:

- **<idx>**
Canonical index of field to be displayed. The value could be evaluated by calling list with the "titleidx" option.

- **<colname>**
The arbitrary name of the column to be displayed. Following restrictions apply:

* No spaces
* Printable characters only
* Size will be cut, when extends the column width.

– **<width>**
The width of the column, this excludes the border marker, which will has to be added for size calculations. each file has one additional marker, minus one from the total sum. The number of columns is 'unlimited' and could contain repetitions.

Nr.	Field-Key	Fieldname	Common
1	PM-HOST-H	ContainingMachine	X
2	TYPE	SessionType	X
3	LABEL-L	Label	X
4	ID-I-PNAME-P	ID	X
5	UUID	UUID	X
6	MAC	MAC	X
7	TCP-T	TCP	X
8	DISPLAY	DISPLAY	-
9	CPORT	ClientAccessPort	-
10	SPORT	ServerAccessPort	-
ffs	VNCBASE	VncBasePort	-
11	PID	PID	-
12	UID	UID	-
13	GID	GID	-
14	CSTYPE	C/S-Type	-
15	JOBID	JobID	-
16	IFNAME	IFNAME	-
17	RESERVED	RESERVED1	-
18	CONTEXTSTRG	CONTEXTSTRG	-
19	EXECPATH	EXECPATH	-
20	HYPERRELRUN	HYPERRELRUN	-
21	ACCELERATOR	ACCELERATOR	-
22	ARCH	ARCH	-

Table 7: Output-Format for MACHINE suboption

Processing-Key	Short Description
DNS	Transforms TCP addresses to numeric format.
IP	Transforms TCP addresses to numeric format.
MACHINE	Sets output to canonical format.
MAXKEY	Sets output to common subset.
PKG:<pkg-list>	Constrains on output to defined list.
SORT:[<sort-args>]	Activates sort filter
TAB_GEN:<tab-args>	Activate table filter
TERSE	Output for post processing.
TITLE	Output of field names.
TITLEIDX	Output of field names with indexes.
TITLEIDXASC	Output of canonical idx numbers.

Table 8: Processing suboptions

GENERIC RECORDS

Generic recors are similar to generic tables, just present a line-oriented format instead. Current supported formats are SPEC, REC, and XML.

* **SPEC** - Useful format for exploration and validation of specifications.

- **REC** - A proprietary record format.
- **XML** - XML based record format.

Raw-Output - TERSE/MACHINE

The **TERSE** mode displays a specified subset, whereas the **MACHINE** mode displays the complete set in a semicolon seperated raw format. This is particularly forseen either to be postprocessed or imported to a database or spreadsheet application. The **TITLE** and **TITLEIDX** keywords additionally display the fields contained in current record format.

4.1.7 SHOW

Displays dynamic information for the given hosts. For now the Display is given as:

```
bash-3.1\$ ctys -a show -W delphi
###############################
Node:delphi.soho
System      :Linux
OS          :GNU/Linux
RELEASE     :2.6.21.6-delphi-005
MACHINE     :i686
MEM-INFO
MemTotal          :  4018 G
MemFree           :    96 G
SwapTotal         : 24579 G
SwapFree          : 24579 G
Top         : iterations=10
top - 10:25:33 up 1 day, 12:39,  1 user,  load aver...
Tasks: 241 total,   1 running, 240 sleeping,   0 st...
Cpu(s):  0.4%us,  0.4%sy,  0.0%ni, 98.6%id,  0.5%wa...
Mem:   4018724k total, 3922936k used,   95788k fr...
Swap: 24579420k total,       4k used, 24579416k fr...

PID USER      PR  NI  VIRT  RES  SHR S %CPU %MEM ...
28241 vadmin     5 -10  388m 308m 296m S    0  7.9 ...
28246 vadmin     5 -10  388m 308m 296m S    0  7.9 ...
28247 vadmin     5 -10  388m 308m 296m S    0  7.9 ...
28248 vadmin     5 -10  388m 308m 296m S    0  7.9 ...
28249 vadmin     5 -10  388m 308m 296m S    0  7.9 ...
28250 vadmin     5 -10  388m 308m 296m S    0  7.9 ...
28251 vadmin     5 -10  388m 308m 296m S    0  7.9 ...
28252 vadmin     5 -10  388m 308m 296m S    0  7.9 ...
28253 vadmin    15   0  388m 308m 296m S    0  7.9 ...
28230 vadmin    15   0 99216  32m  16m S    0  0.8 ...
3640 root      15   0 41332  27m 3596 S    0  0.7 ...
23299 acue      15   0 45984  24m  14m S    0  0.6 ...
20011 root      15   0 42340  22m  13m S    0  0.6 ...
23147 acue      15   0 31344  22m 4448 S    0  0.6 ...
19842 root      15   0 27016  20m 4220 S    0  0.5 ...
29259 root      18   0 43420  19m  12m S    0  0.5 ...
23249 acue      15   0  113m  16m  11m S    0  0.4 ...
2939 root      15   0 21188  16m 4836 S    0  0.4 ...
19945 root      18   0  109m  15m  11m S    0  0.4 ...
23247 acue      15   0 77020  14m 9656 S    0  0.4 ...
23264 acue      16   0 85696  13m 9.8m S    0  0.3 ...
23280 acue      15   0 85696  13m 9.8m S    0  0.3 ...
HEALTH
Total ALARMs=0
```

Figure 2: LIST Display

4.2   Generic Options

-A **<ambiguity-mode>**

Allow ambiguity, this has several effects on values which may or may not be allowed to be ambiguous.

```
<ambiguity-mode>=<0|off|1|on>
```

Allow ambiguity(-A 1) or disallow(-A 0:default).

- **LABEL**: Even though the labels might be ambiguous, the IDs are not, thus an unambiguous labels only restrict the access by labels, but could be used to group sessions together, if access by IDs only is sufficient.

- **HOST**: When lists of hosts and groups are applied and resolved to redundant hosts within the list, this could be a desired circumstance or not. If not activated, redundancies in resulting group lists will be removed silently.

-b **<background-mode>**

Background and/or parallel execution. This option combines the control of detachment from console and the job distribution to multiple targets.

```
<background-mode>=
   -b
      stack
      |
      (
        (sync|off|0)|(async|on|1)
        [,
          (sequential|seq|2)
          |(parallel|par|3)
        ]
      )
```

In addition this option controls the execution of VM-Stacks, which are closely coupled to GROUPS as well as to the background mode, refer to "Stacks As Vertical Subgroups".

- **stack-mode**:
  The stack-mode is a specific enforcement of an appropriate combination for asynchronous operation sof the VMs within an sequential dependant nested VMSTACK. Therefore the values SEQ and SYNC are forced and blocked, thus could not be reset for the actual VMSTACK.
  The VMSTACK in addition decouples CONSOLE operations, though these frequently block the STDIO due to SSH only operations. The dialogue components of a VMSTACK are generally proceeded in ASYNC mode, but after the previous non-interactive task has finished. The finish of a non-interactive task is here the successful startup of a VM/PM.

- **detachment of jobs**:
  The detachment of jobs from the callers console causes the top level dispatcher to start all resulting jobs by the "-f" option of "ssh" and to return immediately. This results in a number of unmanaged jobs which implicitly are executed as autonomous parallel tasks by OpenSSH. The consequence of this is, that no higher level group functions could be performed on the whole set of results. Typical examples are
  - **CREATE**:
    CREATE is performed by default as a DETACHED job, because it just creates interactive desktop sessions which are frequently not dependent on each other.
  - **LIST**:
    LIST is performed by default as ATTACHED job, because it has some overall properties for tasks spanning multiple targets like SORTALL, where the individual sets could be intermixed.

- **execution**:
  The high-level PARALLEL execution with ATTACHED console combines both advantages. The parallel execution reduces for bulk lists of targets the overall processing time to the slowest individual by controlled dispatch. Second it keeps the overall synchronity for performing group tasks like SORT. Typical examples are

  - **CREATE**:
    CREATE is performed by default as a DETACHED job as stated before. In the case of DE-TACHED the PARALLEL property has no effect, because jobs with "ssh -f" return "successfully" straight after execution.

  - **LIST**:
    LIST is performed by default as ATTACHED job, thus the PARALLEL property has frequently a tremendous effect. When listing a a set of 20 machines where each requires about 15 seconds to scan all processes and calculate the results for each plugin, the overall processing time is reduced from 300seconds=5minutes to 15seconds.

Therefore the two properties complement each other, even though some similar effects could occur.

REMARK: The usage of "&" by shell expansion will just superpose the internal job control, which should avoided.

-c <args>

```
<args>=(
        ON
        [,(BOTH|LOCAL|REMOTE)]
        [,ONLY]
      )
       |OFF
```

The operations of ctys utilize special virtual-nameservice information which optionally could be read from the inventory database. Either from the local host, or the remote hosts allocated local database. This option controls the usage and selection of the location of these distributed caches at the local and/or remote site.
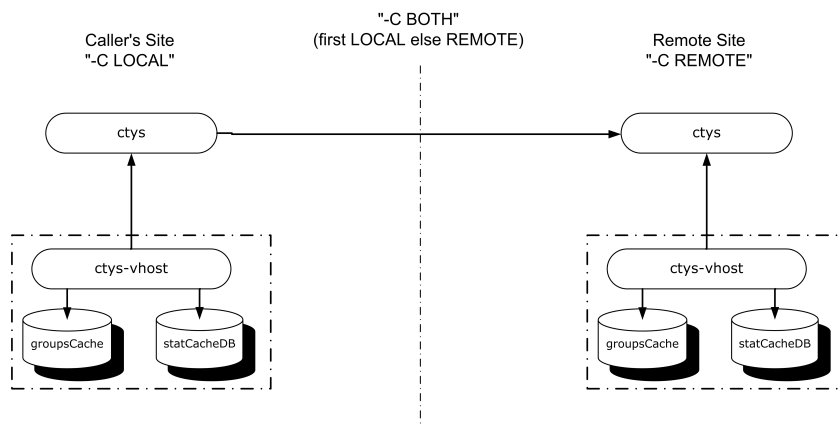
Figure 3: Distributed Caches

Even though the considerable gain of performance by usage of cached data could be abandoned, the availability of the cache facility is mandatory required for the utilisation of stacked VMs. This is due to required information of the contained environment, which is the runtime environment of the nested VM.

The UnifiedSessionsManager itself is designed as a distributed client-server system operating itself by distributing it's tasks to the managed entities, and propagation of the required states within the vertical stack-dependencies. For this tasks at almost each point of operations ctys requires access data for it's managed objects.

Therefore the UnifiedSessionsManager supports a distributed cache model consisting of multiple cache databases - cacheDBs - which has to be in sync if present, atleast might not contain inconsistent data. Partly present data will be handeled by match-priority. Alternatively the filesystem is scanned for available configuration files of VMs, once the execution target is entered.

The following flags are supported in order of control for the selection of the nameservice-caches to be used, or ignored. The default behaviour is to use "BOTH" caches optionally, and to scan the filesystem on the execution target if no cache-hit occurs. For additional information refer to Section "Distributed Nameservice and CacheDB"

- **BOTH**
  Use first local cacheDB, if no match occurs than delay for usage of cacheDB at the remote execution target. When no cache data is found, and the scanning of filesystem is still active, the filesystem on the final execution target is scanned for configuration data for plugins based ob conf-files. The cache DB is created by the tool **ctys-vdbgen(1)** . This is the default behaviour.

- **LOCAL**
  Use local cacheDB at the caller's site. The remote cacheDB on intermediate relays and on the final execution target are ignored. When no cache data is found, and the scanning of filesystem is still active, the filesystem on the final execution target is scanned for configuration data for plugins based ob conf-files. The cache DB is created by the tool **ctys-vdbgen(1)** .

  For the Local option one specific point is to be considered. The basic design and resulting implementation of the locally pre-fetched cache is based on the implementation of options evaluation with permitting repetition of the same option/suboption and the "Last-Win" philosophy. Values from caches are evaluated first, before the actual user supplied entries.

  The entry available in the local cache is resolved to a fully qualified <**machine-address**> with all of it's actually present parts, and inserted as the first argument to the relevant ACTION of the current job from the internal schedular, before the local/remote task is finally executed. Thus user supplied parts will superpose elements from the cache and may lead to desired and/or unintended deviation from the local contents of the cacheDB.

  Anyhow, the input used for evaluation of the <**machine-address**> from the cacheDB should be in sync.

- **OFF**
  Deactivates both types of caches, could only used alone.

- **ON** Activates cache, is set implicitly by all others.

- **ONLY**
  Uses cache only, no dynamic data is fetched.

- **REMOTE**
  Use remote cacheDB at the final execution target. When no cache data is found, and the scanning of filesystem is still active, the filesystem on the final execution target is scanned for configuration data for plugins based ob conf-files. The cache DB has to be created by the tool ctys-vdbgen.

-**C** <**args**>

```
<args>=(
        ON
        [,KEEP]
        [,ONLY]
```

```
           [,RAW]
           [,(FIN|FOUT):<cache-filepath>]
           [,LIFETIME:<seconds>]
           [,AUTO]
        )
        |OFF
```

For ctys two basic types of data-caches are used. The first one is the plugin specific in-mem cache, where frequent operations like "ps" for LABEL mapping will be cached for the lifetime of a process. The second is the cacheDB on-disk cacheing, which could span multiple calls to a specific executable.

The in-mem caching is active by default, because the assumption is made that the systems state might not alter relevant to ctys within a call-cycle. This could be deactivated for the common plugins by with the "OFF" option, which additionally.

The on-disk caching is used for two specific reasons.

- Enables collecting data for overall-processing like sort, where data from all remote tasks is prefetched on localhost into file system and postprocessed as one set.

- Boosts performance for repetitive access to remote data, particularly when this is required within the same task in periods of seconds. Therefore an ageing timer will be set for having "neartime" data. The variable SESSIONCACHEPERIOD (default=20seconds) controls the ageing timer.

  Even though the assumption that the systems state on a local machine is "more ore less static" within an uncritical ordinary call, this could not be said clearly for remote calls with on-disk caching. Thus on-disk caching of runtime data is off by default, except for collector actions, which do not reuse the "pure-data" cache.

  Following suboptions are applicable:
- **AUTO**
  The behaviour in case of an LIFETIME exceed is changed to automatic remove of cache data. Missing files are silently created from origination. This is particularly foreseen for the internal usage of "ctys -a LIST". The AUTO suboptions is supported for local access only.
- **FIN:<cache-filepath>**
  A user-defined cache file to be used instead of collecting remote data. This file has to contain previously cached data, which was held by usage of "KEEP" suboption. The same <cache-filepath> should be used.
- **FOUT:<cache-filepath>**
  A user-defined cache file to be used instead of default, this is the read-write runtime cache. The filepath, if relative, is relative to "$MYTMP", but no "mkdir" is called. An absolute path is used literally.
- **KEEP**
  Keeps cache-files instead of removing them before exit.
- **KEEPALL**
  Keeps cache-files of all subcalls, instead of removing them before exit.
- **LIFETIME**
  The maximum age a provided cache for "ONLY" is allowed to be. If the age exceeds, than as default the action is aborted. This behaviour is the default due to data safety.
- **OFF**
  Deactivates both types of caches, could only used alone.
- **ON**
  Activates cache, is set implicitly by all others.
- **ONLY**
  Uses cache only, no dynamic data is fetched.
- **RAW**
  Stores RAW data in cache, if not set the final results of current operation on it's actual execution-target are stored.

-d <debug-args>

```
<debug-args>=
                <debug-bit-array>[,(PATTERN|P)|MIN|MAX]
                [,(SUBSYSTEM|S):<subsystem-bit-array>]
                [,(WARNING|W):[0-9]]
                [,(INFO|I):[0-9]]
                [,(FILELIST|F):<file-list>[,(EXCLUDE|INCLUDE)]]
                [,(PRINTFINAL|PFIN)[:[0-9]]]


<debug-bit-array>=
                2#(0|1){1,32}|[0-9]*|<any-bash-format-32bit>


<subsystem-bit-array>=
                2#(0|1){1,32}|[0-9]*|<any-bash-format-32-bit>


<file-list>=
                <file>[%<file-list>]


  DEFAULT:

  -d <#integer>

  is equal to:

  -d <#integer>,MAX,WARNING:1,INFO:1
```

Sets the level and range of debug output.

- **<debug-bit-array>[,(PATTERN|P)|MIN|MAX]**
  The debug output could be controlled by one of two basic styles, the level-mode(MIN|MAX) or the match-mode(PATTERN).
  - **level-mode(MIN|MAX)**
    The level-mode sets a threshold from which on(MIN), or up to which(MAX) a trace output is displayed. The switch-on value has to be increment one above the destination output level.
  - **match-mode(PATTERN)**
    The match mode displays trace only by bitwise AND operation.

The debug mode value could be provided in any bash supported notation, but only 32bit arrays should be used.

`<debug-bit-array>=2#(0|1){1,32}|[0-9]*|<any-bash-format-32bit>`

The following variables are predefined to be used for levels and pattern.

- **ERRORS**:
  Traced independently and in any case.
  - **D_UI=1=2#1**   Common UserInterface.
  - **D_FLOW=2=2#10**   Common UserInterfaceExtended, call flow.
  - **D_UID=4=2#100**   Common UserInterfaceDebug, draft data collection.
  - **D_DATA=8=2#1000**   Detailed data processing.
  - **D_MAINT=16=2#10000=16#10**   Maintenance, details of attribute evaluation.
  - **D_FRAME=32=2#100000=16#20**   Traces the framework.

- **D_SYS**=**64**=**2#1000000**=**16#40**   Traces system calls encapsulated by "callErrOutWrapper". Particularly useful for evaluating the required root-permissions for "ksu" and/or "sudo".
- **D_TST**=16384=2#100000000000000=16#4000   Traces sync-points for regression tests.
- **D_BULK**=32768=2#1000000000000000=16#8000   This is the the haystack.

- **(SUBSYSTEM|S):<subsystem-bit-array>**
Subsystems as match-mode bitr array.

  `<subsystem-bit-array>=2#(0|1){1,32}|[0-9]*|<any-bash-format-32-bit>`

  The following variables are predefined to be used for subsystems.
  - **S_CONF**=1
  - **S_BIN**=2
  - **S_LIB**=4
  - **S_CORE**=8
  - **S_GEN**=16
  - **S_CLI**=32
  - **S_X11**=64
  - **S_VNC**=128
  - **S_QEMU**=256
  - **S_VMW**=512
  - **S_XEN**=1024
  - **S_PM**=2048

- Generic Values:
Values to be used for multiple categories.
  - **D_ALL**=65535=16#ffff   This activates all.

- **(WARNING|W):[0-9]**
Warnings to be displayed, level-mode only and no subsystem. "0" switches off. The switch-on value has to be increment one above the destination output level.

- **(INFO|I):[0-9]**
Info to be displayed, level-mode only and no subsystem. "0" switches off. The switch-on value has to be increment one above the destination output level.

- **(FILELIST|F):<file-list>[,(EXCLUDE|INCLUDE)]**
A list of files to be included exclusively or excluded. The names are matched with the presented string on output "<dir>/<file>", where due to performance reasons a simple pattern-match is performed only. For the same reason the EXCLUDE and INCLUDE keywords are applied to the whole set at once.

  `<file-list>=<file>[%<file-list>]`

- **(PRINTFINAL|PFIN)[:[0-9]]**
Prints final call assembly as passed to the execution interface and wait-points. In case of a wrapper script, the pre-wrapper-script and the final pre-execution assembly within the wrapper-script are displayed. In case of a wrapper-script the displayed call string could be used from the command line by cut-and-paste for debugging purposes.

  LEVEL defines the granularity, where by convention the LEVEL==0 represents the ultimate final call. This could be in some cases an internal library call with a bulk of subcalls of minor interest, but by definition is the final CLI execution.

-D (<display>[.<screen>])|<LABEL>

This option controls the output of local display. The values are mapped to the DISPLAY variable, but due to security resons for localhost only. Thus this is practically applicable to the display target VNC and physical multi-monitor configurations only. VMs are from the point of view for TCP/IP generally different hosts.

-f
> Force execution and ignore minor warnings. Basically no "destructive" operation, particularly nothing irreversible will be performed.

-F <remote_version>
> Force remote version.

-g <geometry>|<geometryExtended>
> The geometry for client-side representation. It is the exact syntax of X client "–geometry" parameter with an additional screen parameter as alias or index for usage with Xorg multiple displays.
>
> **ATTENTION**: In order of using xorg.conf and saving effort some minor assumptions as requirements concerning the xorg.conf file are made. Current implementation requires due to stateless filtering the field "Identifier" as first entry in "ServerLayout" sections.
>
> Supported variants:
>
> - Xorg-style: <geometry>
>   Any screen offset has to be calculated manually.
>
>   ```
>   <x-size>x<y-size>[[+,-]<x-offset>[+,-]<y-offset>]
>   ```
>
> - Xinerama-alias-style: <geometryExtended>
>
>   ```
>   <geometryExtended>:=
>     <Xorg-style>
>     [:[<ScreenSection>|<ScreenIndex>]
>       [:[<ServerLayout>]
>         [:[<alternateConfigFile>]
>         ]
>       ]
>     ]
>   ```
>
> All values are evaluated and calculated at the callee's site, thus has to be in conformance of the actual targeted XServer, which e.g. could be a remote DISPLAY in case of DISPLAYFORWARDING.
>
> - **<x-size>x<y-size>:Screen4**
>   The screen from the first ServerLayout section with given Screen section name as alias will be used. The required offsets will be calculated from the "/etc/X11/xorg.conf" file.
>
> - **<x-size>x<y-size>:Screen4:Layout[0,1]**
>   The screen from the LayoutSection named "Layout[0,1]" with given Screen section name as alias will be used. The required offsets will be calculated from the **/etc/X11/xorg.conf** file.
>
> - **<x-size>x<y-size>:4:Layout[0,1]**
>   The screen from the LayoutSection named "Layout[0,1]" with given Screen index will be used. The required offsets will be calculated from the **/etc/X11/xorg.conf** file.
>
> - **<x-size>x<y-size>:4:Layout[0,1]:$HOME/myScreenLayout**
>   The same as before, but with it's own Xinerama configuration file. This could particularly applied in case DISPLAYFORWARDING to a remote screen array, where the two main alternatives for alias usage on remote displays are either the usage of a specific screen section within the locally used configuration file and/or a seperate configuration file only used for remote DISPLAY targets.

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

-H **<help-options>**
> The extended help option is based on system interfaces for display of manpages, PDF and HTML documents. This comprises the man pages and installed manuals.
>
> For additional help refer to the documents chapter **Online-Help** or type **ctys -H help**.

-j **<job-id>**
> The ID of current job. This is an internal call and therefore should just used by developers for test purposes. Any variation of the JOB_IDX for the CLI call may severe job execution seriously and even can damage user data when set for a CANCEL operation.
>
> The originating CLI call should not use this option in productive operations, any subcall may have a propagated value as required. The value within the originating interactive CLI call is set to "JOB_IDX=0". This is the value for the starting point of internal task-scheduler data, and thus the index for the first performed task too. The value is evaluated by plugins for handling job specific PROLOG and within EPILOG for decision of the state of passed job and eventual required post-processing.
>
> In addition a variable "CTYS_SUBCALL" is set.
>
> - "-j $DATETIME:$JOB_SUPER"
> - default:
>   - "CALLERDATETIME=$DATETIME"
>   - "CALLERJOB_IDX=$JOB_IDX"
>   - "CTYS_SUBCALL=$1"

-l **<login-name>**
> The users, which will be used for hosts without an exlicitly given user. The hosts/groups entries provide the common EMail-Style "<user>@<execution-target>". The default is "$USER", when neither "-l", nor an explicit user is provided.

-L **<execution-location>**

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

> This option controls the location and possible split of the involved client and server parts of current session. When connecting a user interface with it's server components the following basic constellations could be distinguished:
>
> - **LOCALONLY**
>   Client and server components are coallocated on users workstation, the display is driven locally.
> - **DISPLAYFORWARDING**
>   Client and server components are coallocated on server, and the display is forwarded to the users workstation.

- **CONNECTIONFORWARDING**
  Client and server are split, whereas the client component is located on the users workstation and the server component is located on the server machine. The connection from server/client component is forwarded on application protocol level.

- **CLIENTONLY**
  This is a client in standalone mode, still used internal only, e.g. for COPY when performing phase-2 with remote-copy from local client.

- **SERVERONLY**
  This is a server in a so called headless-mode.

-M <message>
Free text to be used as prefix for target exec. It will be printed before output.

-n
Just display, do not execute. For test only.

-p <db-directory-path-list>
Path list to directories containing DBs for name resolution, same for each <db-directory-path> as for ctys-vdbgen. ctys will internally handle names by multiple levels of resolution, which depends on the actual executing plugin. The most sophisticated address resolution is frequently required for VMs when using them in a roaming manner on groups of machines, where after some plugin specific resolution of convenient VM-addressing by user an TCP/IP service-access-point for OpenSSH has to be addressed.

The second case to be handled is the addressing of execution entities of type HOSTs transparently within a VM or PM. This will be supplied by ctys-nameservice too. For almost all of the nameservice tasks additionally required for plugin-specific address resolution actions the ctys-vhost command is used internally. This option sets the databases for operations of ctys-vhost. If not present ctys-vhost defaults will be applied.

-P
Use default <db-directory-path>.

-r <xsize>x<ysize>
Remote resolution, which is by default the same as local client size given by "-g" option. This configures the virtual graphic card of the server with the provided resolution. This parameter is not applicable to any application. It has to be defined in the application specific package. The current supported applications are:

- VNC: for vncserver

-s <scope>
Restrics/expands the scope/selected set for mode of operations.

- **USER**
  Own sessions.
- **GROUP**
  Sessions of own group.
- **USRLST=(usr1[,usr2][,...])|all**
  Given list.

- **GRPLST=(grp1[,grp2][,...])|all**
  Given list.

- **ALL**
  This value is remapped to "USRLST=all".

-S [(on|off)][,][<ignore-signal-spec>
   Sets the signal spectrum to be ignored. The values are accepted as numeric values only. Applicable values could be displayed by "trap -l"(within bash).

   The default values are "1,3,19", which is set for CLI0 consoles only by default.

   In case of CLI, generally for any multi-session call, it has to be considered thoroughly whether and which signals could be set.

-t <session-type>[=suboptions]
   Defines the context of execution and the resulting applicable feature set. This could be a flat endpoint-user-session in case of VNC, or a virtual OS-starter in case of a VM session e.g. in case of VMW or XEN. Suboptions specify more detailed characteristics.

   Thus this parameter has to be set first. To load multiple plugins for one call, the environment variable CTYS_MULTITYPE or the "-T" option could be set.

- **<session_type>(default:VNC)**

  - **VNC**
    Remote VNC sessions, calls the scripts ctys-callVncserver and/or ctys-callVncviewer. The specific behaviour is here to set a password for the VNC session from a passwd-file via CLI option. The access rights of this stored passwd in **$HOME/.vnc/passwd** should be checked.

  - **QEMU/KVM, (VBOX), VMW, XEN, PM, VNC, X11, and CLI**
    For details of additional types refer to the specific plugins. For now supported are QEMU/KVM, (VBOX), VMW, XEN, PM, VNC, X11, and CLI.

-T <session-type>[,<session-type>[,...]] | all
   Preloads given list of <session_type> instead of loading the plugins of requested types by "-t" option. Alternatively the environment variable CTYS_MULTITYPE could be pre-set, which has the same result. If CTYS_MULTITYPE and the "-T" option are provided, the option has priority.

   This option is required for the scope control of generic actions, which generally will be applied by calling of all current loaded <session_type> interfaces. E.g. the "-a LIST" action lists active sessions for all actually loaded <session_type>. For display of current active sessions of all available <session_type>, the "-T all" has to be used.

- **<session_type>**
  The name of a dynamic loaded plugin, which is the <session_type>. For now supported are QEMU/KVM, (VBOX), VMW, XEN, PM, VNC, X11, and CLI.

- **all**
  Tries to load all present plugins, this would frequently fail, when the configured resources of bash are exceeded. This could be even caused by a single module, which exhausts available resources - as in any existing system. Thus the default will be set to requested types by "-t" options or NIL by default.

-v

> Show version. Current version scheme is as follows:

-V

> Show version. Current version scheme is as follows:
>
> - **AA_BB_CCC[[abc]DD]**
>   - **AA**: Official major upgrades.
>   - **BB**: Official minor upgrades.
>   - **CCC**: Build <-> Test versions.
>   - **abc**: Development versions, Test-States:
>   * **[a]**lpha
>   * **[b]**eta
>   * **[c]**(g)amma
>   - **DD**: Pre-Release development versions. Anyway, if publicly available might be yet almost stable. This option strongly interacts with the "-X" option, when set only the version number is display, without a <CR>. This is the only relevant information for batch-processing. Else all current loaded components - libraries + CORE-plugins + Application-plugins - are listed.
>   Using this option twice shows in addition to the plugin short-names the actual file of storage. Sub-packages loaded by Application are contained in the list too.
>   This list is generated at the end of execution, thus on-demand-loaded sub-packages are listed too, as far as the have been demanded during current call. The set of the "on-demand-loaded" plugins can vary in dependency of the actual performed control flow.
>   When using this option alone, only the initial by default-loaded-components are listed.

-W <WorkspaceId>|<WorkspaceLabel>

> This parameter requires the tool 'wmctrl' to be present, if not the usage is not provided and an error message is generated before exiting ctys.
>
> When provided by system and successfully detected, the following applies:
>
> - **<WorkspaceId>**
>   The id of the desktop to be used for placing the window. Currently 1-3 digits are supported.
>
> - **<WorkspaceLabel>**
>   The user defined label of the desktop to be used for placing the window. When beginning with a digit, and is shorter than 4 characters, at least one character has to be a non-digit, otherwise it will be detected as <WorkspaceId>. Currently special characters like '&' are not supported, so just digits and ordinary characters and hyphens should be used.
>
>   A list of current desktops could be shown by calling: "wmctrl -d". Where the first column is the id of the desktop, and the last is the label. For further information on wmctrl refer to related man page.

-x <OFF|STAR|CHAIN>

> Defines the resolution of required nested access to a chain of servers. This is e.g. the case when from the machine CLIENT01 the HOST01 is accessed. When now HOST01 required some information from the HOST02 there are basically the options to fetch theses from HOST01 or reply with an open issue and fetch the information from CLIENT01. For requirements exist both cases, but for the common task of user-logins the SSO-keys are only present on the CLIENT01, atleast in case of OpenSSH the ssh-agent might run on CLIENT01 only. When using Keberos with the ticket forwarding option, the CHAIN option should be prefered.
>
> - **OFF**
>   No remote resolution is performed. The application should provide for appropriate output.

- **CHAIN**
  Chained access, which is the immediate forwarding of requests form the site where these occur.
- **STAR**
  STAR access, which is the forwarding of requests form the site initiating machine only, remote issues have to be replied first.

-X

Generate terse output for post processing. The '-v Verbose' flag is not effected and should be only used for testing.

-y

Activates some terminal capabilities, mainly coloring of ERROR, WARNING, and WARNINGEXT. Very handy when debugging, but not yet supported for Emacs-Consoles. As an alternate the variable "CTYS_TERM_COLORS" could be set to "0". When selected the local and remote settings are both set at once. In current version this is set by default when the variable TERM is set to "xterm".

-Y

Activate **'ForwardAgent yes'/'-A'** of OpenSSH. If not active, another SSH authentication is required for each hop. When no mechanism like keys or Kerberos is in place, password authentication will be used. The basic configuration of SSH has to be prepared appropriately.

-z (NOPTY|PTY|1|2)

Control the allocation of a pseudotty by ssh. Therefore one or two "-t" options could be set for the internal "ssh" call.

```
NOPTY    : Eliminates "-l" of standard bash-call
           and "-t" for ssh-call.
PTY      : "-t"
PTY,PTY  : "-t -t"
1        : "-t"
2        : "-t -t"
```

-Z (KSU|NOKSU|SUDO|NOSUDO|ALL)

Controls call permission-grant. The calls requiring impersonation to another users ID, frequently "root" for restricted system resources, are supported to use "ksu" and/or "sudo". This option replaces the default settings from the configuration file. The mechanisms could be switched on/off selectively.

```
KSU      : use Kerberos
NOKSU    : do not use Kerberos (DEFAULT)
SUDO     : use sudo
NOSUDO   : do not use sudo      (DEFAULT)
ALL      : use al provided
```

As an persistent alternative following environment variables could be pre-set.

```
USE_KSU (0=>off 1=>on)          (DEFAULT:0=>off)
USE_SUDO (0=>off 1=>on)          (DEFAULT:0=>off)
```

The evaluation is implemented as a generic check for first match of hard-coded call-check. The following order of permission tests is performed for each system callee.

- user is root
- native access granted
- ksu call
- sudo call

In case of 2.) the current ID is checked for "$USER==root", if not, than a warning is generated, but continued with procedure. This is due to possible security flaws, when assigning root-ID to an ordinary user. Anyhow, when using ctys from a system account, this might be OK.

REMARK: When a user cannot be authentified by one of sudo or ksu, then the system waits for a user interaction. BUT, due to internal "silent" checks the stdio was redirected, of course. Thus the system seems to be "hanging", or requests "Password:" and seem not to continue afterwards, but it "may work". This is a "natural dilemma", because within the generic check function called for each task several times the output has to be suppressed. Currently no detection for an exceptional User-Dialog request is implemented. So for now are two diagnosis facilities implemented:

- A warning as **HINT** is generated, when "-**w**" option for extended warning is set. This shows the wrapped native call, which should be called manually by cut/paste on the ACTUAL EXECUTING system.
- The system variable **CTYS_NOCALLWRAPPER** could be set, which deactivates the wrapping of stdio and stderr for the call wrapper only.

This means, that some redirection for the call context is still active, because it is a required output data, or is simply bulk data which might flaw the whole sense of diagnosis.

When typing a RETURN the process will continue, but disabling the current type of permission mechanism. This could be OK, when KSU and SUDO is set, and KSU has no permissions configured, but SUDO has. It could lead to an later error too, when none could be detected. This scenario occurs for:

- **KSU**: Kerberos credential was timed-out.
- **SUDO**: User and/or call are not configured in sudo for execution target.
- On client machine no permissions for system calls are configured. This case can frequently be ignored safely.
- The only case where this can lead to an error is the missing permission for access to a proprietary client application.

  When using "**sudo**" the flag "**requirestty**" within "/**etc**/**sudoers**" control whether a TTY is required or not. When in order to avoid this uncomment the flag within sudoers file. The "-**z**" option could be used to activate a pseudotty.

5 ARGUMENTS

```
[--]                                              \
['('<any-options global for all remote>')']  \
((([user@]<hostname>)|<groupname>)['('<any-options>')'][ ...]
```

These are the remote options which are given as global and individual options for each host. The options are (almost) the same as for common call.

ATTENTION:

[−−]

> is required when using remote options, otherwise some problems with standard remote options might occur.

<user> default:\$USER=\$USER

> When instead the "-l" option likewise the r-functions is supported the given user(list) is permutated with the listed hosts. Particularly nice for bulk-tests, but anyhow a limit of about 20 sessions to individual hosts (IP-addresses) is hardcoded to avoid some "hard-coded-resets". This value could be reset by following environment variable:
>
> ```
> R_CREATE_MAX=${R_CREATE_MAX:-20}
> ```

<hostname> default:'uname -n'(`uname -n`)

<groupname>

> Any user defined group/macro, for additional information refer to ctys manual.

<any-options>

> Any global option could be provided individually for each host. E.g. individual debugging level on that host only. One implementation specific to be aware of is, that these options are superposed, but not reset, thus the current environment will remain for the following host. The following example shows three hosts, where each has a different debugging level. First of all the debugging flag and level is not forward propagated, and as common for all other environment settings too, "the last wins".
>
> ```
> ...-d 6 -- ( -d 3 ) host01 host02( -d 1) host02(-d 0)...
> ```
>
> So, the given options results in the following scenario:
>
> ```
> -> localhost:   -d 6            = -d 6
> -> host01:      -d 3            = -d 3
> -> host02:      -d 3 -> -d 1   = -d 1
> -> host03:      -d 3 -> -d 0   = -d 0
> ```

<command>

> Command to be executed on the target host, which could be a native physical providing a remote desktop based on VNC, or a virtual machine like Xen. In any case the login will be performed by means of the target system, but the administrative support for seamless execution is provided by this tool.

6 SEE ALSO

ctys use-cases
  ctys-uc-IPMI(7) , *ctys-uc-WoL(7)*

ctys plugins PMs
    ctys-PM(1)

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

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

ctys executables
  ctys-distribute(1) , *ctys-extractARPlst(1)* , *ctys-extractMAClst(1)* , *ctys-genmconf(1)* , *ctys-plugins(1)* ,
  ctys-vping(1) , *ctys-vdbgen(1)* , *ctys-vhost(1)* , *ctys-wakeup(1)*

system executables *dmidecode(8), ether-tool(8), ether-wake(8), nc(1)<a.k.a. netcat>*

7 AUTHOR

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

8 COPYRIGHT

Copyright (C) 2008, 2009, 2010 Ingenieurbuero Arno-Can Uestuensoez
This is software and documentation from **BASE** package,

- for software see GPL3 for license conditions,

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

For additional information refer to enclosed Releasenotes and License files.