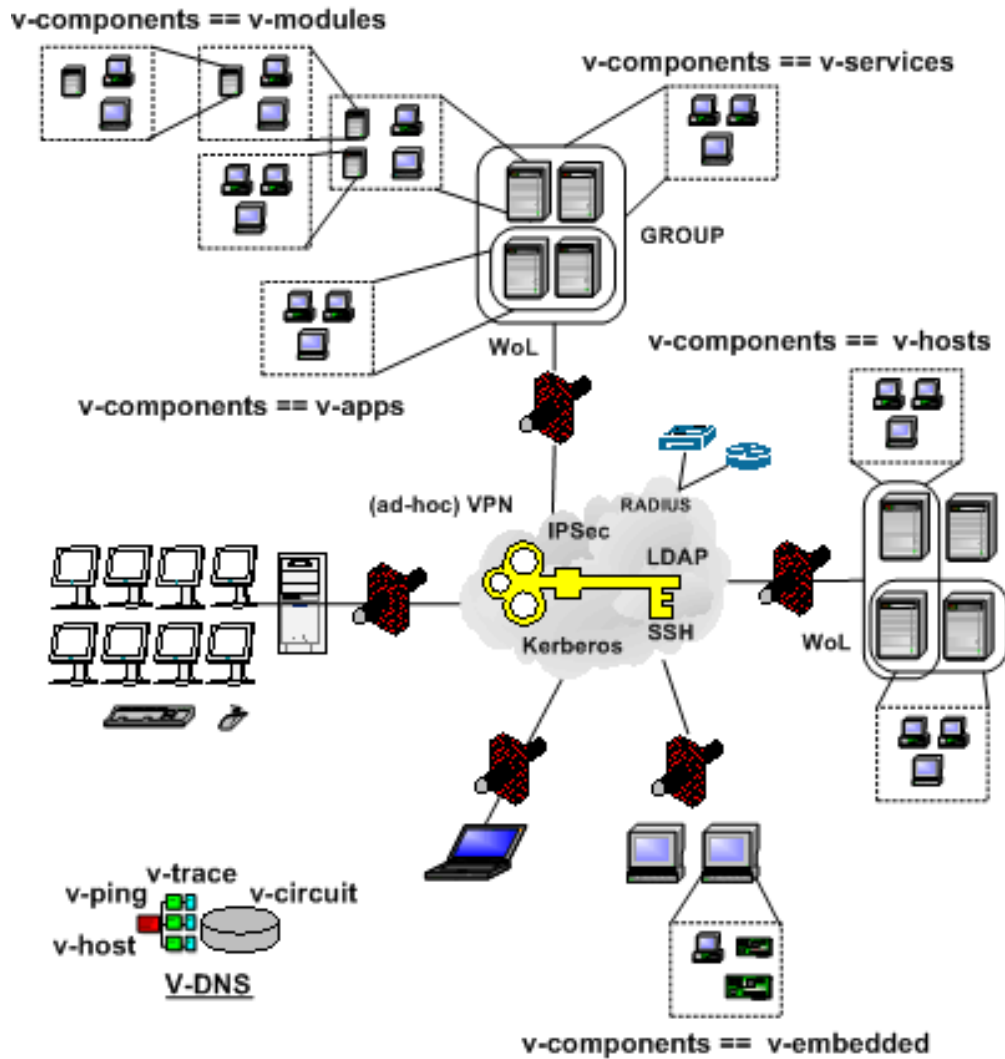


stacked-VMs == v-components



The UnifiedSessionsManager

CTYS-COMMAND-REFERENCE

PDF-PRINT-HELP

Version:01.12.002 - 2011.12.06

Release: International

Copyright 2008,2009,2010,2011

Arno-Can Uestuenseoz



Published by
Ingenieurbuero fuer Telekommunikations und Software-Systemloesungen
Arno-Can Uestuensoez

Zentnerstr. 34
D-80798 Muenchen
Germany

Phone:+49.89.27817287

<https://arnocan.wordpress.com/>
<http://ctys.sourceforge.net/>

Copyright (C) 2008,2009,2010 by
Ingenieurbuero fuer Telekommunikations und Software-Systemloesungen
Arno-Can Uestuensoez

Licenses:
Software: GPL3
Basic-Dokuments: GFDL-1.3
Concepts+Interfaces+Documents: CCL - cc by-nc-nd

The following text is required due to formal reasons, if you are going to use these artifacts for your personal private purposes only you may probably not require to analyse it in detail. The application by scholars, students, and apprentices is specifically desired.

The reason of the introduction of this distinction is the experience of repetitive and ongoing unfair competition with 'criminal elements' contained, having a vast effect on the re-financing of the further development for this project. Thus this step could be named 'self-defence'.

Software: GPL3

UnifiedSessionsManager - ctys - Commutate To Your Sessions
Copyright (C) 2010 Engineering Office Arno-Can Uestuenseoz.

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 3 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program; if not, see <<http://www.gnu.org/licenses/>>.

Basic-Documents: GFDL-1.3

Copyright (C) 2010 Engineering Office Arno-Can Uestuenseoz.
Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.2 or any later version published by the Free Software Foundation; with **Invariant Sections** - the **whole document content**, **whole front cover**, and **whole back cover**. A copy of the license is included in the section entitled "GNU Free Documentation License".

Concepts+Interfaces+Documents:
CCL - Creative Commons License - Non-Commercial, No-Derivs

Copyright (C) 2010 Engineering Office Arno-Can Uestuensoez.

Attribution-Noncommercial-No Derivative Works 3.0 Unported

You are free: to Share and to copy, distribute and transmit the work
Under the following conditions:

Attribution: You must attribute the work in the manner specified by the author or licensor (but not in any way that suggests that they endorse you or your use of the work).

Noncommercial: You may not use this work for commercial purposes.

No Derivative Works: You may not alter, transform, or build upon this work.

With the understanding that:

Waiver: Any of the above conditions can be waived if you get permission from the copyright holder.

Public Domain: Where the work or any of its elements is in the public domain under applicable law, that status is in no way affected by the license.

Other Rights: In no way are any of the following rights affected by the license:

1. Your fair dealing or fair use rights, or other applicable copyright exceptions and limitations;
2. The author's moral rights;
3. Rights other persons may have either in the work itself or in how the work is used, such as publicity or privacy rights.

Notice:

For any reuse or distribution, you must make clear to others the license terms of this work. The best way to do this is with a link to this web page.

Contents

I	Common Basics	13
1	Preface	15
1.1	History	15
1.2	Contact	16
1.3	Legal	16
1.4	Acknowledgements	17
2	Feature Specification	19
2.1	Feature-Sum-Up	19
2.1.1	Supported Hypervisors	20
2.1.2	Tested GuestOS support	21
2.1.3	Supported Native Plugins	25
2.1.4	Tested Client OSs	31
II	User Interface	35
3	Common Syntax and Semantics	37
3.1	General CLI processing	37
3.2	Options Scanners - Reserved Characters	38
3.3	Hosts, Groups, VMStacks and Sub-Tasks	38
3.3.1	Common Concepts	38
3.3.2	Flat Execution-Groups by Include	39
3.3.3	Structured Execution-Groups by Sub-Tasks	40
3.3.4	Stacks as Vertical-Subgroups	40
3.3.5	VCircuits as Sequentially-Chained-Subgroups	44
3.4	CLI macros	44
3.5	Common Options	46
4	Core Data	49
4.1	Overview	49
4.2	Standard Configuration Files	50
4.3	Common Data Fields	51
4.4	Common Processing Options	57
4.5	Specific Variations	60
4.6	Generic Tables	60
4.7	Generic Records	61
5	Address Syntax	63
5.1	Basic Syntax Elements	63
5.2	SyntaxExamples	65
5.3	AddressSyntaxElements	66
5.4	Stack Addresses	69
5.5	Group Resolution	70
5.6	ctys-help-on	72

6	CTYS Call interface	75
6.1	Common Elements and Behaviour	75
6.1.1	Sessions Namebinding	75
6.1.2	Cache for Performance and Collections	75
6.1.3	Access VM Stacks	76
	The Stack Essentials - CREATE and CANCEL	78
6.2	ctys	79
6.2.1	Installation and Setup	79
6.2.2	Configuration	79
6.2.3	VMs and PMs - Sessions Namebinding	79
6.2.4	Option Groups	80
6.3	OPTIONS	80
6.3.1	Actions	80
	CANCEL	81
	CREATE	84
	ENUMERATE	90
	GETCLIENTPORT	96
	INFO	97
	LIST	98
	SHOW	102
6.3.2	Generic Options	103
6.4	ARGUMENTS	119
7	Plugins - Feature Extensions	121
7.1	Prerequisites	121
7.2	Category HOSTs	123
7.2.1	CLI	123
7.2.2	RDP	126
7.2.3	VNC	128
7.2.4	X11	133
7.3	Category PMs	136
7.3.1	ctys-PM	136
7.4	Category VMs	141
7.4.1	ctys-QEMU	141
7.4.2	ctys-VBOX	146
7.4.3	ctys-VMW	150
7.4.4	ctys-XEN	155
7.5	Category GENERIC	161
7.6	Category Internal	161
7.6.1	DIGGER - Forwarding Encrypted Connections	161
	Encrypted Tunnel Support	161
	LIST Tunnels	162
8	Support Tools	163
8.1	ctys-attribute	164
8.2	ctys-beamer	167
8.3	ctys-cloneVM	173
8.4	ctys-config	177
8.5	ctys-convertVM	179
8.6	ctys-createConfVM	180
8.7	ctys-distribute	189
8.8	ctys-dnsutil	194
8.9	ctys-extractARPlst	197
8.10	ctys-extractMAClst	199
8.11	ctys-genmconf	201
8.12	ctys-getNetInfo	204
8.13	ctys-groups	206
8.14	ctys-install	209
8.15	ctys-macmap	212
8.16	ctys-macros	214

8.17	ctys-plugins	217
8.18	ctys-scripts	222
8.19	ctys-vnetctl	224
8.20	ctys-smbutil	231
8.21	ctys-vboxutils	233
8.22	ctys-vdbgen	235
8.23	ctys-vhost	241
8.24	ctys-vmw2utils	267
9	CTYS Call interface	269
9.1	Common Elements and Behaviour	269
9.1.1	Sessions Namebinding	269
9.1.2	Cache for Performance and Collections	269
9.1.3	Access VM Stacks	270
	The Stack Essentials - CREATE and CANCEL	272
9.2	ctys	273
9.2.1	Installation and Setup	273
9.2.2	Configuration	273
9.2.3	VMs and PMs - Sessions Namebinding	273
9.2.4	Option Groups	274
9.3	OPTIONS	274
9.3.1	Actions	274
	CANCEL	276
	CREATE	279
	ENUMERATE	285
	GETCLIENTPORT	291
	INFO	292
	LIST	293
	SHOW	297
9.3.2	Generic Options	298
9.4	ARGUMENTS	315
9.5	ctys-wakeup	316
9.6	ctys-xdg	319
9.7	ctys-xen-network-bridge	322
9.8	ctys-utilities	324
III	Appendices	327
10	Current Loaded Plugins	329
11	File Formats	335
11.1	Common Tools and Formats	335
11.2	Groups	336
11.3	Macros	338
11.4	Static Import File Databases - fdb	338
11.4.1	macmap.fdb	338
11.4.2	enum.fdb	339
11.5	Static Pre-Fetch Cache Databases - cdb	339
11.5.1	statcache.cdb	339
11.6	Dynamic Runtime Cache Databases	340
11.7	Configuration Entries for ctys	340
11.7.1	Actual Processing vs. Administrative Display	340
11.7.2	Configuration File Variants	340
11.7.3	Keywords	341
11.7.4	Interface Keywords	344
11.7.5	MAGICID	346
11.7.6	VMSTATE	346
11.7.7	Virtual Hardware-Platform	347
11.7.8	Stacking Entries	349

11.7.9 Execution Location and Relocation	349
12 Miscellaneous	351
12.1 Basic EXEC principle	351
12.2 PATH	351
12.3 Configuration files	352
13 LICENSES	353
13.1 CCL-3.0 With Attributes	353

List of Tables

2.1	Color coding of implementation and test states.	19
2.2	Supported Hypervisors	20
2.3	Getestete GuestOS	21
2.4	Tested GuestOS	22
2.5	Tested GuestOS	23
2.6	Tested GuestOS	24
2.7	Native Plugins vs. OS-Distribution	25
2.8	Native Plugins vs. OS-Distribution	26
2.9	Native Plugins vs. OS-Distribution	27
2.10	Durch HOSTs Plugins Unterstützte Clients und Desktops	28
2.11	Durch HOSTs Plugins Unterstützte Clients und Desktops	29
2.12	Unterstützte HOSTs-Plugin Sub-Komponenten	29
2.13	Unterstützte Server basierte VMs plugins	29
2.14	Unterstützte Host basierte VMs plugins	30
2.15	Getestete ClientOS	31
2.16	Getestete ClientOS	32
2.17	Getestete ClientOS	33
4.1	Supported File-Extensions	50
6.1	Supported Boot/Install-Modes	86
6.2	Supported Install-Targets	86
6.3	Supported Console-Types - Table 1 of 2	87
6.4	Supported Console-Types - Table 2 of 2	87
6.5	Processing Suboptions	94
6.6	Output Record-Format for MACHINE suboption	95
6.7	Output-Format for MACHINE suboption	100
6.8	Processing suboptions	101
7.1	Prerequisites and applicability of plugins	121
8.1	Supported Attribute Namespaces	165
8.2	Clone Automation	174
8.3	Installation Automation	182
9.1	Supported Boot/Install-Modes	281
9.2	Supported Install-Targets	281
9.3	Supported Console-Types - Table 1 of 2	282
9.4	Supported Console-Types - Table 2 of 2	282
9.5	Processing Suboptions	289
9.6	Output Record-Format for MACHINE suboption	290
9.7	Output-Format for MACHINE suboption	296
9.8	Processing suboptions	296

List of Figures

3.1	Subtask	39
3.2	Groupresolution by Include only	39
3.3	Groupresolution by Subgroups	40
3.4	Combined Subgroups and Substacks	41
3.5	Stack Example for Basic Call-Interface	43
3.6	CONSOLE- and HOSTs-Asynchronity for Stacked-Execution	43
3.7	VCIRCUIT	44
5.1	TAE - Target Application Entity address	66
5.2	Machine-Address	67
5.3	<access-ports>	67
5.4	CONSOLE types	67
5.5	Stack-Address	69
5.6	Group-Address	70
5.7	Groups of Stack-Addresses	71
6.1	INFO Display	97
6.2	LIST Display	102
6.3	Distributed Caches	105
8.1	ctys distributed access	183
8.2	ctys SW-Layers and plugins	217
8.3	QEMU/KVM-Network Interconnection	224
8.4	QEMU/KVM-Network Interconnection - Components	225
8.5	QEMU/KVM-Network Interconnection - Details	226
8.6	Database creation and application	236
9.1	INFO Display	292
9.2	LIST Display	297
9.3	Distributed Caches	300

Part I
Common Basics

Chapter 1

Preface

1.1 History

Version	Date	Author	Description
01.03.003.a01[?]	2008.02.11	Arno-Can Uestuensoez	Initial pre-release as embedded printable help
01.07.001.a01[?]	2008.08.03	Arno-Can Uestuensoez	First major update with numerous additions and partial review.
01.07.001.b02[?]	2008.08.11	Arno-Can Uestuensoez	Minor editorial updates. A lot of tests, some fixes.
01.07.001.b03[?]	2008.08.12	Arno-Can Uestuensoez	Minor editorial updates.
01.07.001.b04[?]	2008.08.16	Arno-Can Uestuensoez	Enhancement of documentation and Web-Site.
01.11.001[?]	2010.04.25	Arno-Can Uestuensoez	Major enhancements and updates.
01.11.002	2010.05.24	Arno-Can Uestuensoez	Documentation and web site enhancements.
01.11.003[?]	2010.05.31	Arno-Can Üstünsöz	Patch Default-Port VMware(TM)-Server-2.x, new tool ctys-beamer, add some documentation.
01.11.005[?]	2010.06.27	Arno-Can Üstünsöz	Alpha version of RDP plugin, bugfixes, added some documentation.
01.11.006[?]	2010.07.14	Arno-Can Üstünsöz	Alpha version of VBOX - VirtualBox(TM) plugin, bugfixes, added documentation, preparation of Typo3-Website.
01.11.008[?]	2010.07.30	Arno-Can Üstünsöz	Alpha-Version EnterpriseLinux, bugfixes, added documentation, First Gnome-Menues, ctys-scripts.
01.11.009[?]	2010.08.16	Arno-Can Üstünsöz	Alpha-Version gnome-starter, ctys-config, Fehlerbereinigungen, Ergänzung Dokumentation.
01.11.010[?]	2010.08.20	Arno-Can Üstünsöz	Verify GuetsOSs: ucLinux-QEMU(ARM+Coldfire), QNX-QEMU(x86), QNX-VBOX(x86),bugfixes, added documentation.
01.11.011[?]	2010.11.07	Arno-Can Üstünsöz	Verify New GuetsOSs: Android, MeeGo, RHEL, QNX. Version Updates: CentOS, Debian, OpenSUSE, OpenBSD, Ubuntu. Bugfixes, extension of documentation. menu generation.
01.11.014[?]	2010.11.22	Arno-Can Üstünsöz	Minor editorial.

1.2 Contact

Public maintenance:	acue@sfl_sourceforge.net
Administrative contact:	unifiedsessionsmanager@protonmail.com acue@UnifiedSessionsManager.eu
Commercial Services: The professional services are offered for end-customers only, so called 'body-leasers' are definitely not welcome.	Engineering Office Arno-Can Uestuensoez - www.i4p.com Ingenieurbuero Arno-Can Uestuensoez - www.i4p.com

1.3 Legal

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

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

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

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

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

QEMU is a trademark of Fabrice Bellard.

All mentioned RealVNC products and their registered names are Trademarks of the Company RealVNC Ltd.

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

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

Xen is a trademark of XenSource Inc.

If some is forgotten, it will be added immediately.

1.4 Acknowledgements

And, of course, I want to thank VMware for supporting their excellent VMware-Server and VMware-Player for free. The VMware-Workstation product initiated to my mind a major step of change and inspired a lot how software is commonly used and developed.

Many Thanks to Mr. Fabrice Bellard for his QEMU, which is the only and one test base for me to demonstrate a nested stack of VMs and it's integrated addressing including state propagation algorithms for now.

Great thank to the inventors of Xen at the university of Cambridge. UK, for their efficient VM.

And, of course, I would have probably no chance without "googling", so, even though it has to do something with business, many thanks for bringing the information of the whole world - and as soon as contacted of the remaining universe for sure - to my desktop. Hopefully I cope the current amount before the remaining universe comes into the scene.

I am meanwhile an enthusiastic user of CentOS/RHEL and OpenBSD, so I am glad having the opportunity to express my thank this way to all supporting persons an companies. Particularly RedHat Inc. for their actual open minded distribution policy and the CentOS team for their great work, and the OpenBSD team for their ongoing support for a base of real security.

And, last but not least, I want to thank very, very much to all the countless contributors for the numerous excellent Open- and Free-Software I use. Hopefully I can express my commitment and thanks with this piece of software, and my next following projects.

And finally I would like to express my thank to my friend Dirk and his wife Gisela, for their patience and enduring support. Their support at all enabled me reaching this milestone, despite of all the various and countless challenges and throwbacks to be managed.

Arno-Can Uestuensoez
Munich, Germany
March 2008

Chapter 2

Feature Specification

2.1 Feature-Sum-Up

The following tables present an overview of the supported components for current release.

The listed PC, Workstation and Server based platforms with listed Hypervisors are supported and tested when marked with "OK". Additional platforms are going to be added for next versions("*").

The utility "ctys-genmconf" supports the detection and generation of relevant control data, the utility "ctys-plugins" verifies actual available operational states and resulting features.

The main development and production platform for the UnifiedSessionsManager is CentOS.

The following pages show the current operational and test states of the various combinations of hypervisors, HostOS, and GuestOS. The actual operational states are visualized by specific colors as shown in next table.





Color	State
	The versions actually targeted to be supported with maximum available feature set.
	OK: Tested and operational in current release.
	NEXT: Scheduled for the next release.
	Probably already partly tested.
*	PLANNED: Intended for a later release.
-	OPEN: Technically possible, but for some reasons not yet planned to be implemented.

Table 2.1: Color coding of implementation and test states.

2.1.1 Supported Hypervisors

Supported Hypervisors on platforms as shown in the following tables.

Plugin	Supported Hypervisor	Versions		
		Previous	Current	InProcess
KVM	KVM		2.6.18/kvm-83 2.6.18-6/kvm-62 2.6.26-1/kvm-72	
OVZ	OpenVZ			
QEMU	Qemu	0.9.0	0.9.1,11.0.0,0.12.2	0.12.3
VBOX	VirtualBox(TM)	3.x	4.x	
VMW	VMware-Player(TM)	1.0.4	1.0.5,2.5.3,3.0.1	
VMW	VMware-Server (TM)	1.0.4,1.0.6,1.0.9	1.0.10,2.0.2	
VMW	VMware-Workstation(TM)	6.0.2,6.0.4,6.5.1	6.5.3,7.0.1	8.x
VMW	VMware-ESXi-Server(TM)			4.x.x
VMW	VMware-ESX-Server(TM)			4.1.0
XEN	Xen(TM)		3.0.3,3.1.0	3.3.0,3.3.1,3.4.2
XEN	Citrix-XenServer(TM)			5.5.0, 5.6.0

Table 2.2: Supported Hypervisors

2.1.2 Tested GuestOS support

The following table lists the already tested OS-Distribution vs. Containing Plugins. The containing plugins comprise the plugin itself as well as the required software and hypervisors.

Distribution	PMs		VMs						
	0	1+n	KVM	OVZ	VBOX	VMW	XEN	QEMU	
	x86	x86	x86	x86	x86	x86	x86	x86	ARM
BSD									
DragonFlyBSD-2.10.1	X	*	X	-	X	X	-	X	-
FreeBSD-7	OK	*	OK	-	*	OK	-	OK	-
FreeBSD-8	OK	*	OK	-	X	OK	-	OK	-
NetBSD-5.0.1	*	*	*	-	OK	*	*	*	*
NetBSD-5.2	X	*	X	-	X	X	*	X	*
OpenBSD-4 ¹	OK	OK	OK	-	OK	OK	-	OK	-
Linux									
CentOS-5	OK	OK	OK	*	OK	OK	OK	OK	-
CentOS-6	OK	*	*	*	OK	*	*	*	-
Debian-4-etch	OK	OK	-	-	-	OK	-	OK	-
Debian-5-lenny	OK	OK	OK	*	OK	OK	OK	OK	-
Debian-6-squeeze	*	*	*	*	*	*	*	*	*
Oracle Linux 6	*	*	*	*	*	*	-	*	-
Oracle Linux 5	*	*	OK	*	OK	*	-	OK	-
Fedora-8	OK	OK	OK	-	-	OK	OK	OK	-
Fedora-10	OK	OK	OK	-	-	-	-	OK	-
Fedora-12	-	-	-	-	OK	-	-	-	-
Fedora-13	*	*	OK	-	OK	*	-	OK	-
Fedora-14	*	*	*	-	*	*	-	*	-
Fermilinux-5.5	*	*	*	-	*	*	*	*	-
Knoppix6.2	*	*	OK	-	*	*	*	OK	-
Knoppix-6.2.1 ADRIANE	OK	OK	OK	-	OK	*	*	OK	-
Mandriva-2010	OK	OK	OK	-	OK	OK	*	OK	-
Scientific Linux 5	OK	OK	OK	-	OK	OK	*	OK	-
Scientific Linux 6	X	X	X	-	X	X	*	X	-
openSUSE-10.3	OK	OK	-	-	-	OK	-	OK	-
openSUSE-11.1							OK		
openSUSE-11.2	OK	OK	OK	-	OK	OK	*	OK	-
openSUSE-11.3	OK	OK	OK	-	OK	*	*	OK	-
openSUSE-11.4	*	*	*	-	*	*	*	*	-
Puppet-Linux	*	*	*	-	*	*	*	*	*
RedHat-Enterprise Linux 5	OK	OK	OK	-	OK	X	X	OK	-
RedHat-Enterprise Linux 6beta	OK	OK	OK	-	OK	X	X	OK	-
Slackware-13.1	*	*	*	-	*	*	*	*	*

Table 2.3: Getestete GuestOS

Distribution	PMs		VMs						
	0	1+n	KVM	OVZ	VBOX	VMW	XEN	QEMU	
	x86	x86	x86	x86	x86	x86	x86	x86	ARM
SuSE-9.3	-	OK	-	-	-	OK	-	-	-
SuSE-10.2	-	OK	-	-	-	-	OK	-	-
Ubuntu-6.06.1-dapper	-	OK	-	-	-	OK	-	-	-
Ubuntu-7.10-gutsy	-	OK	-	-	-	OK	-	-	-
Ubuntu-8.04-hardy	OK	OK	OK	-	-	OK	OK	OK	-
Ubuntu-9.10	OK	OK	OK	*	OK	OK	*	OK	-
Ubuntu-10.10	OK	OK	OK	*	OK	OK	X	OK	-
Ubuntu-11.04	*	*	*	*	*	*	X	*	-
Solaris(TM)									
Solaris-10 ²	OK	OK	OK	-	OK	OK	-	OK	-
OpenSolaris-2009.6 ⁴	-	OK	OK	-	OK	OK	*	OK	-
ILLUMOS ⁵	-	*	*	-	*	*	*	*	-
Nexenta ⁶	-	*	*	-	*	*	*	*	-
OpenIndiana ⁷	*	*	*	-	*	*	*	*	-
DOS									
FreeDOS ⁷	-	-	*	-	*	*	*	OK	-
Balder ⁷	-	-	*	-	*	*	*	OK	-
MS-Dos-5.x ⁷	-	-	*	-	*	*	*	*	-
MS-Dos-6.x ⁷	-	-	*	-	*	*	*	*	-
Windows									
MS-Windows-NT ⁷	*	*	*	-	*	OK	*	*	-
MS-Windows-2000 ⁷	*	*	*	-	*	OK	*	*	-
MS-Windows-XP ⁷	*	*	*	-	OK	OK	*	*	-
MS-Windows-2003 ⁷	*	*	*	-	*	OK	*	*	-
MS-Windows-7 ⁷	*	*	*	-	OK	*	*	*	-
MS-Windows-2008 ⁷	*	*	*	-	OK	*	*	*	-

Table 2.4: Tested GuestOS

Distribution	PMs		VMs						
	0	1+n	KVM	OVZ	VBOX	VMW	XEN	QEMU	
	x86	x86	x86	x86	x86	x86	x86	x86	ARM
Smartphone									
Android-2.2	*	*	OK	-	OK	-	-	OK	*
Windows-7-phone	*	*	*	-	*	*	-	*	*
Netbook									
Android-2.2	*	*	OK	-	OK	-	-	OK	*
MeeGo-1.0	*	*	(X)	-	OK	*	-	(X)	*
Windows-7-phone	*	*	*	-	*	*	-	*	*
Ubuntu-11.04									*
Tablet									
Android-2.2	*	*	OK	-	OK	-	-	OK	*
MeeGo-1.0	*	*	(X)	-	OK	*	-	(X)	*
Windows-7-phone	*	*	*	-	*	*	-	*	*
Ubuntu-11.04									*
Apple(TM) - OS									
Mac-OS-6.1.3	*	*	*	-	*	*	*	*	-

Table 2.5: Tested GuestOS

⁵No WoL for now.

⁶Some severe limitations may occur for Solaris, due the limitation of the "args" output of "ps" command to 80 characters. Thus the LIST action is faulty for some plugins, which means the instances are simply hidden due to argument-parts truncated by "ps". Some specific adaptations will follow. This depends on the argument ordering of the current command/wrapper and the actual contents being truncated. Supported Plugins: HOSTs and PM.

⁷Control by hypervisor only, no native support. Cygwin is foreseen for eventual future adaption. Tested with several versions, e.g. Windows-NT-Server, Windows-2000, and Windows-XP.

Distribution	PMs		VMs						
	0	1+n	KVM	OVZ	VBOX	VMW	XEN	QEMU	
	x86	x86	x86	x86	x86	x86	x86	x86	ARM
Embedded									
FreeRTOS	*	*	-	-	-	-	-	-	*
QNX	*	*	*	-	OK	-	-	(OK)	*
uCLinux	*	*	-	-	-	-	-	*	(OK)

Table 2.6: Tested GuestOS

2.1.3 Supported Native Plugins

The next table shows the passed tests of supported native plugins vs. OS-Distribution. The plugins including required hypervisors are to be executed on the listed OSs. Other OSs and versions might work as well.

Distribution	PMs	VMs						HOSTs			
	PM	KVM	OVZ	QEMU	VBOX	VMW	XEN	CLI	RDP	VNC	X11
BSD											
DragonFlyBSD-2.10.1	X		-	X	-	-	*	X	X	X	X
FreeBSD-7	OK		-	*	-	-	*	OK	*	OK	OK
FreeBSD-8	OK		-	X	-	-	X	OK	X	OK	OK
NetBSD-5.0.2	*		-	*	-	-	*	*	*	*	*
NetBSD-5.1	X		-	X	-	-	*	X	X	X	X
OpenBSD-4	OK	-	-	X	-	-	-	OK	X	OK	OK
Linux											
CentOS-5	OK	OK	*	OK	OK	OK	OK	OK	OK	OK	OK
CentOS-6	OK	*	*	*	OK	*	*	OK	OK	OK	OK
Debian-4-etch	OK	-	-	OK	-	OK	-	OK		OK	OK
Debian-5-lenny	OK	OK	*	OK	OK	OK	OK	OK	OK	OK	OK
Debian-6-squeeze	X	X	*	X	X	X	X	X	X	X	X
Enterprise Linux Server 5 / Unbreakable Linux	OK	OK	*	OK	X	-	OK	OK	(OK)	OK	OK
Fedora-8	OK	-	-	OK	-	-	OK	OK		OK	OK
Fedora-12	OK	*	-	*	*	*	*	OK		OK	OK
Fedora-13	*	*	-	*	*	*	*	OK	(OK)	OK	OK
Fedora-14	*	*	-	*	*	*	*	*	*	*	*
Fermilinux-5.5	*	*	-	*	*	*	*	*	*	*	*
Gentoo	*	-	-	-	-	-	-	*	*	*	*
Knoppix	OK	*	-	*	*	*	*	OK	(OK)	OK	(OK)
Mandriva-2010	OK	-	-	-	-	-	-	OK		OK	OK
openSUSE-10.3	OK	-	-	OK	-	OK	-	OK		OK	OK
openSUSE-11.1	OK	-	-	-	-	-	OK	OK		OK	OK
openSUSE-11.2	OK	OK	-	OK	*	*	*	OK		OK	OK
openSUSE-11.3	OK	OK	*	OK	OK	*	OK	OK	OK	OK	OK
OpenSUSE-11.4	*	*	-	*	*	*	*	*	*	*	*
RedHat-Enterprise Linux 5.5	OK	OK	*	OK	*	OK	OK	OK	OK	OK	OK
RedHat-Enterprise Linux 6.0 beta	OK	X	*	X	*	*	*	OK	OK	OK	OK
Scientific Linux 5	OK	OK	-	OK	*	OK	OK	OK	*	OK	OK
Scientific Linux 6	X	X	*	X	X	X	X	X	X	X	X
Sackware-13.1	*	*	-	*	*	*	*	*	*	*	*
SuSE-9.3	OK	-	-	-	-	OK	-	OK		OK	OK
SuSE-10.2	OK	-	-	-	-	-	-	OK		OK	OK

Table 2.7: Native Plugins vs. OS-Distribution

Distribution	PMs	VMs						HOSTs			
	PM	KVM	OVZ	QEMU	VBOX	VMW	XEN	CLI	RDP	VNC	X11
Ubuntu-6.06.1	-	-	-	-	-	-	-	OK		OK	OK
Ubuntu-7.10	-	-	-	-	-	-	-	OK		OK	OK
Ubuntu-8.04	OK	OK	-	(OK) ⁸	-	-	-	OK		OK	OK
Ubuntu-9.10	OK	*	-	*	*	*	*	OK	*	OK	OK
Ubuntu-10.10	OK	OK	*	OK	X	X	X	OK	OK	OK	OK
Ubuntu-11.04	X	X	*	X	X	X	X	X	X	X	X
Hypervisor-Distributions											
ESXi	*	-	-	-	-	-	-	*		*	-
ESX-4.1.0	X	-	-	-	-	X	-	OK	X	OK	OK
XenServer-5.5.0 ⁹	X	-	-	-	-	-	X	OK	X	OK	OK
Solaris(TM)											
Solaris-10	(OK)	-	-	-	-	-	-	(OK)		(OK)	(OK)
OpenSolaris 2009.6	OK	-	-	X	X	-	*	OK	X	OK	OK
ILLUMOS	*	-	-	*	*	-	*	*	*	*	*
Nexenta	*	-	-	*	*	-	*	*	*	*	*
OpenIndiana	*	-	-	*	*	-	*	*	*	*	*
MS-Windows(TM)/Cygwin											
MS-WNT4 ¹⁰	X	-	-	X	X	X	-	X	X	X	X
MS-W2K ¹¹	X	-	-	X	X	X	-	OK	OK	OK	OK
MS-WXP ¹²	X	-	-	X	X	X	-	OK	OK	OK	OK
MS-W2K3 ¹³	X	-	-	X	X	X	-	OK	X	X	OK
MS-W2K8R2 ¹⁴	X	-	-	X	X	X	-	OK	OK	OK	OK
MS-W7 ¹⁵	X	-	-	X	X	X	-	X	X	X	X
Apple(TM) - OS											
Mac-OSX-10.6.3	*	-	-	*	*	-	-	*	*	*	*
HP(TM) - OS											
HP-UX	*	-	-	*	-	-	-	*	*	*	*
IBM(TM) - OS											
AIX	*	-	-	*	-	-	-	*	*	*	*
Smartphone											
Android-2.2	*	-	-	-	-	-	-	*	*	*	*
MeeGo-1.0	OK	*	-	*	-	-	-	OK	*	*	*
Windows-Phone	*	-	-	-	-	-	-	*	*	*	*

Table 2.8: Native Plugins vs. OS-Distribution

¹⁵Compilation of 'qemu-system-x86_64' with support for '-name' option required.¹⁶Requires Cygwin support and a .

Distribution	PMs	VMs						HOSTs			
	PM	KVM	OVZ	QEMU	VBOX	VMW	XEN	CLI	RDP	VNC	X11
Embedded											
FreeRTOS	*	-	-	-	-	-	-	*	*	*	*
QNX	*	-	-	-	-	-	-	*	*	*	*
RTEMS-Dev	-	-	-	*	*	-	-	*	*	*	*
uCLinux	*	-	-	-	-	-	-	*	*	*	*

Table 2.9: Native Plugins vs. OS-Distribution

¹⁷Compilation of 'qemu-system-x86_64' with support for '-name' option required.

¹⁸Requires Cygwin support and a .

Unterstützte Produkte und Versionen für die jeweiligen Plugins. Diese variieren z.T. für die verschiedenen Plattformen.

Plugin / Toolset	Unterstütztes Produkt	Versionen		
		Vorversion	Aktuell	InBearbeitung
CLI	bash		3.2.39.1, >3.x	
	Cygwin(alpha)		1.7.9-1(2.738)	
RDP	rdesktop		1.6	
	krdc			3.5.10
	tsclient			0.150
	mstsc.exe			x
	vinagre			0.51
	VirtualBox		3.x	4.x
	MS-Windows(TM)		W4NT, W2K, WXP, W2K3, W2K8, W7	
	Cygwin(alpha)		1.7.9-1(2.738)	
VNC	RealVNC		3.3.7, 4.1.1, 4.1.3	
			4.1.1, 4.1.2	
	TigerVNC		1.0.90	
	TightVNC		1.2.9, 1.3.10, 2.0.2	
	MetaVNC			0.6.5
	QEMU/KVM		>0.9.x	
	UltraVNC			1.0.8.2
	VMware	W:6.x	W:7.x	
	XEN		>3.x	
	Cygwin(alpha)		1.7.9-1(2.738)	

Table 2.10: Durch HOSTs Plugins Unterstützte Clients und Desktops

Plugin / Toolset	Unterstütztes Produkt	Versionen		
		Vorversion	Aktuell	InBearbeitung
X11	gnome-terminal		2.22.3	
	xterm		235	
	emacs	>21.x	>22.x, 22.2.1	
	Cygwin(alpha)		1.7.9-1(2.738)	

Table 2.11: Durch HOSTs Plugins Unterstützte Clients und Desktops

Plugin / Toolset	Unterstütztes Produkt	Versionen		
		Vorversion	Aktuell	InBearbeitung
Desktop	Gnome		2.20.7	
	KDE		5.48	
	fvwm		2.5.26	
	xfce		x.x	
Shells	bash		3.2.39.1, >3.x	
	Cygwin(alpha)		1.7.9-1(2.738)	

Table 2.12: Unterstützte HOSTs-Plugin Sub-Komponenten

Plugin / Toolset	Unterstütztes Produkt	Versionen		
		Vorversion	Aktuell	InBearbeitung
QEMU	Qemu	0.9.0	0.9.1,0.11.0,0.12.2	0.12.3
	KQEMU			
	KVM			
VBOX	VirtualBox(TM)		3.1.2	3.2.8, 3.2.10, 4.x
VMW	VMware-Player(TM)	1.0.4	1.0.5,2.5.3,3.0.1	
	VMware-Server (TM)	1.0.4,1.0.6,1.0.9	1.0.10,2.0.2	
	VMware-Workstation(TM)	6.0.2,6.0.4,6.5.1	6.5.3,7.0.1	
XEN	Xen(TM)		3.0.3,3.1.0	3.3.0,3.3.1,3.4.2,4.0.0

Table 2.13: Unterstützte Server basierte VMs plugins

Plugin / Toolset	Unterstütztes Produkt	Versionen		
		Vorversion	Aktuell	InBearbeitung
VMW	VMware-ESX-Server(TM)		4.1.0	
	VMware-ESXi-Server(TM)			4.0.0
XEN	Citrix-XenServer(TM)		5.5.0	5.6.0

Table 2.14: Unterstützte Host basierte VMs plugins

2.1.4 Tested Client OSs

The following table lists the already tested client OSs.

Distribution	ctys			GUI						
	GROUP	DF	CF	X11		WM				
				X11	Xinerama	Gnome	KDE	fvwm	xfce	ffs.
BSD										
DragonFlyBSD-2.10.1	X	X	X	X	*	X	X	X	-	
FreeBSD-7										
FreeBSD-8	X	X	X	X	*	X	X	X	-	
NetBSD-5.2	X	X	X	X	*	X	X	X	-	
OpenBSD-4	X	X	X	X	*	X	X	X	-	
Linux										
CentOS-5	OK	OK	OK	OK	OK	OK	X	X	X	
CentOS-6	X	X	X	X	X	X	X	X	X	
Debian-5-lenny	OK	OK	OK	OK	OK	OK	X	X	X	
Debian-6-squeeze	X	X	X	X	X	X	X	X	X	
Enterprise-Linux Server	*	*	*	*	*	*	*	*	*	
Fedora-8										
Fedora-10	X	X	X	X	*	X	X	X	X	
Fedora-12	*	OK	*	OK	*	OK	*	-	*	
Fedora-13	*	*	OK	OK	*	OK	*	-	*	
Knoppix	X	X	OK	OK	*	OK	X	X	X	
Mandriva-2010	*	OK	*	OK	*	OK	*	*	*	
Scientific Linux	OK	OK	OK	OK	*	OK	OK	-	-	
openSUSE-11.2	OK	OK	OK	OK	*	OK	OK	OK	OK	
openSUSE-11.3	*	*	*	*	*	*	*	*	*	
RedHat-Linux Server 5.5	*	OK	*	*	*	OK	*	*	*	
RedHat-Linux Server 6.0 beta	*	*	*	*	*	*	*	*	*	
Ubuntu-6.06.1-dapper										
Ubuntu-7.10-gutsy										
Ubuntu-8.04-hardy	OK	OK	(OK)	OK		OK	OK	OK	OK	
Ubuntu-9.10	X	X	X	X	X	X	X	X	X	
Ubuntu-10.10	X	X	X	X	X	X	X	X	X	
Hypervisor-Distributions										
ESXi										
ESX	X	X	X	X	*	X	X	X	X	
XenServer-5.5.0	X	OK	X	OK	*	OK	X	X	OK	

Table 2.15: Getestete ClientOS

Distribution	ctys			GUI						
	GROUP	DF	CF	X11		WM				
				X11	Xinerama	Gnome	KDE	fvwm	xfce	ffs.
MS-Windows(TM)										
Windows7(TM)										

Table 2.16: Getestete ClientOS

Distribution	ctys			GUI						
	GROUP	DF	CF	X11		WM				
				X11	Xinerama	Gnome	KDE	fvwm	xfce	ffs.
Apple(TM)-OS										
Mac-OS	*	*	*	*	*					
Solaris(TM)										
Solaris-10	*	*	*	*	*	*	*	*	*	*
OpenSolaris-2009.6	X	X	X	X	*	X	X	X	X	
ILLUMOS	*	*	*	*	*	*	*	*	*	
Nexenta	*	*	*	*	*	*	*	*	*	
OpenIndiana	*	*	*	*	*	*	*	*	*	
Windows										
MS-Windows-NT										
MS-Windows-2000	*	*	*	*	*	*	*	*	*	
MS-Windows-XP	*	*	*	*	*	*	*	*	*	
MS-Windows-200x	*	*	*	*	*	*	*	*	*	
Smartphones										
Android	*	*	*							*
MeeGo	*	*	*							*
Embedded										
QNX	*	*	*	-	-	-	-	-	-	*
uCLinux	*	*	*	-	-	-	-	-	-	*
FreeRTOS	*	*	*	-	-	-	-	-	-	*

Table 2.17: Getestete ClientOS

¹⁷Kein WoL.¹⁸Einige Einschränkungen bei LIST.¹⁹Unter ausschließlicher Kontrolle des hypervisors. Getested mit diversen Versionen, z.B. Windows-NT-Server, Windows-2000, und Windows-XP.

Part II

User Interface

Chapter 3

Common Syntax and Semantics

3.1 General CLI processing

The common structure of the CLI call interface is defined by following basic elements.

```
<command> \  
  <local-options> \  
  [--] \  
  <common-remote-options> <argument-list>  
  
<argument-list>:=  
  <argument>['(<context-options>')'] [<argument-list>]  
<argument>:=<command>  
<context-options>:=<local-options>
```

The options are grouped and assembled by suboptions, which are scanned and operated by the involved plugins only.

Arguments <execution-targets> can contain their own scope of options and suboptions. These are pre-analysed on the caller site, but take mainly final effect on the execution site only.

Within the implementation of ctys the actual application of options within each scan is order dependant. The options are scanned from left-to-right, and in case of competition the last will win.

This changes, when subjobs are generated. Each job is resolved with the global remote options and it's own options, finally superposed with the actual set pre-environment - including from the previous jobs. Due to group resolution and the accessibility of several desktops, some reordering and grouping of tasks can appear.

Pre-required options are prefetched for bootstrap phase itself until the CLI processing parts are active, this is e.g. the case for options related to dynamic and on-demand load of bash-libraries.

All keywords in parameters are converted and treated internally as uppercase. Though 'all' is equivalent to 'ALL', 'AIL', and 'aLL'.

The additional extension as described in the following chapters are "group" instances for a set of hosts,

```
<argument>=(<host>|<group>)
```

```
<group>=<host>{1,n}
```

and macros, applicable as replacement-alias for any arbitrary CLI part and/or subpart, within any position except the <command> itself.

```

<command> <macro-alias>

  <macro-alias>=(
    [<local-options>]
    | [<common-remote-options>]
    | [<arguments>]
    | [<context-options>]
    | [--]
    | <any-resulting-sub-string-literal>
  ){1,n}

```

3.2 Options Scanners - Reserved Characters

The foreseen and implemented scanners are designed to allow implementation by simply nesting loops and using sets generated from basic regular expressions. This is particularly important for simplification of custom plugins. Following special characters are reserved for options definitions syntax:

- '=': Separator for option and it's suboptions. The reason for not using this as repetitive separator are "CALLOPTS" and "XOPTS", which are bypassed options for remote execution. These contain almost for sure a "=", but simplicity of the scanner is the priority here, so a second is chosen for repetition on groups.
- ',': Separator for suboptions belonging to same group.
- ':': Separator for suboption keys and it's arguments.
- '%': Separator for suboption argument values, will be replaced by space on final execution target "%=="'. Could be masked when required as literal by double-input "%%=="'.
- '()': Grouping character pair for target specific options belonging to a common target a.k.a. host.
- '{}' Grouping arguments for multiple targets including their specific options belonging to a common high-level-target a.k.a. SUBTASK .

3.3 Hosts, Groups, VMStacks and Sub-Tasks

3.3.1 Common Concepts

The UnifiedSessionsManager supports multiple execution-targets as combined group entity. A group entity is a logical unit with it's own execution context.

Group objects are mapped within ctys to one or more specific sub-processes, which are called SUBTASKS. SUBTASKS could be allocated implicitly and/or by request, and are distributed locally and/or remotely, and could be used arbitrarily intermixed with the various SUBTASK types and or just in-process host execution.

Curent version supports the following Sub-Tasks, which are described in detail within the following subchapters.

- SUBGROUP/SUBTASK
 - Sets up a collection of <execution-target> as a flat set of entities starting within the same call-context, but executed independently.
- VMSTACK
 - Sets up an execution context for members of a hierarchical stack, thus sequentially dependent on each other from-left-to-right.
- VCIRCUIT
 - Sets up an sequential relay chain with pre-assigned intermediate nodes for establishing an double-encrypted tunnel. The bypassing of Firewalls for specific access groups is typical application.

The subtask-entity could be used as a replacement for any position where an <execution-target> may be provided. A subtask could be customized with it's own context specific set of options, which will be dependent on the specific type - permuted to all it's members. The basic systax is structured as follows.

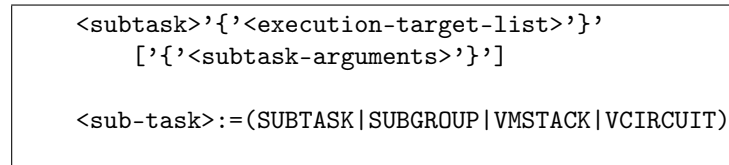


Figure 3.1: Subtask

3.3.2 Flat Execution-Groups by Include

The UnifiedSessionsManager supports for bulk access the concept of preconfigured groups. A group object, contains multiple instances of host objects and is a syntax element for replacement of an host entity, representing multiple nested instances. When providing one or more group entities, either intermixed with host entities or not, one main process(group) controls the whole set of subprocesses to be performed local or remote.

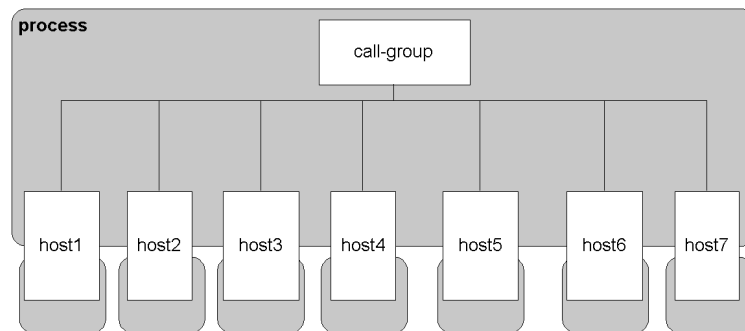


Figure 3.2: Groupresolution by Include only

The group object can replace any valid execution target and supports context options. Group objects has to be present on the callers machine, the tasks will be distributed to each member individually. E.g. the following construct could be used for a group:

```
...myGroup1'(-g :A20)' myHost1'(-d 99)' myGroup'(-W -g :A10)'...
```

Any level of nested includes is supported. circular inclusion will be detected at an default level of 20 and terminated than.

```
"#include <groupname>"
```

One example of groups expansion is given as follows:

```
ctys -a list MYGROUP01'(-d 99)' hostX MYGROUP02'(-d 3)'
```

```
group:MYGROUP01 "#include MYINCLUDE"
                 "host01,host02"
                 "host03"
```

```
group:MYGROUP02 "hostZ"
```

```
group:MYINCLUDE "hostA"
                 "hostB"
```

resulting call is - with MODIFIED ORDER:

```
ctys -a list
      hostA'(-d 99)' hostB'(-d 99)' \
      hostZ'(-d 99)' \
      hostX'(-d 99)' \
      host01'(-d 99)' host02'(-d 99)'host03'(-d 99)' \
```

For current version nesting of braces is NOT supported, but chaining of braces IS. Permutation is performed for now only for the first level of group resolution. Specific points of current scanner to be aware of is, that due to the following

1. re-ordering of entries
2. the "from-set-on" for all relevant, but not reset values of the overwritten context options

it has to be underlined, that when using context options within a group file, all items has to be set explicitly, or none at all. Other wise a number of side effects might occur due to unexpected mixture and interference of options from various contexts. This results technically from the decision to ease the design and implementation within the bash.

3.3.3 Structured Execution-Groups by Sub-Tasks

The group feature is extended within the "ctys" script by the concept of subgroups, which is slightly different from include.

The inclusion of a nested group is performed once at the beginning of a call, and is resolved in a "hungry" style, by complete resolution of the whole dependency tree. See figure:3.2 on page:39.

The resolution of subgroups is performed by a delayed name resolution, which is executed as a separate subprocess. The existence of an unresolved group within the defined dependency tree is checked immediately when matched for the existence of the non-included group definition file, this is done before starting the child process.

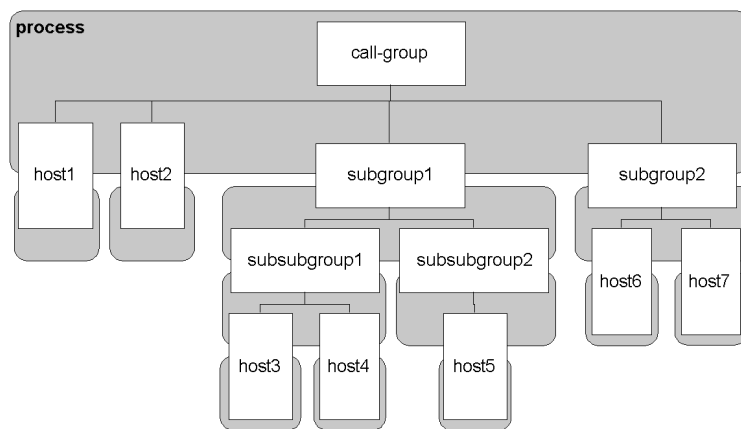


Figure 3.3: Groupresolution by Subgroups

The main advance of subgroups is the specific context of execution, where for example a completely different background operations mode could be established. One common example for this is the scanning of VM configuration files by "ctys-vdbgen" on machines with limited resources, where a sequential processing of multiple user accounts is required. This case could particularly be of relevance, when scanning various VMs on the same PM.

3.3.4 Stacks as Vertical-Subgroups

The VM-Stack implementation by it's design close to a Sub-Group with just a fixed set of context options.

The seamless extension of the design concept of SUBGROUPs to VMSTACKs, extends the "flat-feature" and "horizontal-feature" . These concepts could be applied intermixed, but the nesting of VMSTACKs is not provided.

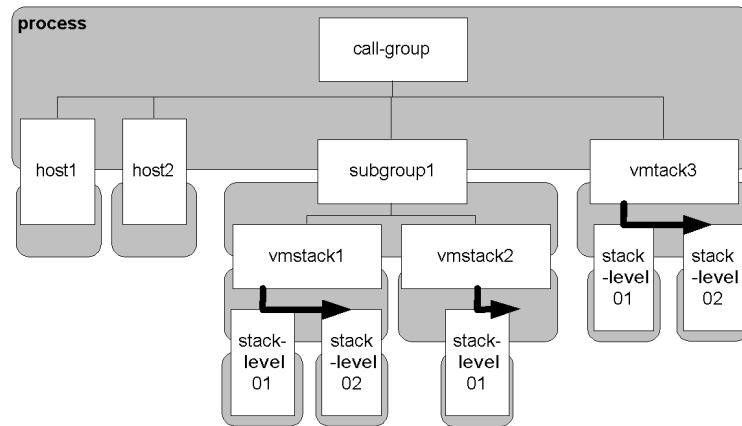


Figure 3.4: Combined Subgroups and Substacks

Due to the inherent execution dependency by the "hierarchical vertical dependency" of the elements of a VM-Stack, the execution will be forced by the framework to set some generic attributes on framework level. This particularly controls the hierarchical execution dependency within the "vertical-feature" of a VM-Stack. In addition specific options for the activation of the stack-control are set.

- "-b SEQ,..."

The VMSTACK feature(at least for now) requires the sequential execution of the parts of the requested stack.

The current version supports a "single-line" of a stack for a single call, thus "upper-trees" and "branches" of entities will be rejected. When multiple instances on a specific level are required, these have to be executed within multiple STACK-REQUESTS, which of course might share various parts of their lower branches. The STACK-REQUESTS could be combined to one "bulk-execution" call, and might not interfere erroneously, when all have the "REUSE" flag set. Anyhow, due to partly unavoidable polling, some repetition-counters and timeout values might be set appropriately, as they are for the most of the cases by the default values.

- "-b SYNC,..."

The sequential execution implies the synchronous execution, because no parallel threads within a single VM-Stack call are supported.

- "-b STACK[:<max-stack-height>],..."

The key STACK forces the previous listed keys to be set as described and rejects any further changes. This key implies and forces to the processing of the whole set of following <execution-targets> as a member of one VM-Stack. Due to possible unintended calls with groups expanding to a mass of targets, a configurable threshold value for the maximum of expected stack members is set by default CTYS_STACKHEIGHT_DEFAULT. This could be modified persistently and/or set call-by-call.

- -a CREATE= STACKCHECK :....

The VMSTACK will be pre-checked concerning various aspects once executed. These checks could be too restrictive for daily business and might not really be required, therefore some should be deactivated when appropriate.

One specific candidate is the CONTEXT property, which represents the location context where the configuration file was originally detected by ENUMERATE. This defines by default the "ContainingMachine" as a pre-requisite for the locality of execution. The configuration attribute EXECLOCATION controls this property, which is actually the PM/HOST attribute. The default value is set appropriately for the several session types, and defines independently from the actual existence of additional requirements whether the machine is fixed to be executed on a specific location. A common reason could be caused e.g. by security, where a critical machine containing data and access keys for financial departments has to be fixed to a specific location only. The value ROADWARRIOR defines the VM to be executable anywhere, when additional pre-requisites are fulfilled. Other VMs, might be more restrictive due to their lack of support for stacking on other entities. The value LOCAL restricts the

execution to the original scan location. Particularly emulator based VMs as QEMU, which in general could be executed anywhere, are set to the default ROADWARRIOR. This eases the initial creation of a cacheDB and requires a smaller amount only, due to the inherent flexibility of the initial execution location.

The complementary attribute RELOCCAP defines the change of a location for an active machine, as provided by means of the utilized hypervisor. The STACKCHECK could be disabled partially or completely, what is foreseen for test cases primarily.

The Figure:3.4 depicts by the symbolic arrows a probable execution sequence as a dependency caused by nested containment. Thus the "stack-level 02" are remotely executed within the instances "stack-level 01", as would be the "stack-level 03" within the "stack-level 02".

The previously mentioned basic checks for a VMSTACK include the consistency of the following characteristics of the stack.

- collectStackData
Collects the data required for further analysis, thus performs the very first check for the availability of the required data.
- verifyCreateOnly
This is a specific test for this version, where the combination of CREATE actions for VMs/PMs is supported only.
- verifyStacking
This checks the consistency in addresses of the actual call commands.
- verifyStackCapability
This verifies the session type of the VM against the STACKCAP attribute, thus the availability of the appropriate hypervisor.

Anyhow, due to the option of dynamically start different customized kernels for various modern OSs, the STACKCAP, which in case of a cached entity is a static snapshot only, might deviate from the last boot of the actual target. E.g. a VMware configured kernel instead of a Xen-ified kernel might currently be active, thus this check has some limitations concerning the synchronicity of its decision base.

- verifyHardwareCapabilityStatic
This verifies the compatibility of the hardware, as presented by the hypervisor to match the requirement of the GuestOS. This is particularly required for two properties, one is the architecture ARCH, which has to match the required CPU particularly for emulators such as qemu-ARM. The second is the virtual RAM, which might be exhausted by the single VMSTACK call and/or by the actually running additional VMs competing for the available resources.
- verifyStackLocation
This check verifies the location of the various stack entities. Therefore first the check of the bottom-level entity assures the location for the whole stack, whereas the additional checks verify the relative stack position of the upper layers, nested within the bottom element.

The check covers several aspects to be considered for wider stacked operations, where the embedded entities are not actually aware, and if, cannot really be sure, where they are actually executed. The first aspect to be covered is the availability of specific resources at a specific physical location - namely machine - only. The location has to be verified for example in order to have access to a specific local hardware-peripheral, which might be available at a small number of machines - PMs - only. Also a specific driver of a VM, which probably is available on specific site only, could be constraint. The second more generic, but possibly much more critical aspect is a possible security flaw, when an intruder becomes able to fake a location in order to hijack the whole, or just a part of a stack. This becomes quickly clear, when an accounting machine, implemented as VM, contains probably some specific data, or access keys. It has to be recognized, that the owner of the executing base machine is definitely the master of the nested upper part of the VMSTACK.

Thus at least a thoroughly performed pre-check for the actual locality before the execution has to be recommended.

The Figure:3.5 shows a 4-level stack example, which could be started with the following conceptual call example.

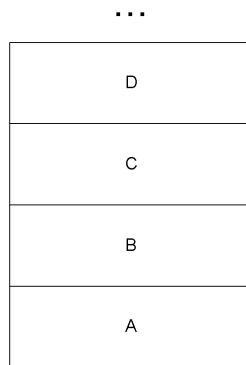


Figure 3.5: Stack Example for Basic Call-Interface

The inter-layer synchronicity of the required sequential execution of the stack entities implies some specific constraints for the eventually chosen CONSOLES. Thus the application of CONSOLE type of CLI has to be considered thoroughly due to its blocking character, which would block the whole upper stack, when applied. The application is still possible, but with the main intention of offering a means for application of the CMD feature. The non-blocking CONSOLE types will be silently forced into non-blocking and parallel operation by "-b async,par". The later independent creation of detachable CONSOLE types could be applied as usual. The usage of native HOSTs sessions is synchronous on session-level, conceptually seen as a non-layer stack-entity, which is embedded into a specific layer instead of being a layer entity by itself. Thus support for embedded execution of custom commands(CMD) is assured by sequential left-to-right operation of a VMSTACK.

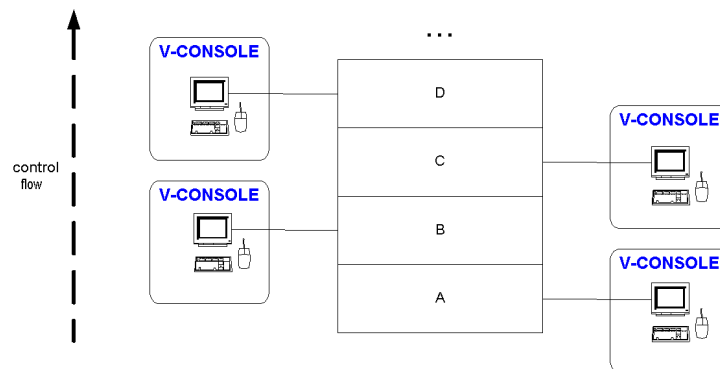


Figure 3.6: CONSOLE- and HOSTs-Asynchronicity for Stacked-Execution

The stack-synchronicity of the control flow for the operation, and though the application of the server components within sequentially dependent script-operations is assured by the "-b STACK" option, which is implicitly set. The attached CONSOLES will be just "popped-up" as chosen.

The following call incrementally startups the stack on the actual physical machine "A", after its activation by usage of the relay wolExecRelayServer(refer to Section ?? '??' on page ??).

```
ctys \
  VMSTACK' { \
    wolExecRelayServer(-t PM -a create=1:A,WOL ) \
    \
    A( -t SESSION-TYPE01 -a create=B ) \
    B( -t SESSION-TYPE02 -a create=C ) \
    C( -t SESSION-TYPE03 -a create=D ) \
  },
```

This is controlled by the detection of the keyword "VMSTACK", which starts a pre-configured SUBGROUP with specific forced pre-assignment of the "-b" option as described before.

The same call splitted to two calls, a first for the WoL call to start "A" by usage of the "wolExecRelayServer".

```
ctys -t PM -a create=1:A,WOL wolExecRelayServer
```

A second call for incremental startup of the stack on the actual physical machine "A".

```
ctys \
  VMSTACK' { \
    A( -t SESSION-TYPE01 -a create=B ) \
    B( -t SESSION-TYPE02 -a create=C ) \
    C( -t SESSION-TYPE03 -a create=D ) \
  }'
```

The current version just limits the allowed user suboptions for the "-b" option, but lets the remaining to the responsibility of the user. This offers the flexibility for example to use intermixed hypervisors, authentication facilities, and CONSOLE types within different levels of a stack call. But some parameters could only be "late-checked" for applicability just before the final execution. For this version no implicit creation of stack entities is supported, thus each CREATE has to be provided by the user, which could be combined to one call.

3.3.5 VCircuits as Sequentially-Chained-Subgroups

REMARK: This feature is currently under development, and thus is possibly partly or at all not yet available. If so, it will follow soon within an intermediate post-release. Same is true for the full range of description.

The VCIRCUIT subgroup is by its functionality close to the VMSTACK subgroup, which executes successively commands on a set of logically vertical grouped host entities. The VCIRCUIT utilizes a chained set of machines in order to establish a temporary static encrypted tunnel. The peer-to-peer tunnel is in addition to its SSH based sections encrypted as a virtual circuit, providing a higher level end-to-end channel.

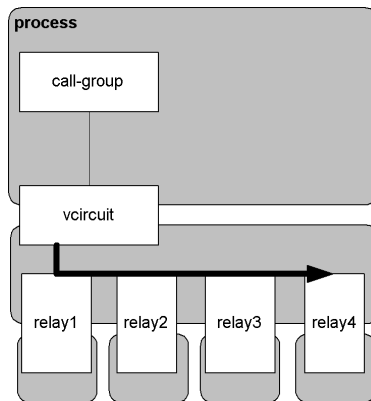


Figure 3.7: VCIRCUIT

3.4 CLI macros

The MACRO feature supports the usage of a predefined string alias as a literal replacement within any position of the CLI call.

A macro can contain any part of a call except the command itself. The whole set of required options including the execution target or only a subset of options could be stored within a macro.

The macro and its content are stored within a file which could be edited by each user or provided as a common defaults file. MACROS are resolved on each executing machine, thus even though a client could send a MACRO to the server, in current version the macro is resolved completely as the first step before the

resulting call is processed and distributed.

A macro is defined within the default file named "default" which is searched in the order:

1. "\$HOME/.ctys/macros/default"
2. "<actual-call-conf-path>/macros/default"

The <actual-call-conf-path> is evaluated from the resolved symbolic link of the call.

The following call syntax is provided:

```
MACRO:(
    <macro-def>
    |
    '{<macro-def>}'
)
```

```
<macro-def> :=
    <macro-name>
    [%<macro-file-db>]
    [%OPTIONAL]
    [% (
        ECHO
        | EVAL
    )
]
```

MACROS could be nested and chained as required. Even though the recursion depth could be arbitrary a counter is implemented, which sets a threshold limiting recursive processing. This is set by the configuration variable CTYS_MAXRECURSE. The variable protects all recursion depths, thus should be handled carefully. Default is 15 levels.

When macros are closely embedded into strings braces could be used, this could e.g. be applied in order to append context options to predefined macros.

```
ctys '{macro:tst-subgroups-01}(-d 99999)'
```

Where the macro "tst-subgroups-01" is defined as:

```
tst-subgroups-01 = -a list SUBGROUP'{host1 host2}'
```

This expands at the end to:

```
ctys -a list host1'(-d 99999)' host2'(-d 99999)'
```

The keyword "MACRO" prefixes the actual macro alias with the following parts.

<macro-name>
The actual name of the alias to be replaced.

<macro-file-db>
The default macro file could be altered by this new file name. The "macros" directories will be scanned for a file with given name.

OPTIONAL

The given macro signed as optional, thus if it is not found it will be ignored silently. Else a missing macro leads to an error and abort.

ECHO

The given macro is inserted by "echo" command into the replacement position, which is the default behaviour.

EVAL

The macro is evaluated on the callers site by "eval" call and the result is inserted into the insertion position.

The MACRO feature could be combined with the GROUP feature in various ways, particularly the combination with the raw syntax of the supported SUBTASKs for SUBGROUP and VMSTACK is applicable.

The following example shows the call of a predefined SUBGROUP with activated remote debugging for the "permuted" targets resulting from the MACRO.

```
ctys '{macro:tst-02%test-subgroups}(-d 99999)'
```

The named MACRO-file-db "test-subgroups" contains here the test-case "tst-02" for a "LIST" action on two remote test-hosts, which is:

```
tst-02      = (-a list) SUBGROUP'{host01 host02}'
```

Thus the resulting actual call executed by ctys after MACRO and GROUP resolution is:

```
ctys '(-a list)' host01'(-d 99999)' host02'(-d 99999)'
```

This call suppresses for now the display of a header, just executes on the remote hosts and displays the actual data-rows. This is due to the missing assignment of a local ACTION, which is required as an overall controller for actions displaying data in competition for the display.

The following call displays the LIST table including a local header.

```
ctys -a list host01'(-d 99999)' host02'(-d 99999)'
```

The following call in addition first collects data, thus does not poison the result data within the table with eventual ERROR messages and WARNINGS, but but them before the table.

```
ctys -a list -C raw host01'(-d 99999)' host02'(-d 99999)'
```

For additional variations refer to the available generic options.

3.5 Common Options

<callopts>

Call options are passed literally to a remote command, therefore no intermediate processing is performed. White spaces are not supported and has to be replaced by '%'. E.g. "bash -e ls" is masked as "bash%-e%ls".

(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.

CMD:<cmd> Replaces the standard definition of a command execution shell by "CLI_SHELL_CMD_DEFAULT='bash -c'" for remote execution.

<xopts>

X-options are passed similar to <callopts>, but to a X11 application. The user has to be aware of the single-hyphen and double-hyphen usage of the various X11, tools for their options. The core parts for geometry and title are set by ctys.

Be aware, that some of X-options such as "-geometry" and "-name" are already implicitly utilized by other options, thus use this if, than CAREFULLY.

(SHELL|S):<shell>

Replaces the standard definition of an interactive shell by "CLI_SHELL_DEFAULT='bash -i'" for remote execution.

When setting an own shell the masking of SPACeS has to be applied by "%" in accordance to common ctys cli rules. The default will be applied on command line interface as:

```
"... -a CREATE=c:bash%i ..."
```

The main difference to the CMD option is the execution of the given command without starting a new shell previously. Therefore available library functions of ctys could be called. For examples refer to Section ?? '??' on page ?? .

Chapter 4

Core Data

4.1 Overview

The internal static configuration data is based on the output records of the ENUMERATE action which has a key role in generating the internal caching database to handle VMs and PMs and addressing the offline GuestOSs.

The ENUMERATE action scans local and remote filesystems and detects the configuration files of each active plugin. This is technically performed by calling an internal interface of each actual loaded plugin in the operational state ENABLED. This is performed on each enumerated execution-target and collected into a common database on the calling machine.

The data is stored in the MACHINE format, which is a semicolon separated ASC-II record format, and could be imported to almost any database and spreadsheet. The description of the records could be displayed by usage of the common keyword TITLEIDX or TITLEIDXASC within each reporting action and tool.

The data scanned by ENUMERATE is pre-cached into a local database and managed by the tool "ctys-vhost" due to the processing time required for a filesystem scan. A similar reason is the included management of entities, which are potentially off-line when the query for specific attributes is performed. Thus caching supports required functionality for off-line PMs and VMs as well, as a reduction of the average query-time to less than a second. In contrast to this, the actual scan of a deeply structure filesystem for a configuration file could vast minutes resulting from a simple attribute value assertion.

The second main application is the scan for actually operating entities by LIST, which is based on the internal representation of the common interface LIST, used for various queries and dynamic ID conversions. For example most of the LABELs are converted by usage of the LIST action. This could be cached too, but due to it's realtime or at least near-time requirement, onyl temporary short-time caches are utilized. Refer to "-C" ans "-b" options.

Even though the data is stored in the standard record format, some minor variations have to be applied to the various kinds of processing actions. For now basically three variations are distinguished:

ENUMERATE=<field-name><processing-options>

The collected distributed static and raw data from the configuration files as provided by the user. Some minor add-ons, such as DNS and MAC resolution, are provided optionally.

The main applications are the internal usage for dynamic path-extension of addressed targets in actions by usage of the UNIX "find" command, and secondary the pre-fetch of this information into a static cache database.

LIST=<field-name><processing-options>

The dynamic data of all actual running plugins, this comprises not only the VMs and PMs, but also the HOSTs and TUNNELs.

The LIST function is the working-horse for displaying and managing the actual dynamic state of all involved physical and virtual machines, including the contained operational facilities.

`ctys-vhost -o <field-name><processing-options>`

The crucial interface to cached offline data for interactive user queries and the internal first-priority access base for configuration data queries. Refer also to `ctys-vdbgen`.

4.2 Standard Configuration Files

The plugins provided with the UnifiedSessionsManager could be generally subdivided into two categories by the way runtime data is handled.

- transient runtime data

These plugins handle dynamic data only, which is valid during their lifetime only. These are particularly all HOSTs plugins, such as CLI and X11, where particularly temporary system IDs with temporary LABELs are utilized as aliases.

This category of data is available for LIST action only, and cannot be enumerated. Anyhow, the dynamic instances of the persisten category are included in LIST action too.

- persisten runtime data

These are mainly VMs, but PMs also, where the majority of required system data is defined within persistently stored configuration files and within the required runtime `bld/build/output/doc-tmp/en/images`.

This category of data as stored instance attributes is available by usage of ENUMERATE and could be therefore prefetched and cached. The entries have to be defined and maintained by the user as supported for the different plugins.

Due to the integration of various hypervisors with different originators, the configuration data differs naturally more than having equal parts. The integration into one more or less seamless and at least basically unified interface is one of the main goals of the UnifiedSessionsManager. The limiting edge of forcing compatibility is reached, when the processing of the configuration data for the various hypervisors has to be handled. Therefore the following file-extensions with additional `ctys`-fields and records are supported. These are the file-extensions, which the plugins specific ENUMERATE actions are aware off, and though could be processed by `ctys`. Other file extensions will be ignored, and therefore are not accesible.

Plugin	File-Extensions
PM	conf
QEMU/KVM ¹	ctys
VBOX	ctys
VMW	vmx,ctys
XEN ²	conf,ctys

Table 4.1: Supported File-Extensions

Basically two types of files could be distinguished, the configuration files with pure configuration data (`conf`, `vmx`), and mixed files(`ctys`), containing configuration data and/or executable script code, which is defined to be bash-code.

Particularly for QEMU, due to the original command line interface only, some wrappers are applied for various reasons. Therefore, beneath the VDE/VirtualSquare wrapper for management of the network interfaces, the `ctys`-wrapper is introduced in order to handle the flexibility of the call interface and the amount of call options offered by QEMU.

¹For QEMU some extended control of `BOOTMODE` could be applied in combination with PXE boot.

²Some specific variations to name-conventions are applied on XEN in current version.

4.3 Common Data Fields

This section represents the core set of data which is used in several actions. Some of its members are varied within specific call contexts when applied, and are therefore in addition specialized within the following sections.

ACCELERATOR

The accelerator available for ENUMERATE or actually executed for LIST. The available types of accelerator depends on the session type. These are for example:

QEMU : KQEMU, KVM, QEMU

VBOX : PARA, HVM

VMW : PARA, HVM

XEN : PARA, HVM

ARCH

Virtual architecture presented to the GuetsOS and hypervisors of upper-stack-layer.

BASEPATH|BASE|B

The path-prefix for the search root by UNIX command "find" to fetch all present configuration files within the subtree. This could be a list of nodes to be scanned, as depicted in the definition of <machine-address> .

This attribute identifies search groups of VMs as stored and organized within parts of the filesystem. Thus defines scopes of entities to be visible only for additional selection criteria.

One typical application is to define a set of VMs within a directory with access permissions for a specific group of users only. Therefore caching and caching of data from that subtree is required for the permitted group only. Due to the supported input parameter BASEPATH of CREATE action, this could be used for views of work-scope organization, as well as basis for load-balancing and various task dispatching groups. Several views could be organized by usage of symbolic links.

This parameter is applicable to VMs and PMs only, not to HOSTs.

CATEGORY|CAT

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

CONTEXTSTRING|CSTRG

A private context storage for the plugin.

CTYSRELEASE

The so called MAGICID describing the current release of the UnifiedSessionsManager which created this record. Therefore each record could be traced to its originator for debugging and compatibility reasons. This is somewhat handy, due to the actually distributed creation of the semantics at least, which is performed as a standalone task on each executing target by usage of the local hypervisor and ctys.

This is foreseen to handle varying data sources of course.

DIST

The distribution installed within VMs guest or PMs. This parameter is applicable to VMs and PMs only, not to HOSTs.

DISTREL

The release of the distribution.

EXECLOCATION

Defines the possible execution locations by a customizable list of possible execution locations. Thus various distribution policies could be implemented including specific views of an upper-layer semi or

fully-automated algorithm.

The availability of the appropriate hypervisor has to be considered by the editor, else a missing type will be detected by an error when execution starts. This could change also dynamically, e.g. when during boot time different kernels providing different hypervisors are chosen.

Due to distribution algorithms which rely on this set when configured and activated by the RELOC-CAP option, the value of this parameter should be maintained thoroughly.

The following key values are supported for EXECLOCATION:

- LOCAL
Could be executed at the install location.
- ROADWARRIOR
A VM which could be started at arbitrary location. Anyhow, the availability of the specific hypervisor is still required.

EXEPATH

The actual path of the executable service access entity.

GATEWAY

The internet Gateway.

HWCAP

The offered virtual HW capacity by the VM. This is particularly foreseen to setup specific devices, which are physically colocated to a specific PM and are accessible local only. Examples might be specific HW-Test-Devices, as well as Machines. Another example are DVD-Recorder, Tape-Drives, specific security devices, or the required DMZ, in order to limit risks by opening connection with piercing of firewalls.

HWREQ

This parameter is similar to the HWCAP parameter, but describes the required HW.

HYPERREL|HYREL HYPERREL|HYREL

Release of the hypervisor used for installing the VM.

HYPERRELRUN|HRELRUN|HRELX|HRX

The release of the current hypervisor. This is in case of ENUMERATE the locally available and foreseen hypervisor, whereas in case of LIST this is the actually executed hypervisor, which in case of some plugings could be varied by call parameters.

ID|I

The ID of a plugin type. The syntactical data type varies for the miscellaneous plugins. For configuration-file based plugins, this is the filepath of a valid configuration file, unique on the executed machine. The plugin types of PMs and VMs generally support a configuration file.

For dynamic plugins with temporary and volatile IDs, like CLI, X11, and VNC, the identifier represents an arbitrary numerical identifier, which is returned by the hosting system and/or the executed software component.

Following current exceptions and specifics apply:

XEN

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

The volatile domain-ID is handled - due to hypervisor architecture and structural and dynamic means of accessibility - similar to an ordinary "UNIX-pid".

HOSTs

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, QEMU and WMWare-Workstation 6.

For the CLI plugin the "initial-call-pid" of the topmost UNIX process is used as ID.

So, it is just an abstract ID, no generic overall-algorithm for it's calculation is available. The only requirement is uniqueness within the required execution scope, which additionally could be deactivated by the "-A" option.

IFNAME

The name of the interface within the GuestOS, which is correlated to this data record.

This may vary due to several reasons, thus the synchronity is within the responsibility of the user.

JOBID

The internal Job-ID assigned by ctys. The Job-ID is relevant for CREATE action only, though the remaining are just temporarily active, could be called "transient" actions. The CREATE action itself is transient too, but it's entity might be a "longer running" item, thus will be called "qualified as persistent" here, emphasizing it's existence after the final return of the CREATE request.

Currently not all types of plugins assign persistent JOBIDs, thus could be listed only during initial execution. One example is the PM plugin when used with WoL.

The data required for display of JOBID is stored within a temporary file related to the PID of the item as displayed by LIST action. This data could be stored in shared mode, which enables anyone to display the full size of the LIST records, or it could be stored as private data, which just grant access permissions to the owner. This is controlled by the variable MYTMPshared within the configuration file "ctys.conf" and/or by pre-setting environment variable "MYTMPshared=NONE".

The SHARED usage might not be security relevant, due to usage of the private data first with priority, whereas "not-own" entities from shared directory are controlled by systems security facilities.

LABEL|L

LABEL is a user defined alias as a user-friendly replacement for the ID. The various plugins set different requirements for the LABEL. So the Xen plugin requires a mandatory domain name for a DomU, which is used as LABEL. The VMware plugin utilizes the optional "displayName" as LABEL. For the QEMU an own configuration file format is defined. The PM plugin uses the DNS name as LABEL.

For the HOSTs plugins the label is a call parameter, which is associated to the callee, temporarily valid for the current session only.

Any contained ":"-colon will be replaced by an "_"-underscore.

MAC|M

The preconfigured Ethernet MAC address. In current release only static configured MAC addresses are supported. Particularly any mapping information of associating Ethernet addresses with TCP/IP addresses has to be statically assigned, and could be generated from DHCP configuration files and/or ping+ARP caches. Address pools of DHCP are not supported.

Only applicable to VMs and PMs.

REMARK: In case of multiple interfaces for an instance, each interface is enumerated as a separate entity and eventually stored in the cache database.

NETMASK

Internet Netmask.

NETNAME

Name of the interface as to be used for external access. This is frequently the DNS name, which is recommended to be configured within DHCP. This should be done, even though the interface is probably used "addressless", e.g. for sniffing purposes. The inclusion within the DHCP database provides automatic conversion into cacheDB and thus enables the usage within the UnifiedSessionManager. This is required for utilising WoL, where the actual interface on the host might be addressless (refer to Section ?? '??' on page ??).

OS|O

The OS running within the queried PM or VM. Not applicable to HOSTs category.

OSREL

The release of the OS.

PM|HOST

The TCP/IP address of the hosting machine, which is derived from the "uname" output. This field exists for all local interfaces and has to be distinguished from the NETNAME.

This could be a PM, which is the founding physical machine, running the whole VM stack contained, or in case of an contained entity within the VM stack it is a VM itself, executing virtually as a PM for a nested upper stack VM itself.

PNAME|P

Almost the same as <ID|I>. This is due to the usage of filename of the configuration as a unique ID at least within the namespace of a single hosts filesystem.

RELAY

The interface, bridge, switch/hub, or the router, which interconnects the VM to the network. Could be a host-only and/or an external connection.

RELOCCAP

The capabilities offered by the VM for relocation of its execution base. The current version supports the following values only, which could be applied in combination:

- **FIXED**
The VM could not be relocated at all. It is executable at the install location only.
- **PINNED**
The VM could not be relocated once it is started. A common reason could be the attachment to a specific hardware device, which possibly might be even available locally only.

An example may be a debugging device for an embedded system, which is accessible by LPT device on local PM only.

- **ROADWARRIOR**
A VM which could be allocated and reallocated arbitrarily, as though not specific requirements to the execution base is given. Anyhow, the availability of the specific hypervisor is still required.
- **<EXECLOCATION>**
The VM could be executed on a member given by the EXECLOCATION parameter only, which itself could contain GROUPs and MACROS. The execution string will be assembled by evaluating any extended-distribution criteria on the VM.

These include the load in the sense of internal load to a VM limited by additional facilities, not just the pure processing capacity of the physical CPU. In addition some restricted and/or limited resources have to be assigned, which could be specific devices. E.g. a special type of printer, plotter, or cutting machine, which has limited access due to serialization via a batch-queue. This requires the distribution of the VM to another stack located on a different physical machine, even though the actual load on the first targeted machine might suffice a simple average-CPU-load balancing criteria.

SERNO

An arbitrary serial number for the VM stored in the configuration file. This number should be unambiguous.

SPORT

Server access port for execution of an administrative TCP/IP connect, separated from the user access. This is the raw port to be used for server specific admin tools, which is different from user's client access.

For XEN this port is not supported due to security reasons.

For QEMU this port represents the monitoring port as a UNIX-Domain socket with specific naming convention.

SSHPORT

Alternative port for "-p" option of SSH, default when absent is given by the system as "22". Multiple ports in varying sets for each interface are supported by OpenSSH. For information refer to "*OpenSSH*"[?, OPENSSSH] and "*SSH The Secure Shell*"[?, SSHDefGuide].

STACKCAP|SCAP

The list of capabilities of the embedded support for the upper-peer-stack-level.

STACKREQ|SREQ

The list of capabilities of the required support from the founding bottom-peer-stack-level.

TCP|T

The ip addresses of the GuestOS interfaces running within the VM. Each interface could have multiple assigned IP addresses.

When enumerating the IP addresses, the MAC entries and the IP entries are scanned and correlated to each other based on the given numbers or the order only. The usage of the ordering position as index is applied specifically to Xen, due to lacking a numbering scheme for it's interfaces.

REMARK:

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

An interface without a MAC address is currently accepted, but generates a warning.

TYPE

Output of the type of session, either of category VM, PM, or a HOST by it's plugin name. The type of a session is to be used for the "-t" and "-T" options. In current version the following sessions are supported in the base set: CLI, X11, VNX, QEMU, VMW, XEN, PM

USERSTRING|USTRG

A string to be customized by the user, forseen as a reminder to be displayed only.

UUID|U

The UUID is not necessarily required, even though providing a quite well fitting globally unique identifier. The value could be generated e.g. by "uuidgen", but should be used as provided in case of PMs for hardware devices. For VMs and GuestOs it could be generated by the tool "ctys-genmconf".

The uuid is generally applicable for VMs and PMs only.

OSREL

The version number of the installed GuestOs distribution within the VM or PM.

The uuid is generally applicable for VMs and PMs only.

PLATFORM|PFORM

Virtual device, which is a unique identifier for the virtual hardware, either a PC-base, Server, or an embedded device.

VCPU

The pre-configured number of V-CPU's.

VERSION

The version of the VM config.

VMSTATE|VSTAT

The following values are applicable as actually stored attributes. The values could be used in query tools either by their literal values, or by choosing a processing-only meta attribute for the selection of a subset or superset.

The state value is semantically checked when generating a cacheDB by "ctys-vdbgen" and "ENUMERATE". The post-processing and analysis tool "ctys-vhost" for now just does a generic pattern match on the record-stream from the cacheDB. Ambiguity has to be avoided by the user.

Additional values could be defined by the user and will be added to the cacheDB. Due to some semantic checks in order to detect mistyped standard attributes, these are required to be deactivated by the `CUSTOM` key before the position of the arbitrary key. The `CUSTOM` key is valid for any following key not matching a pre-defined.

REMARK:

Attribute values have to be stored literally as uppercase, the later match by scanning via `ENUMERATE` is performed as uppercase only.

The following values are predefined standard states:

ACTIVE

The VM is actively participating in operations, thus ready to be used in a production environment.

BACKUP

The VM is a backup of an existing VM, not necessarily, but recommended of `ACTIVE` state.

DEACTIVATED

The VM is present but the configuration data could only be accessed partially. The common reason is the missing of access permissions, in some cases a registration of the VM to the native administration component is required. The concrete definition and required actions are defined by the specific plugins.

TEMPLATE

The VM is a template to be used as custom base for productive VMs. The VM itself could be operable, but does not require so.

TESTDUMMY

The VM is a installed configuration only for testing and validating the basic functionality of the VM.

The processing attribute `MATCHVSTAT` provides means and additional operations-attributes for selection of subsets.

VRAM

The pre-configured amount of RAM.

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`. This is required for example with `VMW` when using the workstation product of version 6.

4.4 Common Processing Options

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.

A typical addressing of an entity within a stack is:

```
<host>[<vm-level-1>][<vm-level-2>]<host-access>
```

For a concrete example this results to:

```
lab00.tstnet0.com[l:tstDomU01][t:tst3]'(
  -a create=l:myVNC01)'
```

or by an alternative address pattern:

```
lab00.tstnet0.com\
  [l:tstDomU01]\
  [p:/home1/qemu/tst3/tst3.ctys]'(-a
  create=l:myVNC01)'
```

The syntax describes a VM stack path to be used in order to execute an action within the topmost element.

Intermediate stack entries will be created by usage of default values when missing, default values are assumed for any missing option. Therefore the provided address information is accomplished from the required cached data by usage of `ctys-vhost`. The execution only performs, when the required data could be queried unambiguously from the available database. Else additional key information has to be provided by the caller.

The following example shows the usage of different authentication methods to the GuestOs.

```
lab00.tstnet0.com'(-Z NOKSU,SUDO -z pty,pty)'\
  [l:tstDomU01]'()'\
  [p:/home1/qemu/tst3/tst3.ctys]'(-a create=l:myVNC01)'
```

Even though the machine "tst3" could be directly addressed when in bridged mode and is accessible, namely already running, in case of CREATE and CANCEL the access to the hosting system is crucial. In case of CREATE, the extended addressing schema contains the physical location where the stack entity has to be executed, which could be extended by usage of wildcards for load-balancing or specific service distribution. In case of CANCEL the hypervisor could be involved into the shutdown, which is hidden else for access from within the GuestOS.

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.

IP

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

MACHINE

Complete records matching the `<regexpr-list>` in terse format for postprocessing. The output is a semicolon separated list of record, compatible with most spreadsheet applications. A title with the actual canonical field indexes could be displayed when combined with `TITLEIDX`. The extended variant `TITLEIDXASC` displays additionally the common column indexes for spreadsheet forms supporting the manual modification.

MATCHVSTAT

The `MATCHVSTAT` key supports selective operations on stored VM configuration records. Therefore in addition to the stored values of `VMSTATE` some operational attributes are defined, controlling sub and supersets.

The following values are applicable for attributes controlling the match process only and are therefore not stored literally.

ALL

This simply sets the value to be ignored and matches any present and valid entry. When applying to `ENUMERATE` on stored configuration files, the `MAGICID` still will be applied and might superposition the semantics of the `VMSTATE` attribute due it's higher severity.

CUSTOM

This key deactivates the validation of keywords, which allows arbitrary keywords to be used. This opens particularly the usage of custom `VMSTATEs` for definition of various scopes of sets to be included into the `cacheDB`.

EMPTY

This value simply sets the select to non-existing state.

PRESENT

This value simply sets the select to any but present state.

The default match value is: "ACTIVE|EMPTY".

MAXKEY

The maximum common set of attributes for `LIST` and `ENUMERATE`.

PKG

The list of packages, a.k.a. plugins, to be displayed, any other will be suppressed.

The syntax is as common: "PKG:pkg01%pkg02%...".

REC_GEN|REC:<tab-args>

Defines proprietary records for formatted output.

SORT[:<sort-args>]

Sorts the body of table with given scope on the column of defined `<sort-key>`. This is mainly a sort on the first column. It first collects therefore the whole data of each machine, before displaying the result almost at once. No progress indicator is shown. Due to the smaller sort-scope the partial delays might not be too long.

Additionally the option `"-C"` influences the scope of sort, where without activated caching the scope is each executing machine, leading to a concatenation of sorted sub-lists. When caching of the complete and raw result is chosen, the scope of sort is the whole result, displaying the list with sort applied to the complete set of records. Anyhow, when the first file is the PM/VM the result should be the same by default, until a specific sort-field is selected, which deviates from the default-field=0/1.

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

ALL

The sort is performed on top-level spanning the whole resulting table content.

EACH

The sort is performed on level spanning solely each of the executing machines. The result is therefore grouped by execution targets.

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>". The <sort-key> is the column index of the resulting output table as displayed, enumeration is an increment beginning with "1".

SPEC_GEN|SPEC:<tab-args>

Defines check records for formatted output.

TAB_GEN:<tab-args>

Defines tables for formatted output. A simple set of macros is defined for the setup of a table definition compatible with most spreadsheet applications.

TERSE

Lists the displayed items in machine processable way, it is the same format as with "-X" option.

TITLE

Optional "title" could be applied for header listing of field/column names.

TITLEIDX

Almost the same as "title", shows in addition the absolute and canonical field/column positions for addressing when a generic table is defined. The resulting output format is defined as the list of all actual selected field names, each displayed with its canonical index. For the output of all fields TITLEIDX has to be combined with MACHINE .

FIELD(index)

The current implementation for ENUMERATE yields to the default:

ContainingMachine(1);ID(4)

The complete record is displayed in combination with MACHINE as:

```
ContainingMachine(1);SessionType(2);Label(3);ID(4);
UUID(5);MAC(6);TCP(7);DISPLAY(8);ClientAccessPort(9);
VNCbasePort(10);Distro(11);OS(12);VersNo(13);
SerialNo(14);Category(15)
```

The output is compatible with various spreadsheet applications .

TITLEIDXASC

Almost the same as TITLEIDX, but with additional display of column indexes for various spreadsheet calculation programs.

USER:<user>%[<credentials>]

The user to be used for native access to the <action-target>. This is required frequently for ACTIONS supporting the "Peer" mode and the "Auto-Stack" mode . Default user is the same as used for authentication on the execution target. The default authentication method is determined by the login target.

The alternative user could be provided either without specific credentials for usage with a configured network based authentication, or with one of the supported types of credentials. Following keywords are case-insensitive.

USER:<user-name>[%<credentials>]

The following applies to the <credentials>, which could be of various types.

credentials:=<credential-type>%<credential>

Current version supports pre-configured network authentication only, either interactive, or by a specific protocol like GSSAPI/TLS, thus the <credentials> field is not yet supported.

XML_GEN|XML:<tab-args>

Defines XML records for postprocessing output.

4.5 Specific Variations

BASEPATH|BASE|B:<output>

(BASEPATH|BASE) [: [<top-level>] [%<bottom-level>]] {1,n}

The path-prefix for the configuration file of the current VM. This identifies search groups of VMs as stored and organized within parts of the filesystem.

Due to the supported input parameter BASEPATH of CREATE action, this could be used for views of work-scope organization, as well as basis for load-balancing and various task dispatching groups. Several views could be organized by usage of symbolic links.

GROUP

The actual group id of remote server process.

USER

The actual user id of remote server process.

PID

The pid of remote server process.

TUNNEL|SERVER|CLIENT|BOTH

List all selected types of connections on selected host. Default is S, which is similar to the definition of a session.

The following sets could be selected:

TUNNEL: tunnels only

CLIENT: clients only

SERVER: servers only

BOTH: Yes, eh, all three.

4.6 Generic Tables

Several actions, particularly the GENERIC class of calls INFO, LIST, ENUMERATE, and SHOW support data to be displayed in multiple specific views. The same applies for some support tools, particularly "ctys-vhost". The views may vary from task to task and should emphasize different topics.

Therefore the output could be adapted by the user with generic tables, which support a simple syntax with required minor knowledge only. These custom calls, which are based on a suboption for the specific action, could be stored as a MACRO and reused later. The recursive MACRO resolution supports for modularized table definitions which could be reused within the same ACTION, but due to canonical standard parts of some ACTIONS as LIST and ENUMERATE, also partly within multiple ACTIONS. An example could be found in Section ?? '??' on page ?? .

REMARK: Currently the actions LIST , ENUMERATE and ctys-vhost , support generic tables only, others will follow within next versions.

The common syntax for definition of a generic table is the following snippet of syntax, which has to be a supported suboption of the called ACTION.

```
-a <action>=(
    <action-other-subotps>
    [TAB_GEN[:<tab-args>]]
)

<tab-args>=<idx>_<colname>_<width>[_L] [%%<tab_args>]{0,n}
[,titleidx]
```

Each field entry has to be separated by a double percent character "%%", as this is a parameter for the table processor itself, not the cli.

TAB_GEN

The generic table processor to be invoked for evaluation of table parameters.

<idx>

The canonical field index as provided by the ACTION. For the display of the actual values refer to "TITLEIDX" .

<colname>

The name of the column to be displayed in the table header.

<width>

The width of the table, which will cut the entry to the given value, if the size is exceeded.

B

This optional key switches to clipping and insertion of a break. The table is expanded in it's length for each of required breaks. The cut is made arbitrarily, without recognition of the actual semantics within the specific field.

REMARK:

In current version "B" is not compatible with the SORT option.

L

This optional key switches to leftmost cutting of fields, clipping it to it's trailing part, when the <width> is exceeded.

title,titleidx,machine

Any ACTION has to support a mandatory suboption TITLEIDX in addition to the implementation of TAB_GEN.

The "TITLEIDX" option displays the titles of each field with its positional index parameter as supported by the canonical record for MACHINE suboption. This is the index to be used by the underlying generic awk-script for the positions to be printed.

The <colname> parameter is case sensitive and therefore displayed literally. Restrictions for the available character set are the exclusion of reserved ctys-characters and the exclusion of any WHITESPACE, including CR. Comments and Whitespaces within the macro file are ignored.

4.7 Generic Records

The GENERIC class of actions LIST and ENUMERATE, additionally 'ctys-vhost' support data to be displayed in record formats.

The output could be adapted by the user with the same suboptions as for generic tables, but representing a line oriented attribute output.

The following formats are currently available:

1. REC

A propriatary record format:

```
record(#rec-idx):={
  {#field-idx, attr-name, attr-val},
  {.....
}
```

2. SPEC

A meta-data record format for testing of data with easy readabilty:

```
record(#rec-idx):={
  #field-idx          attr-name: attr-val
  #field-idx          attr-name: attr-val
  #field-idx          attr-name: attr-val
  .....
}
```

3. XML

An export format for post-processing:

```
<record index=#rec-idx>
  <attr-name index=#field-idx>attr-val</attr-name>
  <attr-name index=#field-idx>attr-val</attr-name>
  .....
</record>
```

.

Chapter 5

Address Syntax

This document describes the common generic address syntax for the command line interface. Additional data interfaces e.g. for LDAP and SNMP exist and may support additional features.

This specification describes the common generic address syntax for the management of single machines and groups of entities. This suffices all supported systems and may for some plugins applicable as a subset only.

The current version provides almost only the `<machine-address>` and the **GROUPS** objects, thus the remaining definitions were required for the design of an extendable overall concept.

5.1 Basic Syntax Elements

The addressing facility including the namebinding is splitted into a logical description as a general view and it's concrete adaptations which could be implemented by multiple presentations. The foreseen and implemented syntax scanners are designed to allow implementation in a straight-forward manner allowing an simple implementation of hierarchical structured syntax definitions by nested loops.

The following characters are reserved syntax elements, the full set and description is given in the chapter "Options Scanners - Reserved Characters".

'= **Option/Attribute Assignment** Separator for an option and it's suboptions.

E.g. the CREATE action may be proceeded for a new client to be connected to a running server, here the default type '-t VNC'.

```
ctys -a create=CONNECT ... myAccount@myHost
```

',' **Option/Attribute Separator** Separator for suboptions belonging to one set of suboptions.

E.g. a hanging machine may be rebooted, where the hypervisor is utilized to a forced reboot of the locked guest os.

```
ctys -t QEMU -a cancel=REBOOT,FORCE ... myAccount@myHost
```

':' **Option/Attribute Value Assignment** Separator for a suboption key and it's arguments.

The string representation of the identifier for a session.

```
ctys -a create=LABEL:UserString ... myAccount@myHost
```

'%' **Option/Attribute Multiple-Value Separator** Separator for suboption argument values.

E.g. VMware 2.x a credential is required, which could be either suppressed and later provided by the opening dialogue mask, or provided in clear on commandline. A secure alternative may be the application of Kerberos.

```
ctys -t VMW \  
-a create=LABEL:MyMachine,USER:myAccountA%mySecretPasswd \  
myAccountB@myTargetHost
```

'()' **Option/Attribute Sets** Grouping target specific context-options belonging to a common target a.k.a. host.

E.g. the previous example could be written as:

```
ctys myHostAccount@myTargetHost' ( \
    -t VMW \
    -a create=LABEL:MyMachine,USER:myVmAccount%mySecretPasswd\
  )'
```

This particularly enables the superposition of specific attributes for each of multiple targets:

```
ctys -t VMW \
  myAccount@myTargetHost00' ( \
    -a create=LABEL:MyMachine,USER:myAccount00%mySecretPasswd00 \
  )' \
  myAccount@myTargetHost01' ( \
    -a create=LABEL:MyMachine,USER:myAccount01%mySecretPasswd01 \
  )'
```

The following arguments handle in general groups of syntax elements defining a set of elements as a group. This comprises the definition of a set of elements on the same level - as siblings, and the definition of a vertical structure of a path - parent-child relations. The contained elements could be targeted instances as well as sets of attributes for a specific instance or a group of instances.

',' **Path-Operator** Separator for sets of attributes and targets on the same level.

The following entities belong to a common level.

```
ctys -t create=LABEL:myTest host00,host01,host02
```

Due to supported common legacy syntax of host addressing the previous example could also be written as

```
ctys -t create=LABEL:myTest host00 host01 host02
```

',' **Set-Operator** Separator for paths of nested attributes and targets.

The address of VM running on a specific physical machine could be written as:

```
ctys -t create=LABEL:myTest myAccount@host00'(-z)'.vm01'(-t VMW -a create=LABEL:...)'
```

'' **Group Operator** This is the common set operator for grouping related syntax elements. Thus braces could be applied to attributes as and to containing instances. This may be particularly required to resolve ambiguity when third-party tools with independent call syntax may be utilized.

(a) **Attribute Groups**

The following application shows the application for grouping of syntax-element when a MACRO is applied.

```
ctys '{macro:tst-subgroups-01}'(-d 99999)'
```

Where the macro is defined as:

```
tst-subgroups-01 = -a list {host1 host2}
```

The final expanded syntax is:

```
ctys -a list host1'(-d 99999)' host2'(-d 99999)'
```

(b) **Target Groups**

Grouping arguments for multiple targets including their specific options belonging to a common high-level-target a.k.a. SUBTASK

The following entity defines a group of entities as a new instance, which is a new instance consisting of the set of its members including their context specific suboptions.

```
myGroup00:={
  myTargetHost00'(-t VMW -a create=LABEL:MyMachine00a,USER:myAccount00%mySecretPasswd00)',
  myTargetHost00'(-t VMW -a create=LABEL:MyMachine00b,USER:myAccount00%mySecretPasswd00)',
  myTargetHost01'(-t QEMU -a create=LABEL:MyMachine01a)'
}
```

The new group instance could be applied as:

```
ctys myGroup00
```

The destination address of each target could be named by any element of it's <machine-address> as described in the following chapter. In some cases a single attribute may be ambiguous within a distributed multi-user environment. This could for example be true for the hostname of a single virtual machine. Thus multiple attributes may be required for a unique address.

'[]' Pre-Attribute-To-Address Conversion Grouping multiple attributes to a unique attribute set for unambiguous resolution to target addresses.

The following entity defines a host unambiguously. This could be applied for example to test machines, which may be a simple backup, or redundant due to intention.

In case of the UnifiedSessionsManager the automated inventory scanner registers by default each present machine without redundancy checks for specific attributes. Thus an identical copy running in physical different private subnets behind NAT routers with identical IP ranges may occur redundant for a variety of attributes. These targets could be addressed e.g. in the following way.

(a) Check execution state

```
ctys [ LABEL:myMultiHost,BASEPATH:/mntn/myTestPool01/myHost007 ]
```

(b) Check run state

```
ctys -a show [ LABEL:myMultiHost,BASEPATH:/mntn/myTestPool01/myHost007 ]
```

(c) Check static information

```
ctys -a info [ LABEL:myMultiHost,BASEPATH:/mntn/myTestPool01/myHost007 ]
```

(d) Display inventory data

```
ctys-host [ LABEL:myMultiHost,BASEPATH:/mntn/myTestPool01/myHost007 ]
```

The brackets pre-resolve the attribute set to an appropriate target address.

The current syntax description may not yet formally be absolutely correct nor complete, but may cover the intended grade of open description and required understanding for it's application. Some modifications are still under development.

5.2 SyntaxExamples

The previous elements provide for flexible and simplified addressing of hosts and contained applications. This could be as simple as

```
ctys-host -a create=LABEL:myDesktop
```

which starts a VNC desktop on '\$USER@localhost' and assigns the symbolic name 'myDesktop' to it. Same for a remote host '\$USER@myHost'

```
ctys-host -a create=LABEL:myDesktop myHost
```

which starts a VNC desktop(the configured default) on the host 'myHost' and assigns the symbolic name 'myDesktop' to it. The so called label 'myDesktop' could be used as a full scale address id for all further calls.

When required also a some more sophisticated call of a VM stack could be performed too:

```

ctys \
  [ UUID:3f95fea7-ee51-445b-95ee-4e432e6e4187 ]\
  .{ \
    [ LABEL:myMultiHost,BASEPATH:/mntn/myTestPool01/myHost007 ]'( \
      -t VMW -a create=LABEL:MyMachine00a,USER:myAccount00%myPasswd00\
    )'\
    ,\
    myTargetHost00'( \
      -t VMW -a create=LABEL:MyMachine00b,USER:myAccount00%myPasswd00\
    )', \
  }'(-d 99999)' \
  ,\
  myTargetHost01'( \
    -t QEMU -a create=LABEL:MyMachine01a\
  )'

```

5.3 AddressSyntaxElements

The following namebinding defines the superset of addressing attributes, which supports explicit addressing of targets as well as generic addressing of single and multiple targets by using search paths and content attributes in analogy to wildcards, a.k.a. keywords or attribute value assertions. The given sub-options are defined by default not to be order dependent, but some may influence the remaining. The keywords are case-insensitive.

The contained parenthesis, angle, and square brackets in the following figures are syntactic helpers. When they are part of the syntax, they will be quoted with single quotation marks.

The addressed top-level entity is the APPLICATION, thus here the `<target-application-entity>`. This contains in analogy to the OSI model the machine as well as the access point.

```

<target-application-entity>:=<tae>
<tae>:=[<access-point>]<application>

<access-point>:={
  <physical-access-point>

\begin{center}\begin{tabular}{l}
<virtual-access-point> \\
<dialogue-access-point> \\
\end{tabular}\end{center}

}

<application>:=<host-execution-frame><application-entity>

<physical-access-point>:=<machine-address>[:<PM-access-port>]
<virtual-access-point>:=<machine-address>[:<VM-access-port>]
<dialogue-access-point>:=<machine-address>[:<HOST-access-port>]

```

Figure 5.1: TAE - Target Application Entity address

The machine is addressed by the `<machine-address>`, which represents physical and virtual machines as well as login-sessions provided by the HOSTs plugin. The specific plugins may support a

subset of the full scope, but the attributes **ID** and **LABEL** are supported in any case. The **ID** attribute is here either a persistent identifier, in case of a VM a configuration file, or a dynamic identifier in case of the HOSTs plugin, e.g. for VNC the DISPLAY number excluding the port-offset. Whereas it is defined for X11 as the PID.

```

<machine-address>:={
  (
    [(ID|I|PATHNAME|PNAME|P):<mconf-filename-path>][,]
    |
    [(ID|I):<id>][,]
  )
  [(BASEPATH|BASE|B):<base-path>[%<basepath>]{0,n}]
  [(LABEL|L):<label>][,]
  [(FILENAME|FNAME|F):<mconf-filename>][,]
  [(UUID|U):<uuid>][,]
  [(MAC|M):<MAC-address>][,]
  [(TCP|T):<TCP/IP-address>][,]
}

```

Figure 5.2: Machine-Address

The type of access varies for the different `<access-point>`. Therefore the `<access-port>` has to be provided in several variants which are specific for the various products. The following figure depicts the major items, additional may be required, which are described within the specific subsystem.

```

<PM-access-port> := (CLIcon|RDPCon|VNCcon|X11con)
<VMaccess>       := (QEMUcon|VBOXcon|VMWcon|XENcon)
<HOSTaccess>     := (CLIcon|RDPCon|VNCcon|X11con)

```

Figure 5.3: `<access-ports>`

These items represent the various plugins, particularly the HOSTs plugin as a set of major desktop and console plugins commonly used for dialogue and batch access to the VMs and PMs.

Each of the listed access types provides several additional options related to the protocol access point and additional feature parameters. These parameters are described within the specific plugins documents.

```

CLIcon := (SHELL)
QEMUcon := (QEMU-SDL|CLI|VNC|X11)
RDPCon := (RDESKTOP)
VBOXcon := (VirtualBox|VBoxSDL|RDP)
VMWcon := (VMware|vmware-rc|firefox|VNC)
VNCcon := (VNCVIEWER)
X11con := (XTERM|GTERM|EMACS|EMACSM|EMACSA|EMACSAM)
XENcon := (CLI|VNC|X11)

```

Figure 5.4: CONSOLE types

`<access-point>` The complete path to the execution environment.

`<access-port>` The port to be used on the access-point.

`<application>` The application itself, which has to be frequently used in combination with a given service as runtime environment.

`<application-entity>` The executable target entity of the addresses application, which could be an ordinary shell script to be executed by a starter instance, or an selfcontained executable,

which operates standalone within the containing entity. E.g. this could be a shared object or an executable.

(basepath|base|b):<base-path>1,n Basepath could be a list of prefix-paths for usage by UNIX "find" command. When omitted, the current working directory of execution is used by default.

(filename|fname|f):<mconf-filename> A relative pathname, with a relative path-prefix to be used for down-tree-searches within the given list of <base-path>.

So far the theory. The actual behaviour is slightly different, as though as a simple pattern match against a full absolute pathname is performed. Thus also parts of the fullpathname may match, which could be an "inner part". This is perfectly all right, as far as the match leads to unique results.

More to say, it is a feature. Though a common standardname, where the containing directory of a VM has the same name as the file of the contained VM could be written less redundant, when just dropping the repetitive trailing part of the name.

<host-execution-frame> The starter entity of addressed container, which frequently supports a sub-command-call or the interactive dialog-access of users to the target system.

(id|i):<mconf-filename-path> The <id> is used for a variety of tasks just as a neutral matching-pattern of bytes, an in some cases as a unique VM identifier within the scope of single machine. The semantics of the data is handled holomorphic due to the variety of utilized subsystems, representing various identifiers with different semantics. Thus the ID is defined to be an abstract sequence of bytes to be passed to a specific application a.k.a. plugin, which is aware of it's actual nature.

The advantage of this is the possibility of a unified handling of IDs for subsystems such as VNC, Xen, QEMU and VMware. Where it spans semantics from being a DISPLAY number and offset of a base-port, to a configuration file-path for a DomU-IDs, or a PID of a "master process".

This eases the implementation of cross-over function like LIST, because otherwise e.g. appropriate access-rights to the file are required, which is normally located in a protected subdirectory. These has to be permitted, even though it might not be required by the actual performed function.

(LABEL|L):<label> <label>={ [a-zA-Z-_0-9]{1,n} (n<30, if possible)}

User defined alias, which should be unique. Could be used for any addressing means.

(MAC|M):<MAC-address>

The MAC address, which has basically similar semantically meaning due to uniqueness as the UUID.

Within the scope of ctys, it is widely assumed - even though not really preredquired - that the UUIDs and MAC-Addresses are manual assigned statically, this could be algorithmic too. The dynamic assignment by VMs would lead to partial difficulties when static caches are used.

<mconf-filename> The filename of the configuration file without the path-prefix.

<mconf-filename-path> The complete filepathname of the configuration file.

<mconf-path> The pathname prefix of the configuration file.

(PATHNAME|PNAME|P):<mconf-path> When a VM has to be started, the <pathname> to it's configuration file has to be known. Therefore the <pathname> is defined. The pathname is the full qualified name within the callers namespace. SO in case of UNIX it requires a leading '/'.

<physical-access-point>:=<machine-address>[:<access-port>] The physical termination point as the lowest element of the execution stack. This is the first entity to be contacted from the caller's site, normally by simple network access.

<target-application-entity> The full path of the stacked execution stack, addressing the execution path from the caller's machine to the terminating entity to be executed. This particularly includes any involved PM, and VM, as well as the final executable. Thus the full scope of actions to be performed in order to start the "On-The-Top" executable is contained.

(TCP|T):<tcp/ip-address> The TCP/IP address is assumed by ctys to assigned in fixed relation to a unique MAC-Address.

(UUID|U):<uuid> The well known UUID, which should be unique. But might not, at least due to inline backups, sharing same UUID as the original. Therefore the parameter FIRST, LAST, ALL is supported, due to the fact, that backup files frequently will be assigned a name extension, which places them in alphabetical search-order behind the original. So, when using UUID as unique identifier, a backup will be ignored when FIRST is used.

Anyhow, cross-over ambiguity for different VMs has to be managed by the user.

<virtual-access-point>:=<machine-address>[:<access-port>] The virtual termination point as an element of the execution stack. The stack-level is at least one above the bottom This stack element could be accessed either by it's operating hypervisor, or by native access to the hosted OS.

5.4 Stack Addresses

The stack address is a logical collection of VMs, including an eventually basic founding PM, which are in a vertical dependency. The dependency results from the inherent nested physical execution dependency of each upper-peer from it's close underlying peer. Therefore the stack addresses are syntactically close to **GROUPS** with additional specific constraints, controlling execution dependency and locality. Particularly the addressing of a VM within an upper layer of a stack could be smartly described by several means of existing path addresses schemas. Within the UnifiedSessionsManager a canonical form is defined for internal processing(Section 3.3.4 '**Stacks as Vertical-Subgroups**' on page 40), which is available at the user interface too. Additional specific syntactical views are implemented in order to ease an intuitive usage. The following section depicts a formal meta-syntax as a preview of the final ASN.1 based definition. A stack address has the syntax as depicted in Figure~5.5.

```

<stack-address>:=<access-point-list>

<access-point-list>:={
    <physical-access-point>
    |<virtual-access-point-path-list>
}

<virtual-access-point-path-list>:={
\begin{center}\begin{tabular}{l}
<empty> \\
<virtual-access-point> \\
\end{tabular}\end{center}

        ['.'<virtual-access-point-path-list>]

\begin{center}\begin{tabular}{l}
[[','<virtual-access-point-path-list>] \\
\end{tabular}\end{center}

}

```

Figure 5.5: Stack-Address

A stack can basically contain wildcards and simple regexpr for the various levels, groups of entities within one level could be provided basically to. And of course any MACRO based string-replacement is applicable. But for the following reasons the following features are shifted to a later version:

Wildcards: An erroneous user-provided wildcard could easily expand to several hundred VMs, which might be not the original intention. Even more worst, due to the detached background operation on remote machines, this can not easily be stopped, almost just by reboot of the execution target and probably any additional involved machine. Which, yes, might take some time, due to the booting VMs.

Level-Groups/Sets: Due to several higher priorities this version supports explicitly addressed entries only.

5.5 Group Resolution

The GROUPS objects are a concatenation of <machine-addresses> and nested GROUPS including specific context options.

```

<group-address>:= (
    [ <machine-addresses>['(' <machine-options> ')']['(',')']{0,n}]
    [ <group-address>['(' <group-options> ')']['(',')']{0,n}]
)[ '(' <group-options> ')']

```

Figure 5.6: Group-Address

Groups are valid replacements of any addressed object, such as a HOSTs. Groups can contain in addition to a simple set of hostnames a list of entities with context specific parameters and include other groups in a nested manner. Each set of superposed options is permuted with the new included set.

The resolution of group names is processed by a search path algorithm based on the variable , **CTYS_GROUPS_PATH** , which has the same syntax as the PATH variable. The search algorithm is a first-wins filename match of a preconfigured set. Nested includes are resolved with a first-win algorithm beginning at the current position.

In addition to simple names an absolute or relative pathname for a group file could be used. For common addressing as the path-seperator a '.' could be used. In case of a '.' the user has to be aware of possible ambiguity, when file extensions are used. The address resolution works on first-match-wins base.

This allows for example the definition of arbitrary categories, such as server, client, desktop, db, and scan. Here are some examples for free definitions of categories based on simple subdirectories to search paths. The level of structuring into subdirectories is not limited.

server/* A list of single servers with stored specific call parameters. Server is used here as a synonym for a backend process. Which could be either a PM or a VM, the characteristics is the inclusion of the backend process only.

client/* A list of single clients with stored specific call parameters. This is meant as the user front end only, which could be a CONNECTIONFORWARDING. The user can define this category also as a complete client machine including the backend and frontend, which is a complete client for a service.

desktop/* A composition of combined clients and servers for specific tasks. This could be specific desktops for office-applications, systems administration, software-development, industrial applications, test environments. Either new entries could be created, or existing groups could be combined by inclusion.

db/* Multiple sets of lists of targets to be scanned into specific caching databases. This could be used for a working set as well as for different views of sets of machines.

scan/* A list of items to be scanned by tools for access validation and check of operational mode. Therefore this entities should contain basic parameters onyl, such as machine specific remote access permissions type.

REMARK: The group feature requires a configured SSO, either by SSH-Keys or Kerberos when the parallel or async mode is chosen, which is the default mode. This is required due to the parallel-intermixed password request, which fails frequently.

For additional information on groups refer to "GroupTargets" and "ctys-groups" .

The GROUPS objects are a concatenation of <machine-addresses> and nested GROUPS including specific context options. The end of the command with it's specific option should be marked by the common standard with a double column '↵'.

```

ctys -a <action> -- '(<glob-opts>)' <group>'(<group-opts>)'
=> The expansion of contained hosts results to:
...
<host0>'(<host-opts> <glob-opts> <group-opts>)'
<host1>'(<host-opts> <glob-opts> <group-opts>)'
...
=> The expansion of contained nested groups results to:
...
<group-member0>'(<glob-opts>)'(<group-opts>)'
<group-member1>'(<glob-opts>)'(<group-opts>)'
...

```

Figure 5.7: Groups of Stack-Addresses

The context options are applied succesively, thus are 'no-real-context' options, much more a successive superposition. More worst, the GROUP is a set, thus the members of a group are reordered for display and execution purposes frequently. So the context options are - in most practical cases - a required minimum for the attached entity.

5.6 ctys-help-on

SYNTAX

<ctys-command>

-H <help-option>

```
<help-option>:=
    (man|html|pdf) [=((1-9)|<help-on-item>[,<help-on-item-list>])
    | (path|list|listall)
    | funcList=<any-function> [@<module-name>[@...]] [,<any-function>...]
    | funcListMod=<any-function> [@<module-name>[@...]] [,<any-function>...]
    | funcHead=<any-function> [@<module-name>[@...]] [,<any-function>...]
    | _ONLINEHELP_
    | _HELP_
    )
```

DESCRIPTION

The **-H** option is the common generic option of all tools for the display of online help.

The default is the display of man pages within a commandline terminal. This could be any valid document within the search list defined by the variable MANPATH. The output format could be optionally specified as PDF and HTML documents.

This tool is also used within menu entries of the XDG desktop of Freedesktop.org for graphical display of online help. Therefore the current version provides the simple HTML lists **doc.html** for the **DOC-Package**, and the **base.html** file for the **BASE-Package**.

REMARK:

For the **-H** option of the call 'ctys -H man' and 'ctys-vhost -H man' the man parameter is mandatory. In all other cases the call '<any-other-ctys> -H' searches for 'man' output by default within MANPATH.

. OPTIONS

The following suboptions and parameters could be applied:

-H path

Displays current document and man path.

-H list

Lists available online documents and manpages.

-H listall

Lists available online documents and manpages including the documents available by MANPATH.

-H (man|html|pdf)[=([1-9]**)[<help-on-item-list>])**

Displays the requested information with one of the formats **man**, **pdf**, or **html**. The following viewers are preconfigured as shell variables within the configuration files and can be adapted as required:

```
CTYS_MANVIEWER=man
```

```
CTYS_PDFVIEWER=(acroread else kpdf else gpdf)
```

```
CTYS_HTMLVIEWER=(konqueror else firefox)
```

Default is **manpage** for the current process with **man 1 ...**. Additional constraints could be applied such as another man-section or a filename, which could be either literally matching or a string to be expanded. In case of expansion the first match is taken.

-H funcList=[<any-function>][@<module-name>[@...]]

List of function names, sorted by function names. In addition the file names and line numbers are displayed too.

-H funcListMod=[<any-function>][@<module-name>[@...]]

List of function names, sorted by file names. In addition the file names and line numbers are displayed too.

-H funcHead=[<any-function>][@<module-name>[@...]]

Displays the contents of function headers, sorted by file names. The following constraints could be applied:

- <any-function>: If given <any-function> than only this is displayed.
- <module-name>: If given <module-name>, than the functions contained within this module only are displayed.

-H (_ONLINEHELP _ | _HELP _)

Displays the predefined online help for the installed package.

. EXAMPLES**<ctys-command> -H (_ONLINEHELP _ | _HELP _)**

Displays the predefined online help for the installed package.

<ctys-command> -H html=base

Displays a summary of links for all documents contained in the **BASE package**.

<ctys-command> -H html=doc

Displays the extended online help as contained in the **DOC package**.

<ctys-command> -H list

Lists available online documents and manpages.

<ctys-command> -H ctys

Displays the **manpage** for ctys with **man**.

<ctys-command> -H man=ctys

Displays the **manpage** for ctys with **man**.

<ctys-command> -H html=ctys

Displays the **manpage** for ctys with CTYS_HTMLVIEWER, by default **firefox** or **konqueror**.

<ctys-command> -H pdf=ctys

Displays the **manpage** for ctys with CTYS_PDFVIEWER, by default **kpdf**, **gpdf**, or **acroread**.

<ctys-command> -H pdf=howto

Displays the **ctys-howto-online.pdf**, which is displayed in alphabetical order before **ctys-howto-print.pdf**.

<ctys-command> -H pdf=howto-print

Displays the **ctys-howto-print.pdf**, which is the first appropriate match.

<ctys-command> -H pdf=command-ref

Displays the **ctys-command-reference.pdf**.

<ctys-command> -H html=CLI,X11,VNC,VMW

Displays the **manpage** for ctys-CLI, ctys-X11, ctys-VNC and ctys-VNM with CTYS_HTMLVIEWER, by default **firefox** or **konqueror**. For incomplete names a search with **find** is utilized for name expansion.

<ctys-command> -H html=ctys-extractARPlst,extractMAClst

Displays the **manpage** for ctys-extractARPlst and ctys-extractARPlst with CTYS_HTMLVIEWER, by default **firefox** or **konqueror**. For incomplete names a search with **find** is utilized for name expansion.

Chapter 6

CTYS Call interface

6.1 Common Elements and Behaviour

6.1.1 Sessions Namebinding

The current version binds the session to its plugin type as an internal selector for required runtime components. The data identifying a specific session is required for the identification of the individual instances.

The combination of the plugin type and the session identifier selects a specific session and the related runtime components.

Even though the key pair allows the usage of address elements within multiple session types, this should be avoided.

6.1.2 Cache for Performance and Collections

The name resolution for static and dynamic data could be performed basically by two approaches:

- from cached data by ctys-vhost and related tools
- by polling of actual runtime data with "ctys -a LIST" call.

Both approaches have their advances and impacts as any cache system has.

When using the cache, this obviously has to be up to date, when a data match occurs. When no match could be get, the realtime polling will be used by default.

The performance impact of the realtime poll vs. cached data is frequently a factor more than 10, which is 5-15 seconds vs. 0.5 seconds. The usual case is even more dramatic, usually 30 to 60 seconds, for CREATE, could be even much longer, depending from the file system to be scanned.

In cases of runtime systems scan via LIST it depends on the actual load. This approach is normally somewhat different from the pure CREATE call, because just a part of the data could be used from the cache DB, which just represents "potential candidates", the actual runtime state has to be polled in any

case, but could be done much more efficient of course.

But anyhow, in specific situations the usage of the poll approach is the only one applicable. The systems bootstrap for example is normally performed with an empty cache DB, so no decision is required. This is particularly the case, due to the usage

of ENUMERATE itself for evaluation and collection of the cache DB.

The cache behaviour could be commonly controlled by the following two keyword:

NOCACHE

Disables the usage of cached data for address resolution. Therefore the call of "ctys-vhost" and related utilities is disabled.

The default behaviour is to use CACHE first, which resolves for the low-medium reference system frequently within less than 0.5 seconds to a match. When no match found POLL is used, which is an internal ENUMERATE call and may have a performance impact of multiples of 10 and even more, depending on the size and structure of the scanned file system. The response time is frequently in the range of 30-60 seconds, and more.

When activated, at least polling has to be allowed, otherwise no resolution is possible.

NO POLL

Disables the polling of data, thus CACHE is required. A missing hit on cached data will lead to failing of resolution.

6.1.3 Access VM Stacks

The utilization of VM stacks, which are nested and therefore encapsulate each level requires for several methods some specific recognition and stack-awareness.

Almost each action requires its own specific handling.

CREATE

The CREATE action is almost only effected specifically, when a stack-level is requested to be activated, for which the containing (n-1)-level stack entity is not yet present, this statement includes any lower missing entity too. Therefore it has to be decided whether the request should be rejected or an implicit stack-recovery has to be performed. When the decision is made for implicit recovery, the algorithm could be said to

"process-upward-from-highest-existent".

CANCEL

The CANCEL action is effected by stack-awareness when a stack entity is requested to be canceled, which has active contained upper stack entities. In this case the decision to be made are one or a combination of

- propagate CANCEL request to upper stack
- use hypervisor only, ignore upper stack
- "kill" if supported (e.g. not Xen) the current level hypervisor
- CANCEL the execution target, which is the execution level of stack "itself", e.g. essential for PMs.

When the decision is made for state-change-forward-propagation, the algorithm could be said to

- (a) "walk-upward-to-all-highest-existent"
- (b) "process-downward-to-request-target"

LIST

This action has to be stack-aware on request, it is the most beneficial dynamic information collector for the user in order to manage VM stacks.

SHOW

This action should be stack-aware on request, it is a beneficial dynamic information collector for the user in order to manage VM stacks.

ENUMERATE

This action must not be stack aware, what is due to the practical circumstance, that the destination VMs has to be active, when to be enumerated for the executable VMs. Thus, as long, as not the offline VM images could be scanned (which will be obviously never the case for all VMs), this call would have lead to the activation of ALL VMs executable, thus it probably could be said, will never finish due to bulk-start-performance-impact!

INFO

Basically almost the same as ENUMERATE.

Due to the impact of stack awareness for execution, this behaviour could be controlled for the methods. Almost each action requires it's own specific handling.

FORCE

Performs the requested action immediately by calling the hypervisor. No forward propagation is performed. The specific range of features for the various hypervisors could and do vary, thus will be specialized within the description of the specific plugin.

STACK

Performs the requested action first by state-propagation "deeper into the nested stack", which means upward due to ISO-like stack model. The behaviour when failing with state-change-request propagation depends on the actual request.

The hypervisor might be called as final anchor in case the propagation fails or timeouts for a CANCEL request, because otherwise the request may fail, even though it could have been finished by usage of the hypervisor. A decision whether or not to perform a "final and ultimate kill" on the stack is not obvious due to the hidden upper part, which basically could include any highly sensitive task, resulting in crucial failure when canceled abrupt. This risk will remain for generic actions, and has to be handled by the application and it's close systems within the encapsulated stack entities.

For the CREATE action - which is in case of STACK an iterative upward execution - the call of the hypervisor is necessarily the first call on each level. Thus in case of CREATE in contrast to the CANCEL action the transparency of the stack is system immanent. The call of the hypervisor could be followed by an state-change-propagation request after success only. The remaining question for partial execution is here what to do with an incomplete VM stack.

The answer for now is "let him live". This has to be monitored and handled appropriately by the user when the "-b on" option for asynchronous background operations is choosen. In case of synchronous operations the job will be canceled, but also no roll-back is performed. Just the monitoring is implied due to synchronity. The dynamic information poll methods LIST and SHOW just ignore a propagation failure, this is the same behaviour as in the case that nothing was found.

SELF

The SELF keyword is currently for CANCEL action used only, but will of such basic level of understanding, that it is shown here. SELF causes the inclusion

of the execution target for requested operations too.

This is due to the technical requirement, that any method for an stacked entity has to be initially performed within it's container entity, which hosts the bottom controller instance of the targeted stack-level - it's hypervisor - for ultimate superposing action. The basic call structure

```
ctys -a <action>=<action-target> <execution-target>
```

of ctys is splitted into the <execution-target> where the first communications peer entity resides, and the <action-target>, which is the destination of <action> to be performed.

The Stack Essentials - CREATE and CANCEL

One of the most beneficial features for stacked VMs would be the implicit creation of missing sessions within the execution stack. Anyhow, even though it is basically possible to do this with some current VM implementations, this feature is shifted.

The reason is simply the lack of a common strategy within the used VMs, where at least XEN and QEMU would be suitable, but others not for an unattended and SERVERONLY iterative CREATE chain. In avoidance of a quick-shot resulting in some "savage-variants" when coming to details, the decision for this version is not to implement the automatic and implicit creation of inactive intermediate stack entities.

Therefore this feature is currently shifted and will be implemented within a later version. For now each stack entity has to be CREATED seperately and explicitly by the user, in manual iteration. For CANCEL anything stays as it is, stacked operations are supported as described.

.

6.2 ctys

ctys

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

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.

6.2.1 Installation and Setup

The installation and configuration is described 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.

6.2.2 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.

6.2.3 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.

6.2.4 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.

6.3 OPTIONS

6.3.1 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**

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>|-)] [%<domain>]]]
)

```

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>|-)[%<domain>]]

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

CREATE

```

CREATE=<machine-address>{1}
[
  CONNECT
  | REUSE
  | RECONNECT
  | RESUME
] [,]
[USER:<user>[%[(<credentials>|-)] [%<domain>]]] [,]
[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 described 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 its 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 plugging. 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 UnifiedSessionManager 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 6.1: 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 6.2: Supported Install-Targets

<boot-img-pathname>

Optional pathname to an alternative 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 accessible 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 6.3: 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 6.4: 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 during 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 accessibility 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 and 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 separate 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>|-)][%<domain>]]

The account to be used for native access to the <action-target>. In addition to the user name and credentials an optional domain name for the user account - e.g. fo MS-Windows(TM) based accouts - could be provided. Missing parameters are requested during authentication.

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.

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 con-

figuration. 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 templates 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 statically 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 separate 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 separated 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 6.5: 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 capabilities supported.	-
22	STACKREQ	The list of capabilities 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 GuestOS.	-
39	PLATFORM	Virtual device.	-
40	VRAM	The pre-configured amount of RAM.	-
41	VCPU	The pre-configured number of V-CPU's.	-
42	CONTEXTSTRG	A private context storage for the plugin	-
43	USERSTRING	A string to be customized by the user.	-

Table 6.6: Output Record-Format for MACHINE suboption

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"
```

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 6.1: INFO Display

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>|-)] [%<domain>]]] [,]
]

```

LIST displays 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-persistent 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 performance 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 separated 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 separator, 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 have 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 6.7: 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 6.8: 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.

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 6.2: LIST Display

6.3.2 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 of 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 DETACHED 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 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 be 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.

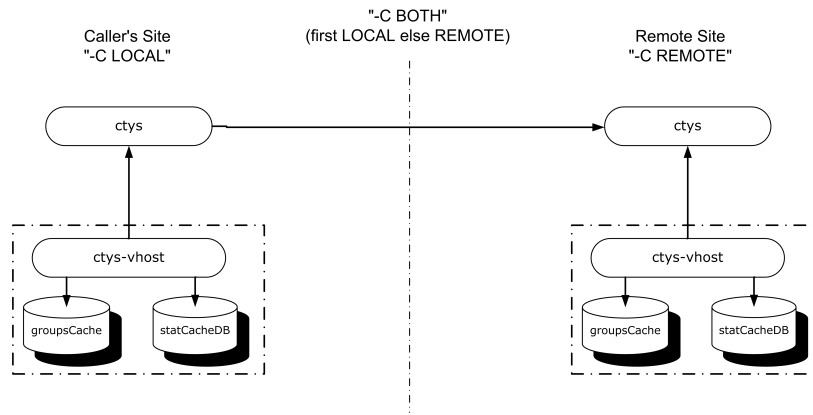


Figure 6.3: Distributed Caches

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 handled 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 its actually present parts, and inserted as the first argument to the relevant ACTION of the current job from the internal scheduler, 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 on 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 caching, 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 `SESSION-CACHEPERIOD` (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#1000000000000000=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 reasons 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 separate 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 explicitly 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 its 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>

Restricts/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 preferred.

- **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 authenticated 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 "**requiretty**" 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.

6.4 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:

[H]

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 permuted 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.

Chapter 7

Plugins - Feature Extensions

This section contains help for the add-on plugins contained in the distribution of ctys.

These are designed to be executed on "PM" which are PhysicalMachines in addition to "VM" which are VirtualMachines.

7.1 Prerequisites

Following pre-requisites have to be prepared for the listed plugins. The specific plugins may have additional requirements, such as installation of a specific GuestOS and usage of specific drivers. These are listed for the plugins specifically.

	VDE TAP	sudo ksu	kernel	HOST OS
CLI				L,B
X11				L,B
VNC			X	L,B
QEMU	X	(X)	(1)	L,(3)
VBOX			X	L
VMW			X	L
XEN		X	X	L,(5)
PM			(2)	L,B

Table 7.1: Prerequisites and applicability of plugins

- (1) KQEMU and KVM is currently not supported within stacks, could be used only for the "lowest" VM of a stack.
- (2) Depends on upper stack, for VMs almost in any case required.
- (3) Currently Linux only is tested, but following will be supported soon:
 - FreeBSD, NetBSD, OpenBSD
 - OpenSolaris
- (4) Currently Linux only is tested, but following will be supported soon:

- NetBSD (OpenBSD, and FreeBSD as soon as available)
- OpenSolaris

L Linux: The preferred distribution is CentOS-5.x.

B BSD: Currently OpenBSD only, but FreeBSD and NetBSD will follow.

A common requirement - as already mentioned - is to establish a distributed Authentication and Authorization system, e.g. SSO. Otherwise a recognizable amount of password demands might occur.

Any supported GuestOS of listed VMs will be supported. For non-listed HOST-OSs used as GUESTOS no native access based on CLI, X11, or VNC exists, so for CREATE and CANCEL the hypervisor could be used only. A controlled shutdown is particularly only possible by means of the controlling hypervisor. So no stack-propagation will be performed.

7.2 Category HOSTs

This section contains the add-on plugins of category HOSTs, which are executed native on OS level for native interaction with hosting GuestOSs as well as for control sessions to the containing OS of the targeted hypervisor.

These are the standard remote sessions based on CLI/Console, X11, and VNC, the workhorses of the UnifiedSessionsManager for interactive user sessions as well as for remote execution of batch commands.

7.2.1 CLI

SYNTAX

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

DESCRIPTION

The CLI-plugin starts a remote shell from within a local shell. No specific default desktop functionality such as XTerm is supported, just a pure CLI access is performed. The handling of desktop windows such as XTerm or GTerm is supported by the X11-Plugin. The default and standard usage of the X11 displayforwarding by activation of the **'-X'** option of the underlying OpenSSH provides the usage of the CLI-plugin as a scripting and automation starter for any X11 application. This provides particularly any application with a proven on-demand encryption for the network interconnection by seamless integration into the **ctys** user interface. An Xterm, gnome-terminal or any arbitrary X11 tool could be started from within a CLI sub-option "CMD". Chained logins by gateways with Overall-Display-Forwarding is supported by OpenSSH, thus by **ctys** too.

When executing CLI almost the same functionality as for an ordinary SSH session is supported. The main advance of using **ctys** instead of an ordinary **ssh**-call is the seamless integration into the **ctys** environment. Resulting from this feature instead of an IP address the `<machine-address>` could be used. Thus the administration of address-to-target-mapping could be completely delegated to **ctys** and will be utilized by internal usage of the full scope of **ctys-vhost**. E.g. the **LABELS** defined within **ctys** could be used to open a remote shell to any "labeled" instance such as a VM or PM. Therefore internal CLI type CONSOLES only use the CLI plugin.

Another quite important aspect is the persistency of any symbolic address entity. The complete set of addresses defined by symbolic elements such as labels is mapped to actual runtime addresses when required. Thus the automation of post-attachement of an X11 application could be easily stored into a script without additional effort for determination of the dynamic runtime identifier such as arbitrary display identifiers.

Another option might be the usage of UUIDs or MAC-Addresses for persistent definition of the handling of commands on frequently changing and reinstalled systems with reassigned IP-Addresses and DNS names. This is provided for any system by dynamic mapping.

The second important feature of CLI is the facility to start native remote commands as shell calls within any managed PM or VM instance. Therefore CLI is the working horse for native execution of GUI-less ctys-tasks. Similar to the call of a system command, any internal ctys library and plugins function could be called remotely by CLI sub-option "CMD".

The default shell used on the target instance is bash, which could be altered interactive by the sub-option "SHELL|S", or persistently by the variable `CLI_SHELL_CMD_DEFAULT`.

It should be mentioned and explained here that there are two quite similar sub-options, which on the first view seems to be redundant, which is actually not the case. The main **difference between** the **CMD** and **SHELL** suboptions is the behaviour. Whereas the SHELL just overlays the standard shell within a Session, the CMD suboption executes a command from within the executed shell - either the standard shell, or a pre-set custom shell.

Additional information containing use-cases with application examples is available from **ctys-uc-CLI**.

OPTIONS

-a action[=<suboptions>]

CANCEL

Not applicable.

CREATE

```
CREATE=[<machine-address>]
      [REUSE|CONNECT|RECONNECT|RESUME]
      [CONSOLE:<>]
      [(CALLOPTS|C):<callopts>]
      [CD:<working-directory>] [,]
      [(XOPTS|X):<xopts>]
      [(SHELL|S):<shell>]
      [(STUBMODE|STUB) [:on]] [,]
      [(CMD):<cmd>]
```

<machine-address>

See standard for syntax. Due to the limited attribute set of a completely dynamic CLI session without own persistent resources only the LABEL is applicable. The hosting OS has to be handled by it's own plugin. The LABEL sub-option is here mandatory.

CALLOPTS

Refer to common options description .

CD:<working-directory>

Change current working directory for the started command. This provides

for example the preconfiguration of specific work-sets, where each terminal could be opened for a specific directory.

CMD:<cmd>

Refer to common options CMD description.

CONSOLE

Not yet supported.

<xopts>

Refer to common options xopts description.

REUSE|CONNECT|RECONNECT|RESUME

Not applicable.

(SHELL|S):<shell>

Refer to common options description .

A specific exception to the synchronous execution style of a CLI shell occurs, when multiple execution instances are addressed by the arguments of current call. When the "-b off" option for background a.k.a asynchronous operations is selected, the standard synchronous foreground operation works quite well. The list of arguments is just executed sequentially as expected. The CLI CREATE action sets implicitly the forced usage of a pseudotty. This is the same as using "-z pty,pty".

(STUBMODE|STUB)[:on]

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.

ENUMERATE

Not applicable.

LIST

Almost the same output as common standard, with following changes in semantics.

id The PID of the local SSH termination point, which is the locally executed relay-instance of ctys and is the parent shell of actual running batch/interactive shell. The CLI plugin does not support a cancel or connect|reconnect|reuse action, so the "id" is here a non-functional hint only.

pid PID of current ctys sessions top.

The following values are not applicable: uuid, mac, dsp, cp, sp

-L (LOCALONLY|LO) | (DISPLAYFORWARDING|DF)

Even though a DISPLAY will actually not be used, this parameter also adopts the behavior of the execution to a providing environment, which is frequently for ctys true.

These are the only location parameters to be applied.

7.2.2 RDP

SYNTAX

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

DESCRIPTION

ATTENTION:

This plugin is actually in **alpha-release**, thus might have some drawbacks. The interface and integration of tools is going to be extended, currently **rdesktop** is the only supported client.

This plugin manages within the UnifiedSessionManger RDP sessions particularly to VirtualBox(TM) and Microsof-Windows(TM) based machines. Due to the intention of the RDP protocol this plugin has only a client component and utilizes any server providing access for RDP based client attachment. The RDP plugin encapsulates and handles the complete interaction with the local and remote components provided by means of the client **rdesktop**.

SSH based connections are the only one supported, thus the only actual connections allowed are to 'localhost'. In case of servers not offering the local-only attachment the user should setup additional security measures e.g. by activating packetfilter rules. The 'unmentioned-exception' within the UnifiedSessionsManager is the attachment of consoles to NON-Unix boxes. In that case a network connection is required for the current version(might change soon) and should be used in secure segments only, e.g. by usage of the **ctys-beamer** to inteconnect to a secure gateway by an SSH tunnel and bridging the last hop by native RDP.

The sessions are generally categorized into two basic configurations, the coallocation of the client with the targeted server - **DISPLAYFORWARDING** - and the distributed client and server location on different hosts - **CONNECTION-FORWARDING** . In the latter case an intermediary SSH tunnel is created, which handles the port-forwarding and an has to do a remapping due to ambiguity within the network scope. Thus the local port number and the remote portnumber on the server may frequently differ. The required option to control the forwarding is **-L**.

Additional information containing use-cases with application examples is available from **ctys-uc-RDP**.

OPTIONS

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


CANCEL

CANCEL=(*<machine-address>*) {1,n}
 |ALL|CLIENT|SERVER|BOTH

<machine-address>

For RDP the following parts of a *<machine-address>* are applicable: ID|I, LABEL|L. When the RDP client is used in shared-mode, the address applies to all sharing clients are handled as one logical unit and CANCEL is applied to all at once. The address could be supported with multiple instances.

ALL|BOTH|(CLIENT|SERVER)

All enumerates are handled identical due to a client only applicability.

CREATE

CREATE=[*<machine-address>*]
 [CONNECT|RECONNECT]
 [CONSOLE:*<rdesktop>*]
 [(CALLOPTS|C):*<callopts>*]
 [(REMOTESETPORT|RBASE):*<base-port>*]
 [(REMOTEHOST|RHOST|INSECURE):*<ext-host>*]
 [(REMOTEPORT|RPORT):*<portnumber>*]
 [(XOPTS|X):*<xopts>*]

<machine-address>

For RDP the following parts of a *<machine-address>* are applicable: LABEL|L and port, here REMOTEPORT. When the RDP server is used in shared-mode, the address applies to all sharing clients. The LABEL suboption is here mandatory.

CONNECT

Almost the same as REUSE, but no new server will be started if missing.

CONSOLE

Current version supports **rdesktop**.

REMOTEHOST

The external(!=localhost) host to be accessed by remotely attached RDP client. This is foreseen for blackboxes and access to MS-Windows(TM) based hosts. It should not be forced for UNIX/Linux based boxes.

RECONNECT

Similar to REUSE, with the difference, that any previous active client will be killed before attaching ONE new client. Therefore in shared mode, when multiple clients could simultaneously share one server, all sharing clients are handled as one logical unit and will be thus killed together. Specific exchange of a single client is not supported.

A special may be recognized, when a local client of a **CONNECTION-FORWARDING** is going to be reconnected, the replacement has to be of the same forwarding type. This applies in general, though one machine is accessed for the new client only.

ENUMERATE

Not applicable.

LIST

The same output as common standard.

7.2.3 VNC

SYNTAX

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

DESCRIPTION

This plugin manages VNC sessions to LINUX/UNIX OS. It encapsulates and handles the complete interaction with the local and remote components provided by the VNC variants **RealVNC** and **TightVNC**.

SSH based connections are the only one supported. The sessions are generally categorized into two basic configurations, the coallocated client and server component - **DISPLAYFORWARDING** - and the distributed client and server component - **CONNECTIONFORWARDING** . In the latter case an intermediary SSH tunnel is created. Therefore a vncserver is started and managed on the target server, whereas a vncviewer could be started on the target host or on any client by "Display Forwarding" or "Connection Forwarding". Another feature offers the intermixed usage of VNC, where the vncviewer is connected to a VM, this is the case e.g. for Xen or VMware-WS.

A particular advance is the introduction of a generic addressing schema based on the <machine-address>. This offers the definition of dynamic LABELs as an alias to an arbitrary session. This LABEL is from than on a fully valid address identifier which could be used within the whole ctys toolset. The management of distributed port numbers as well as e.g. the multiplexing of VNC connections into one SSH tunnel is handled by this module.

Additional information containing use-cases with application examples is available from **ctys-uc-VNC**.

OPTIONS

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

CANCEL

```

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

```

<machine-address>

For VNC the following parts of a <machine-address> are applicable: ID|I, LABEL|L. When the VNCviewer/VNCserver is used in the default shared-mode, the address applies to all sharing VNCclients/vncviewer are handled as one logical unit and CANCEL is applied to all at once. The address could be supported with multiple instances.

ALL|BOTH|(CLIENT|SERVER)

ALL and BOTH kill clients and servers on local machine. Remote clients by CONNECTIONFORWARDING might be exiting when server-loss is detected.

The **SERVER** scope is actually for VNC the same as ALL or BOTH, this is due to the default (non-server) behaviour of attached clients, which exit when detecting a server-loss.

The **CLIENT** scope just kills all client processes by means of OS, which is simply calling kill on their PID. The server processes remain untouched.

REBOOT|RESET|INIT|SUSPEND

These methods just behave as a "soft-kill" which is a more or less soft shutdown, for VNC only! Application shutdown is not supported. So in this case first all clients are killed, following a call to "vncserver -kill :<id>" for all matched. No additional action is performed in case of a failure.

POWEROFF

These method could be seen as a "hard-kill" which is a trial to "soft-kill" and an immediate following process kill by means of native OS. Thus there might be definitely no difference to a controlled shutdown of VNC managing unprepared applications.

The session(s) are basically just killed, so the caller is responsible for appropriate handling of contained jobs.

CREATE

```

CREATE=[<machine-address>]
  [REUSE|CONNECT|RECONNECT|RESUME]
  [BULK:[0-9]{1,3}]
  [(CALLOPTS|C):<callopts>]
  [(CMD):<cmd>]
  [(CONFIG):<config-file-for-windows>]
  [PASSWD:(<passwdfile>|-)]
  [(REMOTEBASEPORT|RBASE):<base-port>]
  [(REMOTEHOST|RHOST|INSECURE):<host>]
  [(REMOTEPORT|RPORT):<port>]
  [(SHELL|S):<shell>]
  [(VNCDESKIDLIST|VDIL):<list-of-xstartup-custom-ids>]
  [(WAITC):<delay-after-viewer-call>]
  [(WAITS):<delay-before-viewer-call>]
  [WM:(DTWM|FVWM|FVWM2|GNOME|KDE|TWM|X11|XFCE)]
  [(XOPTS|X):<xopts>]

```

<machine-address>

For VNC the following parts of a <machine-address> are applicable: LABEL|L

When the VNCviewer/VNCserver is used in shared-mode, the address applies to all sharing VNCclients/vncviewers. The LABEL suboption is here mandatory.

BOOTMODE Not applicable.

CONNECT

Almost the same as REUSE, but no new server will be started if missing.

CONSOLE

Not applicable.

PING

Not applicable.

RECONNECT

Similar to REUSE, with the difference, that any previous active client will be killed before attaching ONE new client. Therefore in shared mode, when multiple clients could simultaneously share one server, all sharing clients are handled as one logical unit and will be thus killed together. Specific exchange of a single client is not supported.

RESUME

Not applicable.

REUSE

When a server process with matching ID or LABEL is already running it will be used, else a new one will be started. In case of non-shared-mode operations of VNC any running vncviewer will be killed by disconnecting through the VNCserver. This is almost the same behaviour as for RECONNECT. When running in shared-mode, just an additional vncviewer will be attached to the server.

REMOTEHOST:<literal-host>

Host to be used literally to be connected to. This could be either localhost, which is the default, or a remote host.

PASSWD::(<passwdfile>|-)

Alternate password source when a password is configured. The following suboptions are available:

- <passwfile>
An alternate password file.
- '-' (hyphen)
No password, thus an interactive password challenge is performed.
- SSHPING
Not applicable.
- USER
Not applicable.
- VNCDESKIDLIST
A list of custom IDs, which could be preconfigured desktops and/or desktop-parts within the **\$HOME/.vnc/xstartup** file of VNC. The list defines parts of a pre-configured desktop to be actually started so it is possible to start specific GUI environments. For an realworld example refer to the installed file, either in the installed package
``${CTYS_LIBPATH}/ctys-01_10_013/conf/vnc/xstartup`
or when actually installed in
`$HOME/.vnc/xstartup`.

Various desktops within the VNC session could be pre-configured and utilized call-by-call at runtime.

This option is supported for VNC sessions only, pre-requisite is the execution of the **xstartup** file, which is by now not performed for KVM, QEMU, XEN and VMW-WS sessions. Currently pre-configured values are:

- demo1
- demo2
- demo3
- demo4
- demo5

The separator is the standard separator character '%'.

REMOTEBASEPORT:<base-port>

Base port as new offset for port calculations from the DISPLAY number. Standard value is 5900.

REMOTEPORT:<literal-port>

Port to be used literally, required for several VMs with fixed Server-Ports.

WAITC:<delay-after-viewer-call>

Delay after start of vncviewer, internally used as delay before check of PID for JOBDATA. Might not be really required to be varied, but provided for completeness.

WAITS:<delay-before-viewer-call>

Delay for start of vncviewer, required when the execution is too fast for the VNCserver to finish it's init.

The practical application could be the usage within a GROUP and/or MACRO, where for security reasons a password based access to multiple <exec-targets> is provided, e.g. for root accounts within a admin group. With setting of

this parameter the initial output of VNCviewer is delayed due to it's own delay, thus a series of password requests occur without being poisoned by trace messages of the parallel executed VNCviewer.

WM:<window-manager-enum>

A single window manager to be used for current session. The values are preconfigured for specific distributions and operating systems within the **xstartup** file of VNC. The provided examples could be customized as required. The appropriate software packages are required to be pre-installed before application. Currently pre-configured values are:

- DTWM
- FVWM
- FVWM2
- GNOME
- KDE
- TWM
- X11
- XFCE

BULK:[0-9]1,3

This is a bulk counter for automatic handling of given number of sessions. Mainly used for test purposes. It extends automatically the supported standard <label> with three leading-zero-digits, for each instance. Which could be DEFAULT. The default limiting maximum is set to 20. <bulk> could be used for CREATE only.

ENUMERATE

Not applicable.

LIST

Almost the same output as common standard, with following changes in semantics.

id: The DISPLAY used by the vncviewer and/or vncserver. For the actual display of the server two cases has to be distinguished:

DISPLAYFORWARDING

The DISPLAY of vncviewer and vncserver are identical.

CONNECTIONFORWARDING

The DISPLAY of vncviewer and vncserver are different, this is due to the intermediate tunnel, which handles the port-forwarding and an has to do a remapping due to ambiguity within the network scope. The following values are not applicable:

uuid, mac, tcp

7.2.4 X11

SYNTAX

```
ctys -t X11 -a action[=<suboptions>] ...
ctys -T X11 -a action[=<suboptions>] ...
ctys -T ALL -a action[=<suboptions>] ...
```

DESCRIPTION

The X11-plugin starts a remote shell within a terminal emulation by default. Alternatively any X11 command with an arbitrary shell could be executed. When executing a command, the functionality is almost equivalent to the CLI plugin. The primary application of X11 is to utilize the default behaviour of starting an Xsession within it's own terminal emulation on the desktop. Therefore the pre-configured choices are: XTERM, GTERM, and EMACS, which is started in "shell-mode".

The X11 plugin opens an X terminal window with an interactive bash. Various consoles could be used, which are actually X-terminals such as Xterm ore gnome-terminal.

Due to the different usage of hyphens for the variuos graphical user interfaces the suboptions 'SH' and 'DH' - 'single hyphen' and 'double hyphen' - were introduced. The "SH" suboption is here mandatory for the usage of Xterm, because the Xterm emulation requires a single-hyphen for it's options, default is "DH".

The default behaviour concerning the terminal emulation is to scan for a gnome-terminal first and prefer it if found, else an xterm emulation will be started by default.

It should be mentioned and explained here that there are two quite similar suboptions, which on the first view seems to be redundant, which is actually not the case. The main **difference between** the **CMD** and **SHELL** suboptions is the behaviour. Whereas the SHELL just overlays the standard shell within a Session - here within a defined CONSOLE, the CMD suboption executes a command from within the executed shell - either the standard shell, or a pre-set custom shell.

Additional information containing use-cases with application examples is available from **ctys-uc-X11**.

OPTIONS

-a action[=<suboptions>]

CANCEL

Not applicable.

CREATE

```

CREATE=<machine-address>[, ]
  [
    (CONSOLE: XTERM|GTERM|EMACS|EMACSA)
    |
    (CMD: <cmd>)
  ] [, ]
[(CALLOPTS|C): <callopts>] [, ]
[CD: <working-directory>] [, ]
[(XOPTS|X): <xopts>] [, ]
[(SHELL|S): <shell>] [, ]
[(STUBMODE|STUB) [:on]] [, ]
[(DH|SH)] [, ]
[(TITLEKEY: <key-name-label>|NOTITLE)]

```

<machine-address>

The LABEL suboption is here the only supported and mandatory part.

<callopts>

Refer to common options parts description.

CD:<working-directory>

Change current working directory for the started command. This provides for example the preconfiguration of specific work-sets, where each terminal could be opened for a specific directory.

CMD:<cmd>

Refer to common options parts description. The CMD could be provided alternatively to a fixed CONSOLE.

CONSOLE:(XTERM|GTERM|EMACS|EMACSA)

A fixed type of a CONSOLE, following types are supported.

CONSOLE:GTERM Starts the "gnome-terminal".

CONSOLE:XTERM Starts the Xterm.

CONSOLE:EMACS Starts an emacs and opens a "shell" buffer.

CONSOLE:EMACSA Starts an emacs and opens an "ansi-term" buffer.

(DH|SH)

Sets a double-hyphen or a single-hyphen for call arguments. Where the single hyphen e.g. is required for old style X11 utility options, the double-hyphen for new options style of Gnome.

NOTITLE

Suppresses the generation of title either from LABEL, or from default. Be aware, that dynamic sessions like X11 without an LABEL visible by ps, are not - or just limited - recognized by LIST action. Thus the only reliable support for sessions started with this flag is the start itself.

SHELL|S:<shell>

Refer to common options parts description.

(STUBMODE|STUB)[:on]

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.

TITLEKEY:<title-name-key>

Alters the options keyword to be used to set the title of an Xwindow, which

is set by default to "title". Some older applications, like Xclock support different, e.g. "name" as options keyword only. Due to the crucial role of the window title, which is the LABEL, this approach should be preferred when "title" is not supported. Do use NOTITLE only if definitely unavoidable.

<xopts>

Refer to common options parts description.

- A specific exception to the synchronous character of an X11 shell occurs, when multiple execution instances are addressed by the arguments of current call. When the "-b off" option for background a.k.a asynchronous operations is selected, the standard synchronous foreground operation works quite well. The list of arguments is just executed sequentially as expected.

In current implementation the user is responsible for handling the appropriate values, which are assigned by default. The enforcement of resetting user defined values could be somewhat tricky due to permutation of bulk arguments, thus is shifted because of priorities. Resulting of this, the actual environment is a superposition of all previous executed target options with the global options.

ENUMERATE

Not applicable to X11.

LIST

Almost the same output as common standard, with following changes in semantics.

id The PID of the first local call below of SSH termination point, which is the locally executed relay-instance of ctys and is the parent shell of actual running batch/interactive shell. Thus the topmost ctys-call.

pid PID of current ctys sessions top. The following values are not applicable: uuid, mac, dsp, cp, sp

-L (LOCALONLY|LO) | (DISPLAYFORWARDING|DF)

The location provides **DISPLAYFORWARDING** - and **CONNECTIONFORWARDING** . **CONNECTIONFORWARDING**. In case of **CONNECTIONFORWARDING** the terminal is executed locally, whereas a remotesession is opened from within the contained shell. In case of additional STUBMODE the shell is a lean SSH login only.

7.3 Category PMs

This section contains help for the add-on plugins of category PMs, which are executed on native physical machines.

7.3.1 ctys-PM

SYNTAX

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

DESCRIPTION

The **PM** plugin adds support for sessions to **PhysicalMachines** with plugins of type **HOSTs** for console access.

The management of the physical machines include particularly the support of boot and shutdown for local and remote machines. The supported interfaces comprise the local system interfaces, "Wake-on-LAN/**WoL**" and "Intelligent Platform Management Interface/**IPMI**". The remote check of the Health-Status of the physical machines is based on the "**lm_sensors**" package.

The PM plugin is "**VM-STACK safe**" as it handles running virtual machines by propagating state change events(e.g. SHUTDOWN) of the physical platform to contained hypervisors and emulators.

Additional information containing use-cases with application examples is available from **ctys-uc-PM.** , **ctys-uc-IPMI.** , and **ctys-uc-WoL.** .

OPTIONS

-a action[=*<suboptions>*]

CANCEL

```
CANCEL=(<machine-address>){1}
(
  [FORCE|STACK]
  [SELF]
  [
    RESET|REBOOT
    |(INIT:<init-state>)
    |(PAUSE|S3)|(SUSPEND|S4)
    |(POWEROFF|S5)[:<timeoutBeforeKillVM>]
  ]
  [TIMEOUT:<timeout-value>]
  [WOL]
  [IF:<if-for-wol>]
)
```

SELF

Self cancels the hosting machine for the call after cancel operations on the stack, if not provided the stack is canceled only. This is due to conformity with the CANCEL operations of VMs, where the hosting system for the hypervisor is addressed. This suboption basically should be applied as default and only behaviour for a PM. But it seems to be somewhat smart, just to target anything within a PM, e.g. when bringing the PM into maintenance mode, without changing the operational state of the PM itself. Due to the main targeted user as a systems administrator, the simplicity of the UI was in this case below the advantage. So sadly accepting some irritation with this suboption here, making it "not default". When SELF is required, it has to be provided.

WOL

WOL flag is supported for NIC drivers supporting the ethtool interface, which are e.g. e1000, tg3 or 8139too, e100. For cards with configuration by modprobe-interface this flag is not applicable.

WOL enables the later call of "CREATE=WOL,..." by setting the volatile registers of the NIC via the call of "ethtool -s *<ifX>* wol g" thus WoL via MagicPacket(tm) is enabled after "halt -p" call.

For successful execution root permissions for operations on the interface by "ethtool" is required and should be configured via sudoers or .k5users.

This state change request is not propagated to the VM-stack, the PM plugin itself is the only one evaluating it.

When the WoL feature is set persistently by means of the OS, the WOL flag is not required.

IF:*<if-for-wol>*

The interface to be prepared for WoL. Access permission for "ethtool" is required. If this suboption is not provided, then the first detected valid interface is used. Therefore each plugin is called by the cty specific init-mechanism in order to prepare the shutdown, though the prepare the interfaces too.

This is particularly for bridged networks required, where the bridged interfaces not propagate the "wol g" flag properly in each case. Thus the

detection of the valid interfaces is performed after the completed plugin shutdown actions.

CREATE

```
CREATE=<machine-address>
[, (REUSE|CONNECT|RECONNECT|RESUME)]
[, WOL[:<wol-delay>]
[
(
, BROADCAST:<broadcast-dest-address>
[, PORT:<broadcast-dest-port>]
[, BMAC:<broadcast-MAC-address>]
)
|
(IF:<broadcast-interface>)
]
[, CONSOLE: (CLI|GTERM|XTERM|EMACS|VNC)]
[, USER:<user>]
[, PING: (OFF|<#repetition>%<sleep>)]
[, SSHPING: (OFF|<#repetition>%<sleep>)]
```

For information on USER , PING , and SSHPING refer to common options.

<machine-address>

The address of the machine to be activated. The usage of <machine-address> implies an exact target to be addressed explicitly.

For multihomed machines a minor limit occurs, when the NIC to be used for WoL is different from the NIC for usage of CONSOLE access. Currently CONSOLE suboption is applied to the target <machine-address>, which is also used as target for the WoL activation. This can cause difficulties, e.g. when a board is used, where - due to BIOS limits - a low rate NIC has to be used for WoL activation, whereas the normal traffic is performed by the remaining NICs with channel bonding. The same principle could be applied due to security reasons when multiple NICs are available. Anyhow, the workaround is to use two calls, one for the WOL packet without a CONSOLE, and a second one for the CONSOLE only, addressing the access port.

BMAC:<broadcast-MAC-address>

The MAC address to be inserted into the broadcast packet, which is defined to be the MAC address of the receiving NIC. In difference to the <machine-address>, which is strongly checked for consistency, the broadcast target is not checked for consistency. This is due to various remote application cases, where the actual data might not be locally available, or a relay with port-forwarding and/or protocol-forwarding might be involved.

BROADCAST:<broadcast-address>

For machines not within the callers subnet, an arbitrary broadcast address could be set. A UDP package containing a **MagicPacket(TM)** is sent to the given address with port "DISCARD=9" set by default. This parameter forces the usage of native script for generating and sending of the WoL packet and cannot be combined with the "IF" suboption. An example is

provided within the following chapter containing additional examples.

The Address has to be of one of the following types:

An arbitrary defined address with an optional port:<ip-address>

This will require the final router for the target subnet to redirect the packet from the destined address to a local broadcast message on the subnet. Thus some additional configuration on the router is required.

A directed broadcast address, where all subnet bits are set. The user has to be aware of the netmask for the target subnet.

CONNECT

Opens just a new CONSOLE, requires therefore the PM in operational state.

CONSOLE:(CLI|XTERM|GTERM|EMACS|VNC)

If given a console will be opened on the site of caller. This will be performed in addition to the standard console, which may be attached to the KVM-connectors.

Default is CLI, which is based on a complete subcall with type CLI.

The "-b" option will be set for the types of consoles as:

```
CLI    => "-b 0"
XTERM  => "-b 1"
GTERM  => "-b 1"
EMACS  => "-b 1"
VNC    => "-b 1"
```

IF:<broadcast-interface>

This defines the local interface where an ethernet broadcast is to be sent. Therefore the tool "ether-wake" is utilized.

This is the default behaviour when neither the BROADCAST nor the IF parameter is supported.

For the default case the first interface will be evaluated from a "ifconfig" call, which depends on the current OS and possible custom configuration.

This parameter cannot be combined with BROADCAST.

PORT:<broadcast-dest-port>

A port to be used on the remote site. A UDP package containing a **MagicPacket(TM)** is sent to the given BROADCAST address with port "DISCARD=9" set by default.

RECONNECT

All current CONSOLES of the user are CANCELED, and a new one is started. The CANCEL just ignores any child process of the enumerated CONSOLES, thus the user is responsible for proper shutdown of running subtasks. Specific exchange of a single client is not supported.

When PM is yet running, it will be rebooted, else just booted and connected with the chosen CONSOLE.

RESUME

In this version the same as create only.

REUSE

When a machine is already active just a connect is performed. In case of a required new session, first the machine is booted, if an appropriate activation method is defined, which is WoL only in this version. When the machine is available, a CONSOLE is opened is requested.

WOL[:<wol-delay>]

The addressed PM will be activated by Wake-On-LAN, which is in current version the only supported method. Therefore the default "ether-wakeup" tool is utilized by default. This could be changed by setting the environment variable `CONSOLE_WOL`, to which the MAC-address of the target will be appended. The current implementation expects all PMs within one subnet. The timer `<wol-delay>` is the time period to be delayed the execution after sending the WoL packet. This timer is waited for once before starting the periodical poll with shorter timeout.

ENUMERATE

This is specific for PM, just the local configuration information is displayed.

LIST

This is specific for PM, just the local configuration information is listed.

7.4 Category VMs

This section contains help for the add-on plugins of category VMs, which are executed on OS level in order to start and operate a specific type of VM.

7.4.1 ctys-QEMU

SYNTAX

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

DESCRIPTION

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

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

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

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

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

OPTIONS

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

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

-a CANCEL

Specific semantics for `SPORT` within `QEMUMONSOCK` as described before.

The `CONSOLE` type `CLI0` requires specific handling for `CANCEL`.

-a CREATE

All standard parameters not listed here could be applied.

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

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

BOOTMODE:<MODE>[%<PATH>]

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

- **sh+ctys**

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

- **ctys-conf**

The same as `ctys`, but the configuration keys are **sourced** from an external `conf-file`, which is required to be coallocated within the same directory. These are created by the **ctys-cresteConfVM** utility.

The optional `<PATH>` parameter provides the temporary alteration of pre-configured boot source media.

BOOTMODE:CD

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

BOOTMODE:DVD

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

BOOTMODE:FDD

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

BOOTMODE:HDD

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

BOOTMODE:PXE

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

BOOTMODE:USB

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

BOOTMODE:VHDD

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

CONSOLE

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

CONSOLE:CLI

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

CONSOLE:CLI0

This mode deviates from the common CLI mode, and is tightly coupled to the VM, thus could not be detached. When the console is detached, the VM will be terminated. When the GuestOS is shutdown in CLI0 mode the console stays still occupied by the QEMU VM after the guest system is halted. In order to release the CONSOLE/Terminal, the monitor has to be used. Call **Ctrl-A-c-
<RETURN>**, and - when the (**qemu**) monitor prompt occurs - execute **quit** within the monitor.

This mode sets particularly the following options and operational modifications.

- **trap: INT,TSTP, QUIT:** These signals are deactivated in the first instance of the called client, and in the entry instance of the remote client(s). Thus the signals, if activated, are transparently passed though to the target peer. The values could be configured by the variable and/or set by the option "-S" .

CTYS_SIGIGNORESPEC.

- **-b 0,2 -z 2:** This mode sets implicitly **-b 0,2 -z 2**, otherwise the the input stream might be disconnected. The background mode is generally not applicable to CLI0.

CONSOLE:EMACS

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

CONSOLE:EMACSA

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

CONSOLE:EMACSAM

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

CONSOLE:GTERM

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

CONSOLE:SDL

This is the standard graphical console of QEMU.

CONSOLE:XTERM

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

CONSOLE:VNC

The VNC console of QEMU. The VNC console access port is implicitly pre-

pared additionally for the following `CONSOLE` types for later attachment: `CLI`, `XTERM`, `GTERM`, `EMACSM`, `EMACS`, `EMACSAM`, and `EMACSA`.

INSTMODE[:<params>)]

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

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

<MODE-SRC>

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

<SRC-PATH>

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

<MODE-TARGET>

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

<TARGET-PATH>

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

(INIT|<custom>)

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

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

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

-a LIST

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

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

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

-g <geometry>|<geometryExtended>

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

CLI

SDL

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

XTERM|GTERM

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

VNC

As expected.

-r <resolution>

Not supported.

PREREQUISITES

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

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

Supported products**VDE2-Installation**

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

QEMU-Installation

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

KVM-Installation

KVM should be installed by standard distribution.

7.4.2 ctys-VBOX

SYNTAX

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

DESCRIPTION

ATTENTION:

Current version of **VBOX-Plugin** is an **ALPHA RELEASE** with partial function, refer to releasenotes. Contained functions comprise CREATE, CANCEL, LIST, ENUMERATE, INFO, and SHOW. Some toolset functions may be missing for now, the following description is defined as a draft for the target version. Current version may not yet contain e.g. the automated inventory registration and the autonomous installer tools.

The **VBOX** plugin adds support for sessions to **VirtualMachines** of type **Virtual-Box - Oracle(TM)**.

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

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

The supported product versions are automatically detected, thus no further parameters for version distinction are required.

For now in case of usage of **ctys-uc-RDP**. protocol the access permissions have to be set before by the user. This could be either performed from the standard GUI of the VirtualBox command, or by e.g. VBoxManage with the following call:

- VBoxManage controlvm vrdp=on

Optionally for required shared mode:

- VBoxManage modifyvm vrdpmulticon=on

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

OPTIONS

-a action[=<suboptions>]

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

CANCEL

The client server communications is handled basically by two different types of client server communications. Following applies to the generic framework interface:

- Tightly coupled synchronous stateful communications, where the cancellation of the client terminates the server too.
- Loosely coupled asynchronous stateless communications, where the server exists independently from the client. Thus a cancel of the client(s) does not influence the existence of the server processes. This mode could be applied in the so called 'headless' mode.

Particularly the wildcard-attribute **ALL** is provided for application on all sessions of selected and permissive user-scope.

CREATE

This starts a VM, where following could be applied.

CONSOLE

All available console types for the supported versions could be used. An overview of provided CONSOLE types is listed within the description of the standard call for CREATE. Particularly the following console types are available for VBOX:

VBOX

The standard frontend, but is of type synchronous, where the states of client and server as tightly coupled. This could be used in DISPLAY-FORWARDING mode only.

VBOXSDL

Similar to the standard frontend, is of type synchronous too, where the states of client and server as tightly coupled. This could be used in DISPLAYFORWARDING mode only.

RDP

This is functionally similar to the VNC type, where the access is loosely coupled by an stateless asynchronous GUI protocol. Thus this type could be arbitrarily attached and detached.

Current version supports the **rdesktop** client only, which is detected automatically, the order of scan for the location of the binaries could be varied by configuration files.

NONE

The initially headless mode.

ENUMERATE

Enumerates all VBOX sessions, therefore first the created ctys-files are searched, which are required by convention to be coallocated with the vdi-file of the virtual system storage. When missing the vdi-files are scanned and for any found the information is accomplished by call of VBoxManage. A prerequisite is here the previous registration of the VM, which could be automated by the ctys scanners.

The current version requires for several functions a valid installation of VirtualBox(TM) on the executing machine. When no installation is found only very basic information is collected and the 'VMSTATE' is set to 'DISABLED'. This is due to the requirement of VBoxManage as the interface for transformation and mapping of the label/name and UUID of the VMs. This is also utilized for the match-filters within the automated scanners.

Once the VM is detected the second optional requirement is the pre-registration of the VMs due to the required analysis of some virtual hardware parameters. These comprise for example the architecture, the virtual RAM, the number of emulated CPUs, and more. When these parameters are not detected they are just ignored and an empty field is added to the machine record.

The next requirement is the access permission to the information, which first requires access to the containing directory and files.

REMARK: This behaviour may be changed and expanded within the next releases.

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

Currently the following selections are supported:

Product/Version	LO	CF	DF	SO
VBOX		no	yes	no
VBOXSDL		no	yes	no
RDP		yes	yes	yes

Forwarding modes and call locations for VBOX versions

PREREQUISITES**Supported products:**

Product	Tested Versions
VirtualBox	>3.x (3.1.2)
rdesktop	>1.x (1.6.0)

Supported products

7.4.3 ctys-VMW

SYNTAX

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

DESCRIPTION

The **VMW** plugin adds support for sessions to **VirtualMachines** of type **VMware(TM)**.

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

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

The scope comprises for now the products VMware-Workstation, VMware-Server, and VMware-Player, which are all TradeMarks of VMware Inc. The supported product versions are automatically detected, thus no further parameters for version distinction are required. Current version supports the **Server(1.x, 2.x)** and **Player(1.x, 2.x, 3.x)** and the **Workstation(6.x, 7.x)** variants.

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

OPTIONS

-a action[=<suboptions>]

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

CANCEL

Following applies to the generic framework interface:

CLIENT

The client server communications is handled by a proprietary frontend, which implements the sharing of a single port, by default 904. This requires some specific treatment particularly for the case of CONNECTIONFORWARDING.

The headless-mode(NONE) for CONSOLE is not supported as initial call, the console could be detached without shutting down the server component later, when background-mode is preconfigured.

This is not applicable for VMplayer.

SERVER

This is not applicable for VMplayer.

BOTH

When this keyword is present the server and client processes of the previous set are filtered.

Particularly the wildcard-attribute **ALL** is provided for application on all sessions of selected and permissive user-scope.

CREATE

VM or Team to be loaded and options passed through. For additional information refer to the related documentation from VMware. E.g. "Workstation User's Manual " Workstation 6.0 Appendix A "Workstation Command-Line Reference"; pg. 403.

<callopts>

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

```
vmware <callopts> <vmx-path>
```

The "-" double hyphen is inserted as required.

Be aware, that some of these such as "-geometry" and "-name" are already implicitly utilized by other options, thus use this if, than CAREFULLY.

CONSOLE

All available console types for the supported VMware versions could be used, but the specific support of the versions may vary. An overview of provided CONSOLE types is listed within the description of the standard call for CREATE.

Particularly the following console types are available for VMW:

VMW

The standard frontend, is available for all. When configured within the VM a detach and re-attach to the running VM is possible. The background mode is required here, which is not available for at least some versions of the Player. For the newer versions beginning with Server/Player-2.x the browser interface is opened.

VMWRC

This is the remote console standalone starter for **vmware-vmrc**, which is available within the Server package. For additional description refer to the ctys-HOWTO/Examples. The login requires proper authentication, thus the USER-parameter should be used in companion. Otherwise a login dialogue occurs.

The following paths are probed for the executable in given order for a custom installed copy.

- (a) \$MYADDONSPATH/vmware-rc-x64

- (b) \$HOME/vmware/vmware-rc-x64
- (c) /opt/vmware/vmware-rc-x64
- (d) /usr/bin
- (e) /usr/local/bin
- (f) /usr/share/vmware-rc
- (g) /usr/share/vmware-rc-x64
- (h) /usr/share/vmware-rc-x86
- (i) /opt/bin
- (j) /opt/vmware/vmware-rc

FIREFOX

This is the starter for the remote console as standard plugin.

NONE

The initial headless mode, available for Server and Workstation, requires some specific configuration of the VM.

VNC

The use of VNCviewer as frontend, available for Workstation-Version-6, requires configuration of VM. The port has to be configured statically within the VM, but is evaluated dynamically by mapping of the `<machine-address>` at runtime.

ENUMERATE

Enumerates all VMW sessions, therefore the vmx-files will be scanned and the matched attributes displayed. Therefore the following order of files will be scanned for values, in each case the searched key is expected to have the prefix `#@#` within the file.

`<pname>`

The standard configuration file for VM, as given.

`<pname-prefix>.ctys`

The prefix of given filename with the `.ctys` suffix.

`<pname-dirname>.ctys`

The dirname of given file with `.ctys` suffix.

`-g <geometry>|<geometryExtended>`

The geometry has a slightly different behaviour to the standard when specific options of proprietary WS-CONSOLE are switched on.

- The positioning parts of parameters of `<geometryExtended>` seem to work in any case correctly.
- The offset of `<geometry>` seems to work proper in any case too.
- The size of `<geometry>` seems to work proper when **AutofitWindow** and **AutofitGuest** are switched off.

— As could be seen, shortly after start of CONSOLE it will resize itself, if previous parameters are set. Which is indeed a pretty well behaviour. What else should that options of CONSOLE control?

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

Currently the following selections are supported:

Product/Version	LO	CF	DF	SO
VMware-WS		no	yes	1)
VMware-Server		yes	yes	1)
VMware-Player				

Forwarding modes and call locations for VMW versions

1)The background-server-mode is currently implicitly supported only.

This requires the options to be selected within the products - which are slightly different, but are almost commonly supported - and will be implicitly started only when starting the whole product.

The server component continues execution when the client is canceled, but could not be started separately. A CONNECT to a running server is supported.

PREREQUISITES

Supported products:

Product	Tested Versions
VMware-WS	6.x, 7.0.1
VMware-Server	1.0.3-1.0.10, 2.0.2
VMware-Player	1.0.5-1.0.6, 2.5.1, 3.0.1

Supported products

background operations

The background operations for the server component of the VM has to be set explicitly for some variants. This is not required for server-products, but should be done for workstation products.

Not setting this leads to an immediate termination of the server, when client closes. Not necessarily with a soft-shutdown!!!

tabs-mode

The tabs-mode for the proprietary CONSOLE should be set off. Even though a tabbed-view could be used too.

Due to the embedded dispatcher for the display the CONSOLE requires here a user interaction for selecting the target display in any case(by tabs, or by menu "tabs"), but when only one display per window is assigned it appears to be little more straight-forward.

authentication

User authentication is for **-P <port>** access even for **-h localhost** required, so for CONNECTIONFORWARDING in any case the user seems to have to perform a login. In addition, it seems that the user has to be a local user on that machine.

As far as I can say for now, only in case of DISPLAYFORWARDING SSO has an effect on sessions. Let me please know, when this is wrong, and don't forget to explain how.

Headless-Start

Found a "nogui" param for vmrun, will be introduced asap.

7.4.4 ctys-XEN

SYNTAX

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

DESCRIPTION

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

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

The scope comprises for now the products XEN community edition, which are all TradeMarks of Citrix Inc. The supported product versions are automatically detected, thus no further parameters for version distinction are required. Current version supports the server versions 3.x. Updates will be available soon.

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

OPTIONS

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

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

CANCEL

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

CREATE

All standard parameters not listed here could be applied.

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

CONSOLE

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

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

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

Applicable forwarding modes and call locations for XEN

X) Supported.

A.) Supported, but will block the call-terminal for the whole session, so might not be used from a single-console environment.

B.) Blocks the console for other calls, thus allows for bulk targets serial execution only.

M.) Mandatory, but could be suppressed with "-z NOPTY" when a terminal with some drawbacks is sufficient. One of specific than is that the password will be echoed for some systems in cleartext, anyhow as a lonely night-rider it might not hurt you.

Types of CONSOLE to be applied depends on the **-b** parameter for background execution too.

The following behaviour applies:

"-b 0" - synchronous foreground execution

In this mode the current execution thread is performed synchronous in the foreground, this means particularly a CLI based console cannot be detached, when multiple tasks are in the queue in order to begin the next. Thus it would result to blocking the remaining sessions until the current has been finished by the caller.

This parameter is allowed to be applied, but the caller has to be aware of the drawbacks, when choosing multiple execution targets.

"-b 1" - synchronous background execution

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

CONSOLE:CLI

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

CONSOLE:GTERM

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

CONSOLE:XTERM

Starts first an xterminal by using the X11 module and initiates the startup of the DomU within the Xterminal session as a native and synchronous call to **xm -c ...**. So it is basically the asynchronous variant of a CLI call.

CONSOLE:EMACS

The **Emacs** is started in shell-mode, this supports the full scope of edit features on the output buffer. The basic principle is similar to any X11 console with an embedded CLI interface.

CONSOLE:VNC

This case is somewhat different to the previous, in the way that two independent calls for the DomU itself are required.

- The DomU has to be started, which is performed by calling **xm <conf>**.
- The VNCviewer has to be attached to the offered port by the DomU. Therefore a timeout will be applied, which could be controlled by the environment variable **XEN_CONSOLE_DOMU_INIT_WAIT**, which is used for a sleep-call. Due to buffer handling some console messages could probably be lost.
The client call is an internal call of the VNC plugin, which is independent and could be applied separately.

CONSOLE:NONE

No console is started, any type could be connected later. "-z 2" - force allocation of a pty by ssh Allocates a pty.

<callopts>

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

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

For additional information refer to Xen manual.

-g <geometry>|<geometryExtended>

The geometry has no effect on the server started within the DomU. Just the client will be set:

CLI

Not applicable.

XTERM|GTERM

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

VNC

As expected.

-L <execution-location>

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

-r <resolution>

Not supported.

<xopts>

Refer to common options parts description.

NOTES

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

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

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

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

Another specifics is a legacy of ctys, which is the definition of ID as a static unique identifier for a VM and PM entity, which does not change, when the entity changes it's state to offline. Resulting of this, the ID is for **VMW**, **PM**, and **XEN** the fully qualified pathname of the configuration file, which is not necessarily unique,

due to NFS mounted shares on multiple PMs and/or VMs. This is still not unique, when combining the PMs hostname and the pathname of the configuration file, because the contained IDs, e.g. TCP/IP address, MAC address, and UUID are now available within multiple entities, and thus will be listed as though when using administrative management utilities. Anyhow, it should be at least guaranteed by the user, that the entities are unique within the scope a single node. The toolset is prepared to handle various constellations, but it makes the selection by the user easier.

For this the following shortcuts and conventions apply.

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

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

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

- **NO SPACES** within ANY entry are supported.
- When multiple LAN interfaces are configured, the MAC-addresses are indexed by their actual order-increment, beginning from 0. These are permuted with any provided TCP address of the same index. E.g. MAC0 => eth0 => IP0=10.1.1.1, IP1=11.1.1.1.
- Due to the variety of consoles - CLIENTS - which could be attached and are not simply correlated, the LIST action only displays the SERVER components, which are Dom0/DomU, the clients has to be listed by an extra call to CLI, X11, and/or VNC.
- The execution of the creation by **xm** and some **virsh** access permissions has to be activated and required to be with root permissions. Therefore the configuration file **/etc/sudoers** and/or **/root/.k5users** has to be configured. The access privileges by **sudo** and **ksu -e** will be checked and set appropriately. The variable **XENCALL** and **VIRSHCALL** could be preconfigured.
- The execution of XEN requires in any case the VNC module.
- The version supported by XEN is the 3.x version. The tested and verified version is Xen-3.0.3 of the CentOS-5.0 distribution, even though any 3.x version might work. The version evaluation is done by usage of rpm or xm or virsh or xmtrace. The installation paths are evaluated by which call and should be prepared for execution by PATH.

- Due to the warning-output of some tools, this is fetched as ctys WARNING, which could be fully activated by **-d** option.
- Particularly the **D_SYS** debug-level, which traces all system calls, might be helpful for tracing permission settings.
 - The XEN plugin is stack-aware, though prepared to propagate CREATE and CANCEL actions, same for LIST.
- **XEN_CONSOLE_DOMU_INIT_WAIT**
This variable contains the sleep value after **xm create ...** and before calling a **gnome-terminal** or **xterm**. Therefore in case of a machine which has difficulties due it's performance the value could be adjusted. The current value of 8seconds seems to be safe for initialization of created DomU.

7.5 Category GENERIC

The standard generic plugins of category GENERIC are the actual subdispatcher for calling the specific implementation. Thus these just use an internal data structure containing the list of active and selected plugins and all their related specific functions.

The available sub-dispatcher are:

- (a) LIST
- (b) ENUMERATE
- (c) SHOW
- (d) INFO

7.6 Category Internal

7.6.1 DIGGER - Forwarding Encrypted Connections

Encrypted Tunnel Support

The CORE plugin DIGGER is responsible for handling of connections and the assembly of appropriate prefixes for each SSH-execution. Therefore it could be seen as the manager for remote connections and their encryption. The DIGGER also contains the functions for the management of encrypted port-forwarding.

In case of local calls with the same user-id as the caller only some PATH extensions and a "cd \$HOME" are prefixed, while for remote calls two additional cases has to be handled.

DISPLAYFORWARDING

In case of DISPLAYFORWARDING an ordinary "ssh" prefix with appropriate options is generated. No specific communication and link resources are required, anything is handled by SSH software component after the execution call.

The call is generally mapped to "\$USER@localhost" when no different user is requested. Thus a call with the "\$USER@localhost" or without <execution-target> arguments is handled by the internal call processor identical.

CONNECTIONFORWARDING

In case of CONNECTIONFORWARDING some additional link establishment effort and communications resources are required.

In this case the final communications resources include a client process on the caller's machine, a server process on the server machine, which could be of type VM or PM. In addition a link resource, the port forwarding tunnel is established. This tunnel is no longer managed by ctys once established, but is created as a ONE-SHOT connection, which terminated and releases it's resources once a communications peer terminates.

Even though it is an implicit internal CORE functionality, the resources should be visible within LIST action with their located local and remote ports, including the user id of the owner. Therefore the DIGGER plugin supports a LIST component, which integrates as a minor sized system plugin into the "LIST Plugin-Collector".

LIST Tunnels

One additional feature of DIGGER plugin is the support for LIST action of current active tunnels. These are listed within the standard LIST output and marked as "SSH(<creator-plugin>)" with the client-server flag "T".

The ID of a tunnel is assembled from the local port and the remote port as "<local-port>-<remote-port>". Thus supporting some "speaking" information.

The tunnel is visible on the client machine only.

Chapter 8

Support Tools

8.1 ctys-attribute

SYNTAX

ctys-attribute

```

[--attribute-create[=<position>]]
[--attribute-delete[=<occurance>]]
[--attribute-replace[=<occurance>]]

(
  [
    --attribute-keyonly
    --attribute-name=<attr-name>]
  |
  (
    [
      --attribute-name=<attr-name>
      --attribute-value-new=<attr-val-list>]
    |
    [
      --attribute-name=<attr-name>
      --attribute-value-new=<attr-val-list>
      --attribute-value-old=<attr-val-list>]
    ]
  )
)

[-d <debug-level>]
[-h]
[-H <help-options>]
[--list]
[-t <session-type>]
[-V]
[-X]
<file-name-list>

<attr-val-list>:=<attr-val>[%<attr-val-list>]

<file-name-list>:=<file-name>[ <file-name-list>]

<occurance>:=(
    first|last|all
    #nr
)
#nr=1-9999999

```

```

                                default:=first

<position>:=(
                                top|bottom
                                #line
                                )
                                #line=1-9999999

                                default:=top

```

DESCRIPTION

ATTENTION: This document is under development, therefore just a temporary pre-release, which may change in future versions. The software is alpha-release.

ctys-attribute provides the management of runtime attributes for dynamic and persistent plugins, VMs, and PMs. The attributes of the persistent data are utilized for the handling of machine information for selection, inventory management, and execution.

The current version supports **offline-modification** only. The **cacheDB** has to be updated either by new automated scan or by manual modification. Both may be extended in new versions.

The file is either scanned for specific attributes, or just a new attribute value assignment is inserted. Therefore various types of attribute syntaxes from the various supported namespaces are recognized. The following namespaces are currently supported:

Namespace	Syntax	VM-Config	Applicable Tools
ctys-configuration	ctys	'#@#'-prefixes	ctys-attribute, ctys-groups, ctys-macros, ctys-scripts, Text-Editor
QEMU/KVM	ctys	'#@#'-prefixes	ctys-attribute, Text-Editor
shell	bash	variables	ctys-attribute, Text-Editor
VirtualBox conf-files+repository 1.)	VBoxManage,XML	call parameters	ctys-attribute, Text-Editor, VBoxManage, VirtualBox-GUI
VMware conf-files+repository 2.)	vmx	keywords	ctys-attribute, Text-Editor, Vmware-GUI
Xen conf-files	Python	variables	ctys-attribute, Text-Editor

Table 8.1: Supported Attribute Namespaces

- 1.) Not all keywords are supported yet.
Native OEM tools are required in addition.
- 2.) The repository is not supported yet.
Native OEM tools are required in addition.

OPTIONS

- attribute-create=<position>**
Create a new entry, this is done without recognition of presence. For sequential scan the position could be provided for order-priority.
- attribute-delete=<occurrence>**
Delete an entry, for sequential scan the position could be provided for order-priority.
- attribute-keyonly**
Creates a key only, an attribute without an assigned value.
- attribute-name=<attr-name>**
The name of the attribute to be modified.
- attribute-value-new=<attr-val-list>**
The list of new attributes to be set.
- attribute-value-old=<attr-val-list>**
The list of old attributes to be replaced. Missing attributes are ignored.
- d <debug-level>**
Same as ctys.
- directory=<directory>**
Directory for storage of created configuration files.
- h|-help|-help**
Print help, refer to "-H" for additional information.
- H <help-option>**
The extended help option is based on system interfaces for display of manpages, PDF and HTML documents. This comprises the man pages and installed manuals.

For additional help refer to the documents or type **ctys -H help**.
- list**
The list of present attributes.
- t <session-type>**
The session type the VM has to be created for. Currently **QEMU** (for QEMU and KVM on x86), **VMW** , **VBOX** , and **XEN** are supported. Each defines a common set of parameters with additional specific variations.
- V**
Version.
- X**
Terse.

ARGUMENTS

<file-name-list> : List of files to be modified.

EXIT-VALUES

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

8.2 ctys-beamer

SYNTAX

ctys-beamer

```

[-b <background-options>]
[--ctys-predetach-holdtime=<timeout-secs>]
[-d <debug-options>]
[--display-only]
[--getfreeport}
[-h]
[-H <help-options>]
[-L <remote-login>]
[--mode=(
    (CTYS-HOPS|CH|0)
    |(SSH-CHAIN|SC|1)
    )
]
[-R <remote-host-list>]
[--ssh-hop-holdtime=<holdtime>]
[--ssh-tunnel-holdtime=<holdtime>]
[-V]
[--x11]
[-X]
[<bypassed-ctys-options>]
(--|--beam-this)
<remote-command>

<remote-host-list>:=<relay-chain>[,<remote-host-list>]

<relay-chain>:=<relay-host>[%<relay-chain>]

<relay-host>:=[<user>@](<host>|<access-point>)

<access-point>:=<physical-access-point>|<virtual-access-point>

<physical-access-point>:=<machine-address>
<virtual-access-point>:=<machine-address>

Evaluated options of bypassed standard set:
  <bypassed-ctys-options>:=[
    -b (0|1|2|3)[,(0|1|2|3)]
  ]

```

DESCRIPTION

UNDER DEVELOPMENT: The current interface may change and the functionality will be expanded. For now basic features for call-transformation with some advanced routing and call distribution are included.

ctys-beamer is the main interface for the transformation of a local call to a remote host for execution . Therefore an encrypted tunnel with an arbitrary number of intermediate hops on the route to remote host could be assigned. The defined route is hereby independent from the so called routing of the TCP/IP internet layer, enabling routing independent and encrypted connections for multi-level firewalls. The **ctys-beamer** also supports the definition of multiple targets for remote execution of the same command.

The particular advance here is the full integration into the common name binding and addressing schema of the UnifiedSessionsManager. This provides for extended address notation as well as for full scope of GROUP objects and MACROS.

The **call-transformation** has to be distinguished from the **remote-execution** option of a command, which is semantically slightly different. The **call-transformation** is a call-neutral pure passive transformation for actual first-time execution on the remote site. This could be used particularly for security reasons, when safe and hidden bypassing routes has to be defined for administration purposes.

The distinction to the **remote-execution** of an executable is the active remote operation of application parts due to defined semantics of the call itself. Thus application specific knowledge for the current remote-execution context could be applied, this is e.g. the case for the combination of remotely collected data into a single list. The **call-transformation** as an independent and call neutral dispatcher might fail in general for cases where the combination of the results from various execution sites into one overall result is required.

A typical call for a UNIX command is

```
ctys-beamer -R app1 -- ls -l -h rpm
```

which performs the directory listing on the host app1. The following call

```
ctys-beamer -R app1,app2 -- ctys -a list
```

creates **two independent** LIST action outputs of running ctys sessions, while the call

```
ctys -a list app1 app2
```

creates **one** LIST output for both hosts, which could be listed e.g. by overall SORT option. The following call

```
ctys-beamer -R netscan/all -- ctys targethost
```

executes independent ctys scans of the 'targethost' remotely on each entity listed within the GROUP 'netscan/all'. While the call

```
ctys targethost
```

executes locally - ctys scanning 'targethost'. The call

```
ctys-beamer -R app1%netscan/all -- ctys targethost
```

executes from the relay-host 'app1' independent remote 'ctys' on each entity within the GROUP 'netscan/all' for the scan of the 'targethost'.

The following examples show the creation of so called encrypted tunnels.

```
ctys-beamer -b 0 -z 2 \
  -Y \
  --ssh-tunnel-holdtime=60 \
  --mode=HD \
  -R :3333%delphi%tst/grouptst02:7777%lab01 \
  -- ctys ws22swi
```

This example particularly shows the handling of GROUPS and here the assignment of port to each entity contained in the GROUP. The exception are GROUP members, which have already a port assigned. The assigned ports have (for now - will be changed) priority over call options.

Background Operations:

For background operation the common option '-b' is evaluated, which is by default set suitable for interactive dialogue operations at the command line to **SYNCHRONOUS** and **SEQUENTIAL** mode.

LIMITATION:

The current version supports **DISPLAYFORWARDING** only. **CONNECTIONFORWARDING** is foreseen for a later version. Some options may vary for the different modes. The SSH modes - performing native ssh calls - may be used for now with common IP addressing([<user>@](<host-name>|<host-address>)[:<#port>]), whereas the CTYS modes - which use ctys as call interface - may cope the full range of addressing.

OPTIONS

Additional options are transparently bypassed to the internal 'ctys' call. This is particularly the case for the '-Y' option activating '**ForwardAgent yes**'/'-A' of OpenSSH.

-b <background-mode-args>

Refer to "ctys" generic options for additional information.

-ctys-predetach-holdtime=<timeout-secs>

The holdtime before closing local foreground process. This applies in case of

'-mode=CTYSHOPS' in order to prevent the early release of current tunnel for display of remote output. The default value is 10seconds, which should suit in almost any case. In almost any case about 5seconds should be OK.

The parameter is not senceful applicable in case of synchronous operations by bypassed '-b' option of ctys..

-d <debug-args>

Refer to "ctys" generic options for additional information.

-display-only

This deactivates the actual final execution and displays the results only when e.g. '-d pf' is activated. The intermediate calls, e.g. for remote evaluation of free ports for an SSH tunnel are evaluated.

-getfreeport

This returns the first available free port on the execution site. Some configuration parameters like the SEED for handling the random generation within a range are provided.

-h

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

-H <help-option>

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

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

-L <remote login>

The remote user for the inherent beamer function to place the execution immediately on an arbitrary execution relay. The semantics is somewhat different from the ordinary remote operations, which is application specific, whereas this 'beam-up' is a generic pre-execution forwarding.

-mode=<tunnel-mode>

Sets the mode for the encryption tunnel to be created for remote execution. Current version supports only one mode.

CTYSHOPS|CH|0

The encryption is performed in assembled sections, where the intermediate peers provide sections of the segment. The executable used is **ctys**, thus **ssh** is used as internal call only. The security of the intermediate hops has basically still to be assured, but the internode communications ist peer-to-peer encrypted.

SSHCHAIN|SC|1

The encryption is performed in assembled sections, where the intermediate peers provide sections of the segment. The security of the intermediate hops has basically still to be assured, but the internode communications ist peer-to-peer encrypted.

-R <remote-host-list>

The remote hosts including the relays on the way to be used. This enables the call of multiple execution hosts as well as an arbitrary chain of relays to be actually passed. The main advance of this function is to pierce multi-level firewall solution with chained DMZs, and still provide reliable and secure connections.

The relay chain could be opened and terminated by the optional assignment of access and termination ports, else the first free within the defined range is used. For the intermediate hops ports could be assigned too, but these are not checked for availability, when automatic assignment for intermediate ports is chosen (default), free ports are assigned as available. The port assignment may not be applicable to any mode of interconnection.

Valid calls are:

```
ctys-beamer -R :3333%hop1%hop2%target ....
ctys-beamer -R :3333%hop1%hop2:3333%target ....
ctys-beamer -R :3333%hop1%hop2:3333%target:22 ....
```

Another closely related advance is to circumvent the routing of TCP/IP by usage of well-defined hops with still reliable access by OpenSSH.

In case of multiple chains these are separated by a ',', which terminates the previous and starts another definition of a chain of hops. Each chain starts at the current node and takes the given path of hops. The definition of multiple chains of hops implicitly leads to parallelism by multiple executions of the identical target process. This could be a quite smart solution for a number of cases, but may be a serious drawback for others. E.g. in case of required unique identifiers exactly-cloned parallel execution threads are impractical, whereas a multiple ping measurement of one target from 100 hosts initiated by one single call may fit quite well.

-ssh-hop-holdtime=<#timeout>

In case of CTYS-modes with applied asynchronous background mode the channel will be hold in idle mode at least for the assigned time. The value could be a numeric value in seconds, or provided with a valid 'sleep' unit-postfix.

-ssh-tunnel-holdtime=<#timeout>

In case of SSH-modes this is the timeout value for the final termination point of a one-shot tunnel. The value could be a numeric value in seconds, or provided with a valid 'sleep' unit-postfix.

-V

Version.

-x11

This parameter activates the X11Forwarding for ssh-tunnels '-mode=SC'.

-X

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

. ARGUMENTS

The command to be executed on the final execution host. This command of course could be any command - including the ctys-beamer itself - which again calls a nested remote operation.

EXIT-VALUES

0: OK: Result is valid.

1: NOK: Erroneous parameters.

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

8.3 ctys-cloneVM

SYNTAX

ctys-cloneVM

```

[--ctys-standard-structure]
[-d <debug-level>]
[--defaults-file=<file-name>]
[-f]
[-h]
[-H <help-options>]
[--ip=<IP-address>]
--label=<the-target-vm-label>
--label-old=<the-source-vm-label>
[--mac=<MAC-address>]
[--no-create-image]
[--no-ctys-standard-structure]
[--no-save-para-kernel]
[--source-directory=<directory>]
-t <session-type>
--target-directory=<directory>
[--tcp=<TCP-DNS-name>]
[-V]
[--vm-state=<VSTAT>]
[--uuid=<UUID>]
[-X]

```

DESCRIPTION

ctys-cloneVM provides the automated duplication of present virtual machines. Therefore first the VM is copied to a defined target and second the configuration parameters are altered to new values. The duplicated new VM is executable without any further required modification. The conversion between different hypervisors - a.k.a. session types - is provided by the tool **ctys-convertVM**.

The **current version** of **ctys-cloneVM** could be applied by **batch processing mode to offline-VMs**, where basic values could be consistently altered for the new VM. **Local targets** are supported only, thus a remote VM target requires the remote copy by OS means.

Additional fine grained configuration parameters could be set by **ctys-createConfVM**. The target VM could be stored locally or to a remote location by secure connections via 'scp' from the OpenSSH package. The full scope of authorisation for remote connections by OpenSSH could be applied, this comprises particularly SSH-Keys and Kerberos.

The current version provides the following features:

- Copy and configuration of VMs including hypervisor:
This is provided for the following supported systems - **QEMU/KVM** , and **XEN** . The actual modification of the included operating system has to be proceeded by the provided native facilities.
- Copy and configuration of VMs with required additional hypervisor registration:
This is provided for all supported systems - **VBOX** , and **VMW** , either open source, or commercial closed source products. Some additional administration such as registration of the VMs is required for some versions of the hypervisors by their native tools. The actual modification of the included operating system has to be proceeded by the provided native facilities.

The following table shows a sumup of available features for the current version. The sessiontypes requiring specific tools for their repositories has to be executed on a platform with installed tools and probably with running and enabled hypervisor including helper processes. In cases with specific storage addressing the appropriate mapping functions are required.

Hypervisor/Emulator	VM-Copy	VM-Config	VM-Registration	GuestOS-Config
QEMU/KVM	x	x	ctys	ffs. 3.)
VBOX	x	x	ctys	ffs. 3.)
VMW	x	x	ctys + OEM 2.)	ffs. 3.)
VMWE - VMware ESX(TM)	ffs.	ffs.	ffs.	ffs. 3.)
VMWEi - VMware ESXi(TM)	ffs.	ffs.	ffs.	ffs. 3.)
XEN	x	x	ctys	ffs. 3.)
XENS - XenServer(TM)	ffs.	ffs.	ffs.	ffs. 3.)

Table 8.2: Clone Automation

- ctys Ctys database by ctys-vdbgen and ctys-vhosts.
 OEM Registration within the native inventory of the OEM.
- 2.) Requires for some versions additional registration in runtime database by VMware-Gui. Registration for VMware-Server-2.x could be performed implicitly when providing User and non-crypted Password at the commandline, which should be avoided.
 - 3.) Requires the native configuration, which depends on the OS. E.g. a MS-Widows(TM) workstation may require the reconfiguration of system parameters for networking and in addition the new registration in it's ADS domain.

The available call options are partly set by the commandline options, but could the majority of the values are used by pre-assigned environment variables. This is similar to the call of **ctys-createConfVM**, the variables are the same. The modification of the configuratrion parameters is accomplished by **ctys-createConfVM**. Therefore the values are just passed through to **ctys-createConfVM**.

OPTIONS

–ctys-standard-structure

Adapts to complete standard storage structure of ctys. This is mainly the assembly of the whole machine into the VM directory. E.g. in case of VirtualBox

the 'Machine' and 'Harddisk' files are stored within the VM directory, whereas else at least the 'Machine' files remain within the standard directory.

This case is the default case.

-d <debug-level>

Same as ctys.

-defaults-file=<file-name>

Filename of pre-stored default values for copied VM, the creation could be performed by usage of **ctys-createConfVM**.

-f

Force the processing when the target directory is present.

-h|-help|-help

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

-H <help-option>

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

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

-ip=<IP>

The IP address, else the source address is used. This value is currently stored in the configuration file only and has to be configured separately within the GuestOS.

-label=<args>

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

-label=<args>

The label - the name - for the source VM to be cloned.

list-env-var-options|levo

List of current options to be set by environment variables. For additional information refer to **ctys-createConfVM**.

-mac=<MAC>

The MAC address, else the source address is used. This value is almost as important as the UUID and has for the networking domain the same functionality. In most cases the remaining TCP/IP parameters depend, on this value and therefore may be automatically derived from a MAC-Mapping database such as DHCP.

The current version supports one MAC address only, this is applied to the first interface only. When more than one interface is required, this has to be set manually.

-no-create-image

Suppresses the copy of the virtual file image.

-no-ctys-standard-structure

Adapts to the storage structure of the supported products, which may be varied partially by ctys specific storage within the defined directory for the VM.

-vm-state=<VSTAT>

The state to be set for the target machine. For additional information on values refer to **ctys-states**.

-no-save-para-kernel

Suppresses the copy of the external runtime kernels. External runtime kernels are frequently required for paravirtualized DomU of Xen and in some cases of QEMU.

-source-directory=<directory>

Parent directory for source VM which is required to be stored within a subdirectory named by '-label-old'. The value of '<directory>' is required to be absolute. If this option is not set, the call directory is used as '<directory>/<label-old>'.

-t <session-type>

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

-target-directory=<directory>

Parent directory for storage of cloned VM. The actually created directory path is '<directory>/<label>'. The value of '<directory>' is required to be absolute.

-tcp=<MAC>

The TCP address, else the source address is used. The DNS name which has to be consistent with the IP address.

-V

Version.

-X

Terse.

. ARGUMENTS

None.

. EXIT-VALUES

0: OK: Result is valid.

1: NOK: Erroneous parameters.

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

8.4 ctys-config

SYNTAX

ctys-config

```

[-d <level>]
[-D <deepness-level>]
[-e [<group-list>|<path-list>]]
[-F]
[-h]
[-H <help-options>]
[-S]
[-V]
[-X]
[<configfile-list>]

```

REMARK

This utility is for now alpha release, and may have some gaps.

DESCRIPTION

ctys-config lists and displays configuration files.

The display can be applied for specific groups, subtrees, or for nested groups. Also a tree-view based on the UNIX command tree is included, which supports the listing of components and subtrees from the standard paths.

The main application is the usage by '**ctys-config -e**' for starting the default browser '**konqueror**' for browsing the configuration directories.

OPTIONS

-d <level>

Debug.

-D <deepness-level>

Display structure of configuration files with the given level of recursion deepness.

-e [<conf-file-list>|<path-list>]

Edit one or multiple configuration files by **Emacs** or another preconfigured editor. The default display mode is **konqueror** in tab-view for multiple directories. When no argument is provided the default paths

```
'MYCONFPATH=${CTYS_LIBPATH}/conf/ctys'
```

and

```
'${HOME}/.ctys'
```

are displayed. When a directory name and/or file list - is provided, these are searched for the relative path provided, first from current directory, than by using each **existing entry** within the standard path as a prefix. Absolute paths are just evaluated for providing a valid filename/pathname.

The default editor could be configured within the distribution specific configuration file by

CTYS_CONFIGEDIT. Currently preconfigured editors/starters are: Emacs, gvim, vim, konqueror, and nautilus.

-F

Display file structure as a directory-tree including contained files, the option **-D** controls the level option.

-h

Print help.

-H <help-option>

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

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

-S

Display directory structure of the configuration files with the given level of deepness in **-D** option.

-V

Version.

-X

Terse.

.

ARGUMENTS

ffs.

.

EXIT-VALUES

0: OK: Result is valid.

1: NOK: Erroneous parameters.

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

8.5 ctys-convertVM

SYNTAX

ctys-convertVM

FFS.

DESCRIPTION

ctys-convertVM provides the automated ...

FFS.

OPTIONS

FFS. . ARGUMENTS

None. .

EXIT-VALUES Result is valid. Erroneous parameters. Missing an environment element like files or databases.

8.6 ctys-createConfVM

SYNTAX

ctys-createConfVM

0: OK: NOI~~2~~: NOK:

```

[--auto-all]
[--auto]
[-C]
[--create-image]
[-d <debug-level>]
[-D <directory>]
[--defaults-file=<file-name>]
[--defaults-file]
[--defaults-file-create]
[--defaults-file-create-with-force]
[--expert]
[-h]
[-H <help-options>]
(
  --label=<label>
  | --label <label>
)
[--list-env-var-options|--levo]
[--no-create-image]
[--no-create-conf-data]
[--no-load-defaults-file]
[--no-save-para-kernel]
[--no-virtiohdd]
[--no-virtionet]
[--no-write-conf-files]
[--save-para-kernel]
-t <session-type>
[--virtiohdd]
[--virtionet]
[-V]
[-X]

```

DESCRIPTION

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

VMs the tools **ctys-cloneVM**. and **ctys-convertVM**. could be used.

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

The current version provides the following support:

- Creation of configuration file and wrapper script for the integration of present VMs:
This is provided for all supported systems - **QEMU/KVM** , **VBOX** , **VMW** , and **XEN** , either open source, or commercial closed source products. The actual installation of the virtual machine and the included operating system has to be proceeded by the provided native facilities.
- Creation of integration files and additionally the raw VM:
This is provided for the hypervisors **QEMU/KVM** , and **XEN** . The support for VirtualBox(TM) and VMware(TM) products by native interfaces is planned. For this variant the complete installation and configuration of the virtual machine is provided by the UnifiedSessionsManager, whereas the operating system has to be installed by the provided native means.
- Complete turn-key installation of VMs including the guest OS:
This is provided for the **QEMU/KVM** , and **XEN** in combination with appropriate operating systems. The installation procedure is divided into two stages.
 - (a) Creation and configuration of the raw VM
 - (b) Installation of the contained OS

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

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

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

boot of installers.

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

Table 8.3: Installation Automation

Following install modes for automated install procedures could be utilized.

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

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

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

For example

```
ACCELERATOR=KVM \
DIST=Scientific \
RELEASE=5.4.1 \
OS=Linux \
OSVERSION=2.6.18 \
ctys-createConfVM -t qemu --label=tst213 --auto-all --no-create-image
```

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

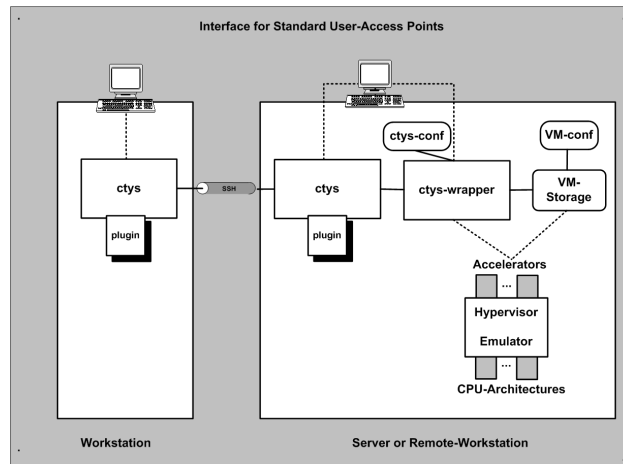


Figure 8.1: ctys distributed access

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

- bootmode
- check
- console
- print
- vncaccessdisplay

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

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

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

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

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

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

-bootmode=<boot-modes>

The bootmode for now implements the following options

- **INSTALL**

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

- **PXE**

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

- **VHDD**

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

-check

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

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

-console=<help-option>

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

-initmode

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

-initmodeonly

The virtual storage devices are created and initalized only.

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

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

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

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

-listoptions

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

-print

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

-vncaccessdisplay=<#nr>

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

OPTIONS**-auto**

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

-auto-all

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

-c

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

-d <debug-level>

Same as ctys.

-D <directory>

Directory for storage of created configuration files.

-defaults-file

Use defaults file, name is '<LABEL>.defaults'.

-defaults-file=<filename>

User defaults file, name is as provided.

-defaults-file-create

Create defaults file, for later usage, file name is as provided. If no filename is given, the default name is '<LABEL>.defaults' The file contains all configuration variables, where the values are protected if already set.

-defaults-file-create-with-force

Create defaults file, for later usage, file name is as provided. If no filename is given, the default name is '<LABEL>.defaults' The file contains all configuration variables, where the values overwrite any previous value set before execution. In this mode the values could only be altered by interactive dialogue.

-expert

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

-h|-help|-help

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

-H <help-option>

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

installed manuals.

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

-label=<args>

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

list-env-var-options|levo

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

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

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

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

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

Actually used sources for default values:

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

Applicable modifications:

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

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

```
VAR name:Initial Value
```

```
C_SESSIONTYPE:QEMU
  LABEL:testDisplay
  MAC:
  IP:
  BRIDGE:
```

```

DHCP:
NETMASK:
TCP:
GATEWAY:

```

```
EDITOR:acue
```

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

```

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

```

```

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

```

```

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

```

```
DEFAULTBOOTMODE:HDD
```

```

DEFAULTINSTTARGET:/hda.img
BOOTIMAGE_INST_SIZE:8G
BOOTIMAGE_INST_BLOCKSIZE:256M
BOOTIMAGE_INST_BLOCKCOUNT:32
BOOTIMAGE_INST_BALLOON:y

```

```

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

```

```
VMSTATE:ACTIVE
```

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

```
-bash-3.2$
```

-no-create-image

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

–no-create-conf-data

Suppresses the creation of configuration data. This implies the suppression of the write of configuration data, therefore should be used for the creation of defaults file only, which is processed by filtering an existing configuration file.

–no-load-defaults-file

Suppresses the initialization by reading a present defaults file. If not set a present file is sourced into the process by default.

–no-save-para-kernel

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

–no-virtiohdd

Suppresses the usage of virtiohdd for HDDs.

–no-virtionet

Suppresses the usage of virtionet for NICs.

–no-write-conf-data

Suppresses the storage of configuration data into files, thus no data including the wrapper files is stored.

–save-para-kernel

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

-t <session-type>

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

-V

Version.

–virtiohdd

Activates the usage of virtiohdd for HDDs.

–virtionet

Activates the usage of virtionet for NICs.

-X

Terse.

. ARGUMENTS

None.

.

EXIT-VALUES

0: OK: Result is valid.

1: NOK: Erroneous parameters.

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

8.7 ctys-distribute

SYNTAX

ctys-distribute

```

[-d <debug-options>]
[-F (1|2|3|force|forceclean|forceall)]
[-h]
[-H <help-options>]
[-l <remote login>]
[-M <access-mask>]
[--menu-create=(PRIVATE|SHARED)]
[-P (
    (UserHomeCopy|UHC)
    |(UserHomeLinkonly|UHL)
    |(SharedAnyDirectory|SAD),<subopts>
    |(SharedAnyLinkonly|SAL),<subopts>
    |(AnyDirectory|AD),<subopts>
)]
[-V]
[-X]
<list-of-target-accounts>

```

DESCRIPTION

ctys-distribute is the main interface for local and remote installation. The installation of a single target and an arbitrary list of remote targets for automated bulk installations is provided.

Various distribution and installation methods for distribution packages and installation types could be applied, these are for the current version:

File based installation

File based installation uses an existing installation or an arbitrary copy of a valid directory tree. The contained revision could be either installed as a copy or as references by symbolic links for execution only. The configuration files are installed locally for the calling user in any case. These are not replaced by default, just when '-F' option is used.

The **ctys-distribute** utility therefore resolves the actual physical location of the executable and performs the installation solely based on its own physical directory tree. Therefore any copy of the ctys subtree could be used for independent local and remote installation, which is consistent as defined by the implemented release.

This has particularly some advance for custom installations, e.g. when the set of documents is required to be reduced it is sufficient just to delete the

relevant files/directories.

In case of remote installation the current version uses excessive local-remote-interaction by **ssh/scp** based commands, thus for practical reasons requires automated remote login procedures. These could be for example either based on Kerberos or SSH-Keys. This might be reduced to at least one authentication in future versions.

The call of

```
<path-to>/ctys-distribute.sh -F 2 -P UHC userA@hostB
```

installs the complete copy contained within the '`<path-to>`' directory tree. Therefore symbolic links are resolved to their actual physical paths, the resulting directory path to a valid copy is used. The copy has to be valid, but not necessarily to be installed, or the current users active installation. The default when omitting a '`<path-to>`' is resolved by the PATH search algorithm and proceeded similar to the previous case. Resulting in using the current installation as the install source for the user '`userA@hostB`', where the current configuration for '`userA@hostB`' is deleted and replaced completely due to the '`-F 2`' option. Where **UHC** is the shortcut for '**UserHomeCopy**'

The call of

```
<path-to>/ctys-distribute.sh -F 2 -P UHL
```

installs the complete current installation on for the user '`userA@hostB`', where the current configuration is deleted and replaced completely by '`-F 2`', and the source executables are referenced by symbolic links only. The documents are physically shared by search-path references. Where **UHL** is the shortcut for '**UserHomeLinkonly**'

Once installed one of the following calls may be applied for further help. When a full installation with has proceeded, the suffix '`.sh`' could be omitted for the created symbolic links:

```
<path-to>/ctys-distribute.sh -H man
<path-to>/ctys-distribute.sh -H html=base
<path-to>/ctys-distribute.sh -H html=doc
```

```
<path-to>/ctys.sh -H man
<path-to>/ctys.sh -H pdf
<path-to>/ctys.sh -H html
```

```
<path-to>/ctys.sh -H html=base
<path-to>/ctys.sh -H html=doc
```

```
<path-to>/ctys.sh -H path
<path-to>/ctys.sh -H list
```

The following keys are not case sensitive, additionally the uppercase letters could be used as abbreviations - which are not case sensitive too, e.g. '`UHC`' for '`UserHomeCopy`', or '`uHc`'.

UserHomeCopy

installs for the current local or remote user a user-local copy into it's HOME directory with the standard ctys-directory structure.

UserHomeLinkonly

installs for the current local or remote user a partial user-local copy, where the specific configuration files are copied, whereas the executables and documentation is set as symbolic links only.

SharedAnyDirectory

variant installs a local copy for share usaged, thus sets up a common installation within any directory, but should be in accordance to common standards of the actual UNIX variant.

AnyDirectory

variant installs a local copy only for shared usage.

Package based installation

Package based installation is forseen by means of the standard mechanisms. "ctys-distribute" could be used as a installer for the following install applications:

rpm-CentOS

The distributions **rpm-CentOS**, **rpm-Fedora**, and **rpm-OpenSUSE** are supported by the provided rpms. In the "install section" ctys-distribute can be applied as:

```
INSTBASE=${RPM_BUILD_ROOT}/usr/share
```

```
INSTTARGET=${INSTBASE}/${RPM_PACKAGE_NAME}-${RPM_PACKAGE_VERSION}
```

```
${PWD}/bin/ctys-distribute.sh -F 1 -P AnyDirectory,${INSTTARGET} local
```

This installs the complete package without configuration of user accounts.

The user accounts have to be configured by calling:

```
/usr/share/bin/ctys-distribute.sh -F 1 -P UserHomeLinkonly localhost
```

ATTENTION:

When a previous installation is present the configuration directories should be updated too due to the circumstance, that these also contain some dynamic evaluation code. But these probably may contain some custom patches, thus the '-F 1' option preserves the configuration directory '\$HOME/.ctys'. For forced update these may either be removed manually, or by usage of '-F 2' option.

deb-debian

ffs.

pkg-OpenBSD

ffs.

pkg-Solaris

ffs.

The utility scp of the OpenSSH package is used for any transfer, thus remote SSH access is required.

The call evaluates its own actual physical file location by recursive dereferencing symbolic links and uses the actual containing distribution for

further steps. Thus if a specific distribution outside the search path is to be installed, the call path has to be provided explicitly: `<path-to-installbase>/bin/ctys-distribute`

OPTIONS

-d `<debug-args>`

Refer to "ctys" generic options for additional information.

-F `<args>`

Bypassed to `ctys-install1.sh`

-h

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

-H `<help-option>`

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

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

-l `<remote login>`

The remote USER to be used as a EMail style prefix for all given remote hostnames within the `<list-of-target-accounts>`. This option is supported with hostname-only targets, it cannot be intermixed with EMail style `<USER>@<HOST>` targets.

-M `<access-mask>`

Access mask for post-chmod, where the value is literally passed to:

```
"chmod -R <access-mask> <base-lib-dir>"
```

The user-local file access permissions remain unmodified.

-menu-create=(PRIVATE|SHARED)

Creates menu entries during installation by call of **ctys-xdg**. The creation of SHARED manus requires root permissions. The **-force** option is not provided, thus the entries could be installed on supported and verified desktops only. For the remaining the tool **ctys-xdg** has to be called manually.

-P `<pkg-type>`

This option supports the selection of prepared installation packages, which are automated by preconfigured parameters or to be finished by interactive dialogue from the command line interface.

The packages are prepared during build operations within the local "packages" directory for the current version.

For system directory appropriate access permissions are required.

- **(UserHomeCopy|UHC):**

Creates a copy in `$HOME/lib` and sets links to `$HOME/bin`.

- **(UserHomeLinkonly|UHL):**

Creates sets links to `$HOME/bin` from execution directory.

- **(SharedAnyDirectory|SAD), <install-directory>[, <link-directory>]:**

The links will be set within the standard directory conventions.

- `<install-directory>`: Directory for installation of non-user specific files.

- `<link-directory>`: The links to be set, default is `"$HOME/bin"`.

- **(SharedAnyLinkonly|SAL), <reference-directory>, <link-directory>:**

Any directory could be linked, this could be used e.g. for providing symbolic links within standard search PATH.

- <reference-directory>: Directory for targets of symbolic links to non-user specific files.
- <link-directory>: The links to be set.

- **(AnyDirectory|AD), <install-directory>:**

Copies only without setup of configuration. Foreseen for build process only.

- <install-directory>: Directory for installation of non-user specific files.

-V

Version.

-X

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

. ARGUMENTS

The target accounts appropriate for the ssh/scp-call, where the install process should be executed.

The provided formats are a list of EMail style hostnames or pure hostnames. EMail style hostnames are not supported in conjunction with the "-l" option. For any hostname without a user-prefix either the current USER or the user from the "-l" option is prepended.

. EXIT-VALUES

0: OK: Result is valid.

1: NOK: Erroneous parameters.

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

8.8 ctys-dnsutil

SYNTAX

ctys-dnsutil

```

[-c]
[-C]
[-d <debug-level>]
[-h]
[-H <help-options>]
[-i]
[-l <USER>]
[-n]
[--reverse <runtime states>
=
  [(REVERSE|R|-),]
  PING|SSH
  [,PM|VM]
]
[-V]
[-X]
[<dns-server-list>]

```

DESCRIPTION

ctys-dnsutil supports the display of data which is mainly based on the data requested from DNS by usage of "host -l <server>" call. The only current application of this utility is to generate lists as input for additional processing or display. Extended queries are supported by 'ctys-vhost' utility . It has to be recognised, that not all machines might be handled by a reachable DNS server. This is particularly true for VMs located within host-only-networks performing on isolated networks by means of routing.

One important application is the usage of this tool for the LIST action of plugins from PMs. The output list will be used as initial data set for actual available active PMs. Additional constraints related to actual runtime-state for members of raw-list will be applied for various tools and several post-analysis. The most basic checks are based on ping and ssh access checks, but the type of the machine - PM or VM - and the hierarchy could be evaluated by several approaches.

For an initial definition and assignment of a managed PM the "ctys-genpmconf" utility has to be executed. The generated data from the directory /etc/ctys.d/pm.conf is used as an final proof, that the polled TCP/IP-address is related to a PM. A VM is defined as a contained VM characterised by it's configuration file, which is in the case of current supported VMs an ASC-II file with specific syntax.

OPTIONS

-c

Uses "ctys-vhost" for PM/VM evaluation instead of polling the real instance. The basic implicit access checks for ping and ssh are still performed. The databases for ctys-vhost has to be prepared, thus not usable for initial scan to generate that databases of course.

-C

Basically the same as "-c", but here no implicit dynamic checks are performed at all. The only dynamic evaluated data is the query of the DNS server.

-d <debug-level>

Same as ctys.

-h

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

-H <help-option>

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

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

-i <quad-dotted-IP>

Show numerical TCP/IP-Address. This is supported in combination with "-n" option only without "-X" option.

-l <USER>

Remote user to be used for network logins.

DEFAULT=CTYS_NETACCOUNT(DEFAULT->USER)

-n

Show TCP/IP-Address as name. This is supported in combination with "-i" option only without "-X" option.

-reverse <runtime states>

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

Only the possible targets which are operable or 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.

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

- **REVERSE|R|-**

This reverses the resulting set, thus the "not matching" states only will be shown.

- **PING[:<packetcnt>[%<timeout>]]**

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.

On some nodes the timeout may take some time, so be patient when such a node is in the DNS query.

This option might be really senseful, once SSO is established and probably a common net-access-user with limited permissions for probing-only is configured.

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

- **PM**

Checks whether machine is a PM. Therefore first SSH-check is activated and performed, and on the remaining set of accessible machines the PM-check is performed.

PM accessibility is defined as the accessibility of the running OS on PM and the presence of the file "/etc/ctys.d/pm.conf".

- **VM**

Checks whether machine is a VM. Therefore first SSH-check is activated and performed, and on the remaining set of accessible machines the VM-check is performed.

VM accessibility is defined as the accessibility of the running OS on VM and the presence of the file "/etc/ctys.d/vm.conf".

-V

See ctys, version output.

-X

See ctys, terse for machine output.

. ARGUMENTS

Any DNS server to be used in "host" call.

.

EXIT-VALUES

0: OK: Result is valid.

1: NOK: Erroneous parameters.

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

8.9 ctys-extractARPlst

SYNTAX

ctys-extractARPlst

```

[-d <debug-args>]
[-E]
[-n|-i|-m]
[-h]
[-H <help-options>]
[-p <db-dir-path>]
[-P]
[-q]
[-V]
[-X]
<ping-hostlist-in-same-segment>

```

DESCRIPTION

ctys-extractARPlst 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

-d <debug-args>

Refer to "ctys" generic options for additional information.

-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, refer to `"-H"` for additional information.

-H <help-option>

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

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

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.

8.10 ctys-extractMAClst

SYNTAX

ctys-extractMAClst

```

[-d <debug-args>]
[-E]
[-n|-i|-m]
[-h]
[-H <help-options>]
[-p <db-dir-path>]
[-P]
[-q]
[-V]
[-X]
<pathname-of-dhcpd.conf>

```

DESCRIPTION

ctys-extractMAClst generates a 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

-d <debug-args>

Refer to `"ctys"` generic options for additional information.

-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

Default output is "-n".

- -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

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.

8.11 ctys-genmconf

SYNTAX

ctys-genmconf

```

[-d <debug-options>]
[-h]
[-H <help-options>]
[-I <TCP-address>]
[-k <ctys-vhost-search-key>]
[-l <remote-user>]
[-M <MAC-address>]
[--no-progress]
[-P]
[--progress]
[-r [(WITHIP|WITHMAC|ALL)]]
[--rounds=<rounds>%<cpu-cycles>%<hdd-cache-cycles>\
    %<hdd-write-through-cycles>]
[--testfs=<test-directory-storage>]
[-u]
[-U <UUID>]
[-V]
[-X]
[-x (PM|VM)]
<execution-target-list>

```

DESCRIPTION

ctys-genmconf generates the initial configuration entry for a PM or VM for

- `"/etc/ctys.d/[pv]m.conf"`.

The content of this file is required by the VMSTACK feature 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 is 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

-d <debug-args>

Refer to "ctys" generic options for additional information.

-h

Show help.

-H <help-option>

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

For additional help refer to the documents or type **ctys -H 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 VHOST-SEARCH-ARGS argument for "ctys-vhost" is applicable.

-l <remote-user>

Refer to "ctys" generic options for additional information.

-M <MAC-address>

Preset value.

-no-progress

Suppresses the displays of progress indicators.

-P

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

Default output is stdout.

-progress

Displays various progress indicators for HW capabilities measurements.

-r <args>

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 necessary, but could be present.
- ALL Enumerates all ethernet interfaces which is "WITHIP||WITHMAC", localhost is excluded.

-rounds=<args>

The rounds argument changes the basic amount of measurement cycles for predefined measurement patterns. The following arguments could be applied:

- <rounds>
The number of cycles for all tests except the write-through tests.
- <cpu-cycles>
The number of cycles for the CPU load test pattern.
- <hdd-cache-cycles>
The number of applied cache-storage pattern
- <hdd-write-through-cycles>
The number of write-through pattern. This test is frequently long running, the data is scaled to 2*RAM when the storage space is available, else to storage/2.

-testfs=<test-filesystem>

The testfilesystem for storage of data, the default is set to '/var/tmp'.

-u

Generate a UUID.

-U <UUID>

Preset value.

-V

Show version.

-x <category>

Configuration files for the various machine categories. The output is slightly different.

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

. ARGUMENTS

An optional list of <execution-target>. When the "-P" option is chosen, the remote files will be updated, when sufficient 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 synchronous and sequential.

EXIT-VALUES

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

8.12 ctys-getNetInfo

SYNTAX

ctys-getNetInfo

```

[--(brdglst|--bl)=<contained-if-prefix>]
[--(brdglstxen|blx)=<contained-if-prefix>]
[--(brdgcheckxen|bcx)=<bridge>]
[--checkif=<if>]
[-d <debug-options>]
[--firstfreeport=<MIN> [%<MAX>]]
[--getbroadcast=<if>]
[--gethostip=<ip|dns>]
[--getip=<if>]
[--getmac=<if>]
[--getmask=<if>]
[-h]
[-H <help-options>]
[--ifbrdcast [=<if>]]
[--iflist|--il}
[--iffirst|--if]
[--(iflistnames|iln) [= (ALL|PREFIX|POSTFIX|MATCH) [, <namestring>]]]
[--iflistmacip|--ilmi]
[-V]
[-X]

```

DESCRIPTION

ctys-getNetInfo displays various network related information. It can be executed local and remote by usage of 'ctys-beamer'. The main task is to provide support for analysis of networking problems in relation to the actually available network-bridges.

OPTIONS

- brdglst[=<contained-if-prefix>]**
Lists bridges. An optional string as prefix for contained interfaces is applicable. E.g. the string 'peth' lists all bridges related to Xen.
- brdglstxen[=<contained-if-prefix>]**
Lists Xen bridges. An optional string as prefix for contained interfaces is applicable. The default is 'peth'.
- brdgcheckxen=<bridge>**
Check whether brige is a Xen bridges.
- d <debug-args>**
Refer to "ctys" generic options for additional information.
- h**
Show help.

-H <help-option>

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

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

-iflist

List interfaces.

-iflistnames

List interface names.

-iflistmacip

List interface IP and MAC addresses.

-V

Show version.

. ARGUMENTS

none

.

EXIT-VALUES

0: OK: Result is valid.

1: NOK: Erroneous parameters.

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

8.13 ctys-groups

SYNTAX

ctys-groups

```

[-c]
[-d <level>]
[-D <deepness-level>]
[-e [<group-list>|<path-list>]]
[-F]
[-h]
[-H <help-options>]
[-l]
[-m <1|2|3|4|5|5x|6|7|7u|8|8u>]
[-S]
[-V]
[-X]
[<group-list>]

```

DESCRIPTION

ctys-groups lists and displays groups from groups DB and groups caches. This tool utilizes ctys-vhost and adds some usability to the more detailed interface parameters.

The display can be applied for specific groups, subtrees, or for nested groups. Also a tree-view based on the UNIX command tree is included, which supports the listing of components and subtrees from the variable CTYS_GROUPS_PATH.

The output format of each group entry for default short-listing is:

```
<size-kbytes> <#hosts>/<#includes> <#total> <group-filename>
```

For additional information of groups syntax refer to "GroupResolution" .

The '-e' option provides by default for Emacs based edit , but could be adapted by configuration of CTYS_GROUPSEDIT

OPTIONS

-c

List cached groups the output format is:

```
<filesize> <lines=members> <group-cache-file-path>
```

-d <level>

Debug.

-D <deepness-level>

Display structure of groups with the given level of recursion deepness.

-e [<group-list>|<path-list>]

Edit one or multiple groups by **Emacs**. When no argument is provided the

CTYS_GROUPS_PATH is evaluated and an Emacs instance is opened with all present directories. When a directory and/or group list - which is actually a file list - is provided, these are searched for the relative path provided, first from current directory, then by using each **existing entry** within **CTYS_GROUPS_PATH** as a prefix. Absolute paths are just evaluated for providing a valid filename/pathname.

The default editor could be configured within the distribution specific configuration file by

CTYS_GROUPS_EDIT. Currently preconfigured editors/starters are: Emacs, gvim, vim, konqueror, and nautilus.

-F

Display file structure of groups with the given level in **-D** option.

-h

Print help.

-H <help-option>

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

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

-l

Lists group definitions.

<filesize> <lines=members> <group-name>

-m <1|2|3|4|5|5x|6|7|8>

List group members in different formats.

- 1 listing
- 2 listing
- 3 listing
- 4 callable for cut-and-paste
- 5 callable for cut-and-paste
- 5x callable for cut-and-paste with absolute overall indexes for groups and hosts,
- 6 listing of contained remote accounts in EMail format
- 7 account list for machine processing
- 7u account list for machine processing, with stripped redundancies.
- 8 host list for machine processing
- 8u host list for machine processing, with stripped redundancies.

-S

Display directory structure of groups with the given level of deepness in **-D** option.

-V

Version.

-X

Terse.

ARGUMENTS

An optional list of groups to be displayed only, if not provided all present groups from directories within the variable `CTYS_GROUPS_PATH` are scanned and displayed. For additional information refer to the `S-OPTION-OF-CTYS-VHOST`. The format of `<group-list>` is:

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

The separator is here slightly different from the "ctys-vhost" option. For ctys-groups optionally the common suboptions-argument separator "%" or the common UNIX arguments separator "SPACE" could be used as separator.

The usage of the listed `<group-names>` is as follows:

- **name**
The name is expected to be a group name, so search is performed for a file of name "name".
- **name/**
The name is expected to be a directory name, so search is performed for a subdirectory within the `CTYS_GROUPS_PATH`. All files within that subtree are listed.
- **name01/name02/name03/**
The path is expected to be a subdirectory pathname, so search is performed for a subdirectory-path within the `CTYS_GROUPS_PATH`. All files within that subtree are listed.
- **/name01/name02/name03/**
The path is expected to be an absolute directory pathname, so search is performed within the subtree of given path only. All files within that subtree are listed.
- **/name01/name02/name03/name**
The name is expected to be an absolute filename, so test is performed for that file only.

The nested containment hierarchy by "include" is expanded before output in any case. The following example lists the **desktop/** and **scan/** directories.

```
ctys-groups scan/ desktops/
```

EXIT-VALUES

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

8.14 ctys-install

SYNTAX

ctys-install

```
[libdir=<default=\$HOME/lib>]
[bindir=<default=\$HOME/bin>]
[templatedir=<default=\$HOME/ctys>]
[remove<default=unset>]
[noconf<default=unset>]
[force<default=unset>]
[forceall<default=unset>]
[forceclean<default=unset>]
[linkonly<default=unset>]
[version|--version|-V]

[-h]
[-H <help-options>]
[-X]
```

DESCRIPTION

ctys-install is the base interface for installation and update. For the common tasks **ctys-distribute** should be preferred, which has a more straight forward interface.

"ctys-install" is used internally by "ctys-distribute" with some detailed options. Therefore several call options are provided on a level of higher granularity. "ctys-install" itself makes sub-calls of "ctys-install1", which is only foreseen to be called by the user via wrapper scripts.

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 common 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.

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 independently 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 directory 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 configuration, GROUP and MACRO files, and the generated cacheDB files.

- **ctys-install forceclean**

The option FORCECLEAN forces the previous deletion of installed components, no backups are created.

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 local 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"

- Centralized Install by Search PATH

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

ctys-install1

Internal call for ctys-install.

OPTIONS

[**libdir**=<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.

DEFAULT=\$HOME/lib

[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 empty 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" alphabetically/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" alphabetically/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]

EXIT-VALUES

0: OK: Result is valid.

1: NOK: Erroneous parameters.

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

8.15 ctys-macmap

SYNTAX

ctys-macmap

```

[-d <debug-options>]
[-n|-i|-m]
[-h]
[-H <help-options>]
[-p <db-dir-path-list>]
[-u]
[-V]
[-X]
<simple awk-regexp>

```

DESCRIPTION

ctys-macmap searches within the given list of 'macmap.fdb' for matching lines, and shows the requested fields of result. Simple awk-regexp 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 displayed 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

-d <debug-args>

Refer to "ctys" generic options for additional information.

-h

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

-H <help-option>

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

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

-n|-i|-m

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

-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!

-u

The output is displayed in uppercase only.

-V

Version.

-X

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

. ARGUMENTS

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
```

The comparison of MAC addresses is non-casesensitive, where the internal storage format is in uppercase. It should be avoided to edit the database manually, but when required the MAC addresses has to be stored with uppercase characters [A-F].

.

EXIT-VALUES

0: OK: Result is valid.

1: NOK: Erroneous parameters.

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

8.16 ctys-macros

SYNTAX

ctys-macros

```

[-a]
[-c]
[-d <debug-level>*]
[-D <deepness-level>]
[-e [<macro-file-list>|<path-list>]]
[-E]
[-F]
[-f <macro-file>]
[-h]
[-H <help-options>]
[-l]
[-N]
[-S]
[-V]
[-X]
[<macro-list>]

```

DESCRIPTION

ctys-macros searches within a macro file and lists the matched macros as requested. Therefore the standard path is searched for a macro file with the provided name or 'default'. The result could be filtered by adding an optional filter rule as argument.

Even though the options **-S**, **D**, and **-F** are already provided for display of sub-trees, the actual evaluation in current version is proceeded within a flat search of the provided

CTYS_MACRO_PATH.

The '-e' option provides by default for Emacs based edit , but could be adapted by configuration of **CTYS_MACROSEDT**.

OPTIONS

- a**
Adds **atoms** to the displayed list.
- c**
Adds **combined** to the displayed list.
- d <debug-level>**
Same as **ctys**.
- D <deepness-level>**
Display structure of macros with the given level of recursion deepness.

-e [**<macro-file-list>** | **<path-list>**]

Edit one or multiple groups by **Emacs**. When no argument is provided the

`CTYS_MACRO_PATH` is evaluated and an Emacs instance is opened with all present directories. When a directory and/or group list - which is actually a file list - is provided, these are searched for the relative path provided, first from current directory, than by using each **existing entry** within `CTYS_MACRO_PATH` as a prefix. Absolute paths are just evaluated for providing a valid filename/pathname.

The default editor could be configured within the distribution specific configuration file by

`CTYS_MACROSEDIT`. Currently preconfigured editors/starters are: Emacs, gvim, vim, konqueror, and nautilus.

-E

List **expanded** definitions. Sets implicitly the "-c" option.

-f **<macro-file>**

The basename of the macro file to be searched within the standard path list. DEFAULT:"default"

-F

Display file structure of groups with the given level in **-D** option.

-h | **-help** | **-help**

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

-H **<help-option>**

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

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

-l

Lists **available macro files** within the standard searchpath. Each of this has to be fully self-contained file, which could be selected by usage of it's basename only. The selection order in case of multiple occurrence is "first-wins".

-N

List **non-expanded** definitions, default is names only. The scope of display (atoms, combined) is not influenced.

-S

Display directory structure of groups with the given level of deepness in **-D** option.

-V

Version.

-X

Terse.

ARGUMENTS

An optional list of macros to be displayed. The names are case sensitive and are expand completely.

EXIT-VALUES

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

8.17 ctys-plugins

SYNTAX

ctys-plugins

```

[-d <common-debug-option>]
[-E ]
[-h]
[-H <help-options>]
[-l <remote-user>]
[--quick-list]
[--quick-tab]
[-t <plugin-type>]
[-T <plugin-preload-typelist>]
[-V]
[-X]
[-Z <set-sudo-ksu>]
<list-of-target-accounts>

```

DESCRIPTION

ctys-plugins is a test-tool for verification and analysis of the current status of the installation of ctys and present plugins.

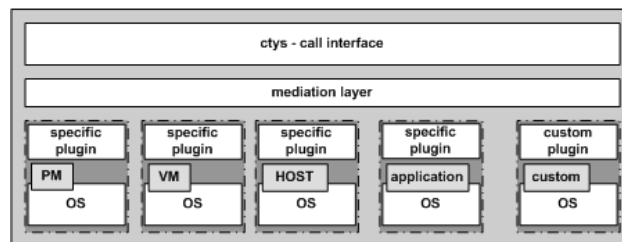


Figure 8.2: ctys SW-Layers and plugins

Due to some restrictions it operates partly on a basic level, involving some specific pre-defined trace levels for output control. The required options are documented in the following sections. Extended trace-levels necessarily require some deeper familiarity with internal design, which is normally not required for top-level handling.

The utility could be performed locally or remotely by full support of ctys-addressing but without context options on the command line. If for whatever reasons remote context options are required, then they 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 one of:
 - `/etc/sudoers`
 - `$HOME/.k5users`
 - `$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**:

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

```
Checking PLUGINS-STATES now...
```

```
-----
```

```
<trace-output>
```

```
-----
```

```
...results to:
```

- 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 individual plugin.

OPTIONS

-d <common-debug-option>

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

Anyhow, the **most important application** is for installation and first time systems configuration by **analysis of the called system utilities**. The difficulties with the bunch of utilized systems calls frequently leads to some trouble, which becomes immediately obvious, once the output of the following call is displayed:

```
"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 seamless 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 **the current node 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 completely 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.

-h

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

-H <help-option>

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

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

-l <remote-user>

The remote user.

-quick-list

Displays a short list of overall states for selected plugins on each target.

```
ctys-plugins -E -T all --quick-list localhost
```

Hostname	Plugin	Accelerator	Version	MAGIC-ID	State
acue@ws2	CLI		01.06.001a09	CLIBASH	ENABLED
acue@ws2	RDP		01.11.007alpha	RDPRD	ENABLED
acue@ws2	VNC		01.10.013	VNCR4	ENABLED
acue@ws2	X11		01.06.001a09	X11	ENABLED
acue@ws2	QEMU	QEMU	01.10.013	QEMU_091	DISABLED
acue@ws2	VBOX		01.11.006alpha	NOLOC	DISABLED
acue@ws2	VMW		01.11.005	VMW_GENERIC	DISABLED
acue@ws2	XEN		01.10.013	DISABLED	DISABLED
acue@ws2	PM	VMX	01.10.008	PM_Linux	ENABLED

-quick-tab

Displays a short table of overall states for selected plugins.

The following call contains the 'ctys-groups' call for stripping off context arguments, which in some cases are valid, thus could not be dropped in general. When this is not done, some might be detected as valid and lead to execution. In future releases a new scanner and parser are foreseen to be provided, which will clear this specific.

```
ctys-plugins -T all -E --quick-tab $(ctys-groups -X -m 8 netscan/all)
```

For limitation of error messages and some ssh info the error channel should be redirected.

```
ctys-plugins -T all -E --quick-tab $(ctys-groups -X -m 8 netscan/all)\
  2>/dev/null
```

This results to the following list, where the output order may vary call-by-call due to parallel execution.

Hostname	PM	CLI	X11	VNC	RDP	KVM	QEMU	VBOX	VMW	XEN
root@lab05.soho	PAE	x	x	x	x	-	-	-	S2	-
root@lab02.soho	VMX	x	x	x	x	-	QEMU	V	-	-
root@hydra.soho	VMX	x	x	x	x	-	-	-	-	-
root@delphi.soho	VMX	x	x	x	x	-	-	-	S1	-
root@olymp.soho	VMX	x	x	x	x	-	-	-	-	-
root@app1.soho	VMX	x	x	x	x	-	-	-	-	-
root@app2.soho	VMX	x	x	x	x	-	-	-	-	-
root@lab04	VMX	x	x	x	x	-	-	-	W7	-
acue@lab04	VMX	x	x	x	x	-	QEMU	-	W7	-
vadmin@delphi.soho	VMX	x	x	x	x	-	-	-	S1	-
acue@lab05.soho	PAE	x	x	x	x	-	-	-	S2	-
acue@lab02.soho	VMX	x	x	x	x	-	QEMU	V	-	-
acue@delphi.soho	VMX	x	x	x	x	-	-	-	S1	-
root@lab03	PAE	x	x	x	x	-	-	-	-	HVM
root@lab01.soho	PAE	x	x	x	-	-	-	-	-	PAR
5000@lab01.soho	PAE	x	x	x	-	-	-	-	-	-
acue@lab03	PAE	x	x	x	x	-	-	-	-	-
acue@app1.soho	VMX	x	x	x	x	KVM	KVM	-	-	-
acue@app2.soho	VMX	x	x	x	x	KVM	KVM	-	-	-

Where the fields have the semantics as given by following list, the values are foreseen to be expanded for more details:

- (a) PM: Type of CPU support
 - VMX: Intel VT-X
 - SVN: AMD SVN
 - PAE: Extended Mem by PAE
- (b) HOSTS: CLI, X11, VNC, RDP
 - x: ENABLED
 - else: DISABLED
- (c) VMs: KVM, QEMU, VBOX, VMW, XEN
 - Common:
 - V: ENABLED
 - -: DISABLED
 - KVM: Fully ENABLED, else check
 - Module
 - VDE-Switch
 - QEMU: ENABLED without KVM-Support, else check
 - Module
 - VDE-Switch

- VMW
 - P[123]: Player
 - S[12]: Server
 - W[67]: Workstation
 - C2: Client only with standalone remote console VMRC.
 - XEN
 - HVM: Fully ENABLED, else check
 - * Kernel
 - * Access: sudo, ksu
 - PARA: ENABLED with para-virtualisation only, else check
 - * Kernel
 - * Access: sudo, ksu
- The output display shows e.g. that:
- On app1+app2: The user acue has access to KVM and QEMU, but not root. This is due to missing VDE-switch for root, which has to be setup by 'ctys-vnetctl'
 - On lab03: The user acue has no access to XEN, but root has. This is due to access permission by sudo for user acue.

-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.

-V

Version.

-X

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

-Z <set-sudo-ksu>

Sets appropriate **access permissions by sudoers and/or kerberos/ksu**. For additonal information refer to Z-OPTION.

. ARGUMENTS

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 synchronuous and sequential.

. EXIT-VALUES

0: OK: Result is valid.

1: NOK: Erroneous parameters.

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

8.18 ctys-scripts

SYNTAX

ctys-scripts

```
[-d <level>]
[-e [<scripts-list>|<path-list>]]
[-h]
[-H <help-options>]
[-l]
[-m [<0>]]
[-V]
[-X]
[<scripts-list>]
```

DESCRIPTION

ctys-scripts lists and displays scripts from `CTYS_SCRIPTS_PATH`.

The '-e' option provides by default for Emacs based edit , but could be adapted by configuration of `CTYS_SCRIPTEDIT`

OPTIONS

-d <level>

Debug.

-e [<group-list>|<path-list>]

Edit one or multiple groups by **Emacs**. When no argument is provided the

`CTYS_SCRIPTS_PATH` is evaluated and an Emacs instance is opened with all present directories. When a directory and/or group list - which is actually a file list - is provided, these are searched for the relative path provided, first from current directory, than by using each **existing entry** within `CTYS_SCRIPTS_PATH` as a prefix. Absolute paths are just evaluated for providing a valid filename/pathname.

The default editor could be configured within the distribution specific configuration file by

`CTYS_SCRIPTEDIT`. Currently preconfigured editors/starters are: Emacs, gvim, vim, konqueror, and nautilus.

-h

Print help.

-H <help-option>

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

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

- l**
Lists group definitions.
`<filesize> <lines=members> <group-name>`
- m** [`<0>`]
List group members in different formats.
 - 0 listing(default)
- V**
Version.
- X**
Terse.

ARGUMENTS

An optional list of scripts to be displayed only, if not provided all present scripts from directories within the variable `CTYS_SCRIPTS_PATH` are scanned and displayed. The given strings are just matched within the pathname of found script files, thus could be partial filenames.

EXIT-VALUES

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

8.19 ctys-vnetctl

SYNTAX

ctys-vnetctl

```

[-b <virtual-bridge>]
[-d <level>]
[-f]
[-g <sbit-group>*]
[-h]
[-H <help-options>]
[-i <interface>]
[-l <remote-user>]
[-r <remote-hosts>]
[-s <ALTERNATE-QEMUSOCK>]
[-S <ALTERNATE-QEMUMGMT>]
[-u <non-privileged-user>[.<group>]]
[-n]
[-V]
[-X]
[-Z <set-sudo-ksu>]
(cancel|check|create|info|ports|list|listall)

```

<remote-user>:=<user>[.<group>]

<remote-hosts>:=[<user>@]<host>[,<remote-hosts>]

DESCRIPTION

ctys-vnetctl creates and manages the complete set of Network-Devices required for interconnection by KVM and QEMU. The creation and deletion is performed by just one call. Therefore **ctys-vnetctl** encapsulates and combines the subset of functionality of required tools supporting the TAP/TUN devices by VDE .

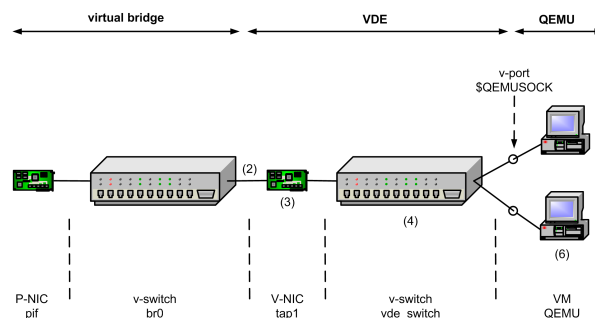


Figure 8.3: QEMU/KVM-Network Interconnection

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 'VDE Remote Configuration'.

- **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' the network is disconnected for a short time, so any access to NFS or any other networked file system is interrupted temporarily, which eventually leads to missing of additional tools required for call-completion, e.g. for reconnecting to the network.

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 conditions. Local-only users with "sudo" control by complete locally configured environments are verified to work stable.

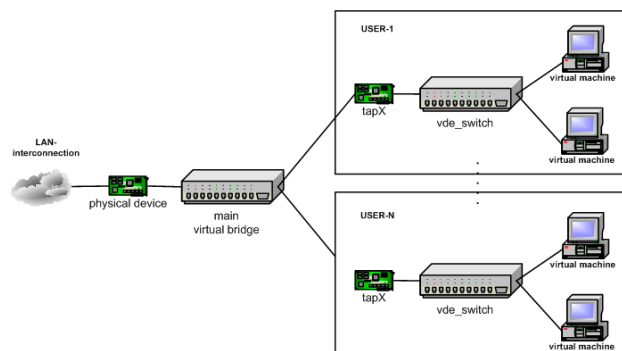


Figure 8.4: QEMU/KVM-Network Interconnection - Components

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. The current debian setup works the same way, where even the name of the first interface is reused as the name of the created bridge.

One reason for "doing" the bridge allocation this way within the Unified-SessionsManager is the minimized risk of detaching the remotely handled VMs for too long from the network services, which might make them unusable from then on. This aspect has to be emphasized due to the intention of frequent on-the-fly creation of networking devices. This naming and address-assignment concept will probably be modified slightly in future versions. Anyhow, the remote usage of "ctys-vnetctl", once the authenti-

cation is configured properly and security facilities are setup thoroughly, offers a simple interface for centralized setup of VM stacks . This is particularly true in combination of remote usage of GENMCONF and PLUGINS.

The usage of `ctys-vnetctl` assures the appropriate environment for the used of the wrappers "vdeq" and "vdeqemu" of the package VDE-SOURCEFORGE, which is the recommended tool when TAPTUNbyVDE has to be created. This utility could be used in any comparable case too, but fit particularly for QEMU setup.

The configuration files if QEMU are shared here, thus the consistency of QEMUSOCK is ssured. 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 combined within this script:

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

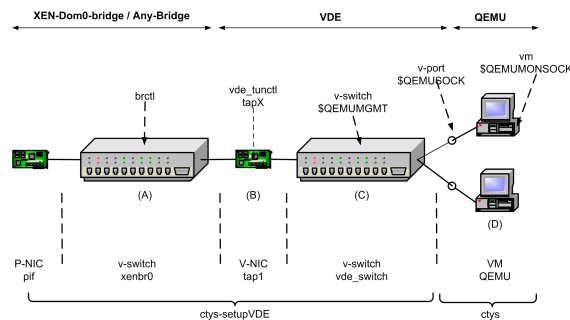


Figure 8.5: QEMU/KVM-Network Interconnection - Details

Two types of virtual bridges/switches(see figure:NestedProtocolStacks are managed by `ctys-vnetctl`.

- "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-vnetctl`.

TAPTUNbyVDE prepares a TAP device with a attached new bridge, therefore it requires the VirtualDistributedEthernet - sourceforgeVde package. Additional information within a Wiki containing some helpful tutorials for virtualsquare-basinet working could be found at the website of 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

ATTENTION:

The default bridge used is the first found in alphabetical order. For some default installations this might not be the intended. When DHCP is used the first to check in case of errors is the actual used bridge. The bridge to be used could be forced by the **-b** option.

The following steps are performed by ctys-vnetctl:

- (a) 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
```

- (b) 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'

(c) 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 verified with:

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

unixterm ${QEMUMGMT}
```

For additional information refer to examples of the manual.

OPTIONS

-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, then 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.

- 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.

- 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 preferably 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---"
```

-h

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

-H <help-option>

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

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

-i <interface>

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

-l <remote-user>

Refer to "ctys" generic options for additional information.

-r <remote-hosts>

List of remote hosts for execution. Either a list of valid hostnames, ipaddresses, or EMail-Format hostnames. Multiple entries could be provided in the following format:

```
<remote-hosts>:=[<user>@]<host>[,<remote-hosts>]
```

-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.

-u <non-privileged-user>[.<group>]

Owner of the created TAP device. Default is current user. In addition to the EMail style user name, here the full scope for common UNIX file access control is supported. This includes the optional group name.

```
<remote-user>:=<user>[.<group>]
```

-V

Version.

-X

See ctys, terse for machine output.

-Z

See ctys.

. ARGUMENTS

- <command>

- **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 etup for selected USER. Therefore a simple "ctys-vnetctl 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 de-finetly.
- **create** 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.

EXIT-VALUES

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

8.20 ctys-smbutil

SYNTAX

ctys-smbutil

```

[-d <debug-level>]
[-h]
[-H <help-options>]
[-i <numerical-IP>]
[-l <remote-login>]
[-n <dns-TCP>]
[-r [(REVERSE|R|-),]PING|SSH[,PM|VM] ]
[-V]
[-X]

```

DESCRIPTION

ctys-smbutil displays a short list of current available SMB server and workstations.

OPTIONS

-d <debug-level>

Same as ctys.

-h

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

-H <help-option>

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

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

-i <numerical-IP>

Show numerical TCP/IP-Address. This is supported in combination with "-n" option only without "-X" option.

l <login>

Account for remotecalls.

-n <dns-TCP>

Show TCP/IP-Address as name. This is supported in combination with "-i" option only without "-X" option.

-r <runtime states>

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

Only the possible targets which are operable or 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>=[(REVERSE|R|-),]PING|SSH[,PM|VM]`

- **REVERSE|R|-** This reverses the resulting set, thus the "not matching" states only will be shown.
- PING**[:<packetcnt>[%<timeout>]] A RUNNING session is simply "pinged". Resulting statement of operational mode may result on applied security policies, but not exclusively on the state of the running OS's IP-stack.

-R <remote-host>

The remote host for the inherent beamer function to place the execution immediately on an arbitrary execution relay.

-V

See ctys, version output.

-X

See ctys, terse for machine output.

. ARGUMENTS

. EXIT-VALUES

0: OK: Result is valid.

1: NOK: Erroneous parameters.

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

8.21 ctys-vboxutils

SYNTAX

ctys-vboxutils

```

[-d <debug-options>]
[-h]
[-H <help-options>]
[-n]
[-V]
[-X]
[
  (
    (FETCH|F) (
      VBOXPATH4ID|P4I
      |VBOXUUID4PATH|UUID
      |VBOXNAME|NAME
      |VBOXCFGFILE|CFGFILE
      |VMSTATE|STATE|VS
      |VMMAC|MAC
      |VMRDP|VR
      |VMRDPLST|VRL
    )
  )
  |
  (
    VMMANAGE (
      GETCFGFILE
    )
  )
  |
  (
    (LIST|L) (
      SERVERS_RAW
      |CLIENTS_RAW
    )
  )
]

```

DESCRIPTION

ctys-vboxutils is a collection of wrappers for miscellaneous function calls for display and conversion of inventory and machine data. The call could be executed on localhost or on a remote host with the tool 'ctys-beamer'.

OPTIONS

-d <debug-args>

Refer to "ctys" generic options for additional information.

-h

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

-H <help-option>

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

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

-V

Version.

-X

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

ARGUMENTS

The commands to be executed on the localhost or remote machine.

FETCH <subargs>

- **(VMWPATH4ID|P4I) (<ID>|<UUID>|<NAME>)** Transforms the provided object ID to a pathname.
- **(VBOXUUID4PATH|UUID) (<ID>|<UUID>|<NAME>)** Transforms the provided ID to a UUID.
- **(VBOXNAME|NAME) (<ID>|<UUID>|<NAME>)** Transforms the provided ID to a NAME.
- **(VBOXCFGFILE|CFGFILE) (<ID>|<UUID>|<NAME>)** Transforms the provided ID to a configuration(XML) file pathname.
- **(VMSTATE|STATE|VS) (<ID>|<UUID>|<NAME>)** Fetches the state.
- **(VMMAC|MAC) (<ID>|<UUID>|<NAME>)** Fetches the MAC address list.
- **(VMRDP|VR) (<ID>|<UUID>|<NAME>)** Fetches the RDP port.
- **(VMRDPLST|VRL) (<ID>|<UUID>|<NAME>)** Fetches the list of provided RDP ports.
- **CHECK|CHK <subargs>**
Checks whether the provided machine is already registered.

LIST <subargs>

Lists the requested entries in raw format for postprocessing.

EXIT-VALUES

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

8.22 ctys-vdbgen

SYNTAX

ctys-vdbgen

```

    [--append]
    [--base=<base-path-list>]
    [--cachedb=<output-db-directory-path>]
    [--filecontext]
    [--no-splitted]
    [--progress]
    [--progressall]
    [--replace]
    [--scan-all-states]
    [--splitted]
    [--splitted-keep-files]
    [--stdio]
    [--threads=<#max-number-of-parallel-scans>]

    [-b]
    [-c]
    [-C]
    [-d <debug-level>]
    [-h]
    [-H <help-options>]
    [-t]
    [-T]
    [-V]
    [-X]
    --
    <target-list>

```

DESCRIPTION

ctys-vdbgen generates databases containing inventory data and static generic address mapping for VMs and PMs. The data is collected from configuration files with additional data from proprietary vendor interfaces and stored for usage by **ctys-vhosts**. The collection of the inventory data is performed by scanning local and remote machines. The scan could be applied with by pre-defined constraints for scope and performance variation such as the operational states of the VMs and the specific assigned hypervisors, or restrictions on the search scope for filesystem locations limited to specific partitions or directory-subtrees. The scan once started is executed fully automated, just an operational installation of the `UnifiedSessionsManager` and access privileges for the scan-operator has to be pre-assured. The scanned database could be shared and utilized by numerous runtime components, e.g. virtual nameservices and graphical starters. The facilities scale up to thousands of entries even by the usage of a

spreadsheet compatible file database.

Dynamic entities without persistent data are not scanned into the database and are handled by their dynamic data only. Dynamic entities are for now CLI, RDP, VNC, and X11 instances.

The whole automated initial creation of the cacheDB takes some Minutes for scanning filesystems and presents than the actually available stock of VMs and PMs. The contained entities could be addressed from than on by various attributes. The call interfaces of all components provide for the option '-p' for the selection of alternative databases, e.g. for specific views or pre-validation of test-scans without influence on the production environment. The common database is shared by the whole set of utilities including commandline tools and graphical applications.

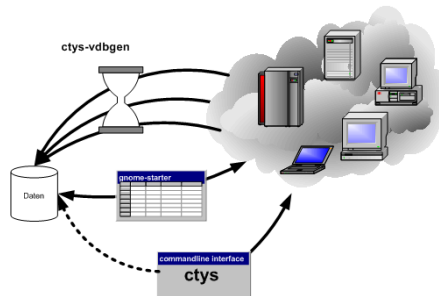


Figure 8.6: Database creation and application

The database contains the hypervisors KVM, QEMU, VBOX, VMW, and XEN, additionally PMs. For each entry also default login desktops or commandline applications are stored for automation of console interconnection as well as native logins - based on CLI, X11, VNC, and RDP.

ctys-vdbgen 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 <output-db-directory-path>, or "\$HOME/.ctys/db/default" respectively.

The mapping contains available VMs for a given list of PMs as well as the PMs. Multiple databases could be generated as specific access-groups which could be selected by the standard parameter **-p** for all ctys utilities with "cacheDB" access.

The scanned entities rely either on proprietary vendor files or interfaces, and/or on ctys specific configuration files containing extended GuestOS information. These documented ctys configuration files could be created by the utility **ctys-createConfVM** automatic, or interactive. Alternatively manual creation could be proceeded. These stored configuration could be in general imported into the UnifiedSessionsManager by the ENUMERATE action for dynamic runtime purposes as well as for storage within the database.

REMARKS:

- **Multiple-Entries of IP, and/or MAC addresses**

All entries are evaluated and checked for matching indexes. This means for a the index of an "IP0" an matching index of "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. This is due to a network-centric view on the interfaces, which together found a network and represent their hosts to the outside.

- **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!

This can be varied by option for some utilities.

- **Whitespaces**

Whitespaces are generally not 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 additionally checks for the PM configuration which is located in

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

For additional information refer to GENMCONF.

A special append-mode is supported for addition of data. This mode does a pure concatenation only, redundancy of added data is not 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.

ATTENTION:

When debian is accessed with an automounted filesystem the internal call of 'find' command does not force the mount including recognition of the complete contained filesystem, even worse, does this different call-by-call! So the work-around for now is either a fixed mount, or a 'ls -R' before the scan.

OPTIONS

ATTENTION:

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 containing 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:

- `"-cacheDB=<output-db-directory-path>"`
 - `DEFAULT_VDBGEN_DB=$HOME/.ctys/db/default`
 - `DEFAULT_DBPATHLST=$HOME/.ctys/db/default`
- or
- `"-cacheDB=<output-db-directory-path>"`
 - `DEFAULT_VDBGEN_DB=<install-path>/conf/db/default`
 - `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.

-h

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

-H <help-option>

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

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

-no-splitted

This deactivates the splitted mode and passes the whole parallel operations control to ctys controlled by the parameter **-b**

-progress

This activates the formatted table output, therefore some traces are filtered and displayed 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 intermediate result is displayed. The final sum-up data is additionally displayed as soon as the complete set of results for the current entity is available.

-replace

This activates the explicit replace mode, which is required when a "enum.fdb" already 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.

-scan-all-states

This option ignores the actual availability of an appropriate hypervisor and scans all present configurations. This is particularly helpful for test environments, where the hypervisor on a machine might be changed frequently.

-splitted

This provides for the by-target-and-by-session splitted parallel operations. This option utilizes the **-b seq,ser** option and establishes it's own background and parallel operations control. The number of parallel tasks is controlled by the environment variable **CTYS_VDBGEN_PARTARGETS** which is set to **DEFAULT=10**. In addition for each task a temporary cache file is created within the cacheDB directory and kept for later analysis. Thus e.g. in case of machine and plugin specific bug-tracing the specific results are available for fast analysis.

-splitted-keep-files

Keeps temporary files when in 'splitted' mode. These have to be removed manually once kept.

-stdio

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

-threads=<#max-number-of-parallel-scans>

Number of tasks to be scanned in parallel, applicable for 'splitted' mode only.

Default is **CTYS_VDBGEN_PARTARGETS=10**.

-t

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

-T

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

-V

Version.

-X

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

. ARGUMENTS

Any target to be enumerated, could be hosts and nested groups. Additional context arguments are supported, but should be added carefully, e.g. a non-applicable action like '-a CREATE' will lead to malfunction.

Current version supports lists of one or more intermixed host and groups without suboptions. Suboptions are provided as pre-positioned common options for all, or within group files as **CONTEXT**. The default scan options for the '-progress' indicator are set implicitly, when the option is selected.

. EXIT-VALUES

0: OK: Result is valid.

1: NOK: Erroneous parameters.

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

8.23 ctys-vhost

SYNTAX

ctys-vhost

```

[-c <spent cost on execution environment>
  =:(
    MINCNT|MAXCNT|CNT
  )
]
[-C <DB sources>
  =:(
    OFF
    |CLEARTMP
    |CLEARALL
    |GROUPS
    |KEEPALL
    |LIST
    |LISTCACHE
    |LISTTARGETS
    |LISTGROUPS
    |MEMBERSDB
    |MACMAPONLY
    |MACMAP
    |REBUILDSCACHE
  )
]
[-d <debug-level>]
[-h]
[-H <help-options>]
[-i <input-list>=[CTYSADDRESS|CTYS]]
[-I <0-9>]
[-l <USER>]
[-M <result-set-output-reduction>
  =:(
    FIRST|LAST|ALL|COMPLEMENT|SORT|USORT|UNIQUE
  )
]
[-o <output-list>
  =:(
    (
      (
        [ARCH] [,]
        [ACCELERATOR|ACCEL] [,]
        [CATEGORY|CAT] [,]
        [CONTEXTSTRING|CSTRG] [,]
        [CPORT|VNCPORT] [,]

```

```

[CTYSADDRESS|CTYS] [,]
[CTYSRELEASE] [,]
[DIST] [,]
[DISTREL] [,]
[EXECLOCATION] [,]
[EXEPATH|EXEP] [,]
[GATEWAY|GW] [,]
[GROUPID|GID] [,]
[HWCAP] [,]
[HWREQ] [,]
[HYPERREL|HYREL] [,]
[HYPERRELRUN|HYRELRUN] [,]
[IDS|ID|I] [,]
[IFNAME] [,]
[INDEX] [,]
[LABEL|L] [,]
[MAC|M] [,]
[NETMASK] [,]
[NETNAME] [,]
[OS|O] [,]
[OSREL] [,]
[PLATFORM|PFORM] [,]
[PM|HOST|H] [,]
[PNAME|P] [,]
[RELAY] [,]
[RELOCCAP] [,]
[SERIALNUMBER|SERNO] [,]
[SERVERACCESS|SPORT|S] [,]
[SSHPORT] [,]
[STACKCAP|SCAP] [,]
[STACKREQ|SREQ] [,]
[TCP|T] [,]
[TYPE|STYPE|ST] [,]
[USERSTRING|USTRG] [,]
[USERID|UID] [,]
[UUID|U] [,]
[VCPU] [,]
[VERSION|VERNO|VER] [,]
[VMSTATE|VSTAT] [,]
[VNCBASE] [,]
[VNCDISPLAY|DISP] [,]
[VRAM] [,]
)
[TITLE|TITLEIDX|TITLEIDXASC] [,]
[MACHINE|MAXKEY] [,]
)
| (TAB_GEN|TAB) [:<tab-args>]
| (REC_GEN|REC) [:<tab-args>]
| (SPEC_GEN|SPEC) [:<tab-args>]
| (XML_GEN|XML) [:<tab-args>]
) [,]

```

```

    [IP|DNS|D] [,]
    [SORT[:ALL|A|UNIQUE|U|REVERSE|R|EACH
        |([0-9][0-9][0-9]| [0-9][0-9]| [0-9])]
        [%]] [,]
]
[-p <db-directory-path-list>]
[-r]
[-s]
[-S <BasicDataManagementSupport>
    =:(
        CONTENTGROUP
        |LISTALL
        |LIST
        |LISTDB
        |MEMBERSDB
        |LISTGROUP
        |MEMBERSGROUP([2345678])|([678]u)[:<groups-list>]
    )
]
[-T <type-list>]
[-V]
[-R <runtime states>
    =:(
        [MARK|(REVERSE|R|-),] (PING|SSH) [, (PM|VM)]
    )
]
[-X]
<awk-regexp>[
    (
        AND
        |E:<#field0>:<#field1>
        |F:<#field>:<string-literal>
        |NOT
        |OR
        |<awk-regexp>
    )
]

```

DESCRIPTION

is the basic address resolution interface for runtime execution of commands based on ctys addressing ctys-vhost . The main task of this tool is to support a scripting interface for the functional link between VMs and PM, and their contained OS. Even though a considerable amount of functionality is provided, the user interface is designed in a manner for simplified application of the common daily tasks. The basic queries just require actually a few arguments, e.g.

```
ctys lab02 tst320
```

displays by default the record for the VM 'tst320' registered for execution on the PM 'lab03'. The following query

```
ctys lab02 tst32
```

displays on the PM 'lab02' all VMs with the pattern match '*tst32*', whereas

```
ctys lab02 tst3.0
```

displaye the pattern match '*tst3.0*' - where the '.' matches any character. When required a more detailed set of constraints on a query could be applied as required.

The 'ctys-vhost' utility combines functionality of:

- Hostname resolution for physical and virtual machines, including off-line machines for a pre-start queries.
- Address conversion into and from the extended syntax **<machine-address>** for network wide-unique addressing of VMs, PMs, and additionally HOSTs/Login-Sessions.
- Dynamic availability checks including SSH-Logins by **ctys**.
- Inventory management front-end functions for various attributes generated by **ctys-vdbgen**.
- Load balancing.
- Integration of DHCP data from **ctys-extractMAClst**, **ctys-extractARPlst**, and **ctys-macmap**.
- Import- and Export- functionality by various formats.
- Handling of GROUPS objects in combination with **ctys-groups**.
- Creation and refresh of cached pre-resolutions for network data.
- Handling of multiple databases as specific **views**.

The basic similarity to the UNIX 'host' function for name resolution to virtual machines is expanded by 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. This implies some dynamic data handling due to frequently changing 'physical containment positions' of roaming VMs. Therefore a basic load-balancing is included, which is currently still a pre-release.

Due to targeted simplicity and efficiency the address resolution is based on a combination of RECORD based and unstructured flat pattern matching by regular expressions only. Even though the record is structured by fields as given by '-a ENUMERATE=MACHINE', the match itself could be performed quite reliable and performant as a simple regexpr given by an '**awk-match**' string expression to be applied to RECORD-STRING, whereas the output is still displayed on the level of records and fields. Combined multiple-regexprs are supported and are iteratively applied on the intermediary results of each step. The set of regular expression could be either combined by AND or OR operators, where AND is the default behaviour. This opens for more complex queries. The behaviour of the operator preference is described in detail within the following **ARGUMENTS** section. The filtering of the output as requested by the '-o' option is performed on the final set of resulting records only. This could be applied for example, when a table display has to be sorted, where the option '-o SORT:1' sorts the resulting table by the first column. The call with

the option '-o sort:1,reverse' results in reverse sorting of the first column.

This simplified pattern matching approach is applicable when the given set of search attributes result in a unique match. When ambiguity occurs due to multiple matching records, the set of records are handled as defined by the '-M' option. Ambiguity occurs frequently when using a reduced set of selection attributes on a database for VMs in a networked environment. This is due to the frequently desired availability of multiple redundant access targets for the physically identical VM. Either the specification of additional attributes, or the application of load-distribution policies will resolve ambiguity. In current release a basic COST option '-C' extends the '-M' option for some usable load distribution within execution-groups.

In most other cases uniqueness should be given, e.g. the key UUID is defined to be applied as unique, but could be tampered unintentionally e.g. by co-allocated backups. For avoiding of backup-access the '-M first' option might be helpful, when the name of the backup is an extension. The more reliable approach here is to set the **VMSTATE** to **BACKUP** and using the **VMSTAT** filtering attribute for the **ENUMERATE** action when collecting the inventory data.

The current implementation enables the management of up to multiple thousands of VMs with the given simplicity with single-query responses in the range of about 0.6-0.8 seconds (measured, but of course depends on machine etc.). One of the next versions will additionally support LDAP based nameservices, targeting an enterprise environment with a number of almost an unlimited amount of distributed services to be managed.

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

The address resolution is performed by the following steps :

- Check the static list of given ExecGroup for possible candidates.
- List the active sessions on the given ExecGroup.
- Take the appropriate PM/VM by utilising '-C' and/or '-M' option.

The following list shows some examples of using ctys-vhost as an interactive query tool.

- Given partial strings, e.g. '192.168.1' lists all machines of that subnet. When the '-M active' option is chosen, all currently active sessions within that subnet are listed.
- Any string could be used as partial pattern, e.g. parts of MAC-Ids or fragments of UUIDs. The given string will be matched against complete record, mostly an awk-regex, thus any part, even spanning multiple FIELDS could be used. But currently not regex, 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 or reduced applied sets.
- 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 for the selection or 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. These are `AND`, `OR`, `NOT`, `E:`, and `F:`, described within the section related to the **ARGUMENTS**.

OPTIONS

-c <spent cost on execution environment>

Cost as for load distribution when selecting a target. Companion options apply to resulting set of equal cost.

<spent cost on execution environment>=(MINCNT|MAXCNT|CNT)

- **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.

-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 pre-processed 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

Enumeration 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 enumeration 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 its MACHINE format entry from the staticCacheDB.

Each group is cached within a file by its own, thus the access could be performed by just one file-selection for the complete nested resolution of its entities.

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

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

- **ADJUST**
clears enum.fdb from redundant records of multiple scans. This is not necessarily required because the **REBUILDCACHE** clears redundant records before importing the current set.
- **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:

- 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).

- 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.

Another 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. Redundant record from multiple-rescans are cleared.

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

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

-H <help-option>

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

For additional help refer to the documents or type **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 performance 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 <login-name>

The users, which will be used for hosts without an explicitly 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 <USER>

Remote user to be used for SSH-access-checks, when the "-x" option is activated.

DEFAULT=CTYS_NETACCOUNT(DEFAULT->USER)

-M <record-filter>

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

<result-set-output-reduction>
=(FIRST|LAST|ALL|COMPLEMENT|SORT|USORT|UNIQUE)

- **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 handled carefully.

```
<output-list>=
(
  (
    (
      [ARCH] [,]
      [ACCELERATOR|ACCEL] [,]
      [CATEGORY|CAT] [,]
      [CONTEXTSTRING|CSTRG] [,]
      [CPORT|VNCPORT] [,]
      [CTYSADDRESS|CTYS] [,]
      [CTYSRELEASE] [,]
      [DIST] [,]
      [DISTREL] [,]
      [EXECLOCATION] [,]
      [EXEPATH|EXEP] [,]
      [GATEWAY|GW] [,]
      [GROUPID|GID] [,]
      [HWCAP] [,]
      [HWREQ] [,]
      [HYPERREL|HYREL] [,]
      [HYPERRELRUN|HYRELRUN] [,]
      [IDS|ID|I] [,]
      [IFNAME] [,]
      [INDEX] [,]
      [LABEL|L] [,]
      [MAC|M] [,]
      [NETMASK] [,]
      [NETNAME] [,]
      [OS|O] [,]
      [OSREL] [,]
      [PLATFORM|PFORM] [,]
      [PM|HOST|H] [,]
      [PNAME|P] [,]
      [RELAY] [,]
      [RELOCCAP] [,]
      [SERIALNUMBER|SERNO] [,]
      [SERVERACCESS|SPORT|S] [,]
      [SSHPORT] [,]
      [STACKCAP|SCAP] [,]
      [STACKREQ|SREQ] [,]
      [TCP|T] [,]
      [TYPE|STYPE|ST] [,]
      [USERSTRING|USTRG] [,]
      [USERID|UID] [,]
      [UUID|U] [,]
      [VCPU] [,]
      [VERSION|VERNO|VER] [,]
```

```

    [VMSTATE|VSTAT] [,]
    [VNCBASE] [,]
    [VNCDISPLAY|DISP] [,]
    [VRAM] [,]
  )
  [TITLE|TITLEIDX|TITLEIDXASC] [,]
  [MACHINE|MAXKEY] [,]
)
| TAB_GEN[:<tab-args>]
| REC_GEN|REC[:<tab-args>]
| SPEC_GEN|SPEC[:<tab-args>]
| XML_GEN|XML[:<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.

- **ACCELERATOR**

The accelerator as configured.

- **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.

- **EXEPATH**

The location of executable for starting the VM.

- **GATEWAY**

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

- **GROUPID**

The group id of user that created this entry.

- **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**

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

- **HYPERRELRUN**

The release of the present hypervisor when this record was created.

- **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.

- **INDEX**
The index of the record within the current snapshot of the selected database. This is a transitive value, which may change for each database change.
- **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 post-processing.
- **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 single hosts filesystem.
- **REC_GEN|REC**
Generates output format as structured proprietary record format, which is foreseen for online validation as well as automated postprocessing. Refer to common format of generic tables for additional information.
- **RELAY**
The relay interface, device, virtual bridge, virtual switch, or virtual hub, the VM is interconnected too within its 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 of the target.

- **SERIALNUMBER|SERNO**
An arbitrary serial number for the VM stored in the configuration file. This number should be unambiguous.
- **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!
- **SPEC_GEN|SPEC**
Generates table output format where each attribute is on a separate line. This format is particularly forseen to check values with tight reference to the documentation. Refer to common format of generic tables for additional information.
- **SSHPORT**
A list of provided SSH ports on this interface. Currently supported for OpenSSH only.
- **SORT[:<sort-args>]**
Enables the post-sort filter.

 <sort-args>=[ALL|A|EACH] [% (UNIQUE|U)] [% (REVERSE|R)] [%<sort-key>]
 - **REVERSE|R** Activates a final filter for call of "sort -r".
 - **UNIQUE|U** 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
- **TAB_GEN|TAB**
Generates table output format. The default format could be and is configured as a custom macro. Refer to common format of generic tables 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 separate entry is generated, thus it is listed as a separate 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.

- **TITLEIDX**
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 editable/customizable string from the user.
 - **UID**
User ID created this record.
 - **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 supported 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,1 -M all '59\[0-9\]\[0-9\]'
```


Where a standard baseport of 5900 is assumed.
 - **VRAM**
The amount of pre-assigned VRAM.
 - **XML_GEN|XML**
Generates output format in XML format for ease of postprocessing. Refer to common format of generic tables for additional information.
- p** <db-directory-path-list>
Comma separated 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** <remote-hosts>

-R <remote-hosts>

List of remote hosts for execution. Either a list of valid hostnames, ipaddresses, or EMail-Format hostnames.

-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 applies 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.

-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 <BasicDataManagement>

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 handled by the ctys-vhost command as appropriate.

```
<BasicManagementSupport>=
(
  LISTALL
  |LIST
  |LISTDB
  |MEMBERSDB
  |LISTGROUP[:<groups-list>]
  |MEMBERSGROUP([2345678])|([678]u)[:<groups-list>]
)
```

```
<groups-list>:=<group>[,<groups-list>]
```

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[:<group-list>]**
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.
- **MEMBERSGROUP[2-8][:<group-list>]** The
values MEMBERSGROUP[2-5] display the same as MEMBERSGROUP, but with a slightly different output format, whereas MEMBERSGROUP[6-8] shows the contained accounts or hosts.
 - MEMBERSGROUP2 - List of resulting targets with specific context options, for screen display.
 - MEMBERSGROUP3 - Similar to MEMBERSGROUP2, but additionally formatted with line-breaks.
 - MEMBERSGROUP4 - List of resulting targets with specific context options, prepared for inclusion by cut-and-paste.

- MEMBERSGROUP5 - List of resulting calls prepared for usage by cut-and-paste.
- MEMBERSGROUP6 - List of accounts in EMail format for screen display only.
- MEMBERSGROUP6u - Unique list of MEMBERSGROUP6.
- MEMBERSGROUP7 - List of accounts in EMail format for machine processing.
- MEMBERSGROUP7u - Unique list of MEMBERSGROUP7.
- MEMBERSGROUP8 - List of hosts for machine processing.
- MEMBERSGROUP8u - Unique list of MEMBERSGROUP8.

-T <type-list>

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

-V

Shows version.

-x <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 are formatted with the 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 accesibility is defined as the accesibility of the running OS on PM.

- **VM**

Checks VM for accessibility, this is particularly related to the SSH key. VM accessibility is defined as the accessibility of the running OS on VM.

- X

See `ctys`, `terse` for machine output.

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

ARGUMENTS

Use `"-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 advance 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 chosen thoroughly. The first is to suppress shell-expansion for the characters to be passed to the internal awk-expression. These might be required in rare cases only, but offer some advantage.

Some Examples:

- `.` (a single dot)
All items within the database, which is default when missing at all.
- `inst`
All items which contain any string "inst"
- `^inst`
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'.

The next point to be aware of is the order of precedence for the logical operators AND and OR. This is evaluated from **left-to-right**, the **precedence is inverted**. This means, that lower precedences are grouped together by virtual braces. This academic sounding remark offers tremendous positive practical impacts. First the theory - as a result the following rules are identical.

`x AND a OR b OR c`

is equal to

a OR b OR c AND x

This could be written as

x AND a OR x AND b OR x AND c

Current version does not support braces, but if - the previous could be written as

x AND (a OR b OR c)

The next consequence is the repetition of the rules when a higher valued operator AND follows a lower valued group.

x AND a OR b OR c AND y

This again could be written as

x AND (a OR b OR c) AND y

Or better as

(x AND (a OR b OR c)) AND y

This is consequently equal to

y AND (x AND (a OR b OR c))

Which could be written as

y AND x AND (a OR b OR c)

The main advance of the previous theory is the tremendous simplification and size-reduction of vast powerful and fast scanners. Thus within just a few lines a complete scanner with a recursive logic could be implemented as present within the 'ctys-vhost' command.

The call

```
ctys-vhost -o sort:1 \
    tst00 and not f:2:PM and tst103 or tst00 or tst00
```

or

```
ctys-vhost -o sort:1 \
    not f:2:PM and tst103 or tst00 or tst00 and tst00
```

selects the following set

label	styp	accel	dist	distrel	os	osrel	PM	if	TCP
tst003	VMW		SuSE	9.3	Linux	2.6	delphi.soho	0	172.20.2.133
tst003	VMW		SuSE	9.3	Linux	2.6	delphi.soho	0	172.20.2.133
tst005	VBOX	HVM			Windo		lab02.soho		
tst005	VBOX	HVM			Windo		lab02.soho		
tst005	VBOX						lab02.soho		
tst005	VBOX						lab02.soho		

from a database with 837 entries.

The actual queries could be visualized for analysis purposes. Therefore the intermediate metadata and the resulting matched records are displayed by the option '-I 2'.

The first call produces the output

```
time ctys-vhost -I 2 -o sort:1 \
  tst00 and not f:2:PM and tst103 or tst00 or tst00

START R-Methods
CHECK                =/homen/acue/.ctys/db/default
START R/W-Methods
RM RTCACHE           =/tmp/ctys.acue/ctys-vhost.20101106184019.cdb
QUERY:cacheDB repetitive:"tst00"
  ....!!!!X!!!!!!!!.x.....x.....x.....x50
  .....x.....x.....x.....x.....x100
  .....x.....x.....x.....x.....x150
  .....x.....x.....x.....!!!!X!!!!!!!!.x200
  .....x.....x.....x.....x.....x250
  .....x.....x.....x.....x.....x300
  .....x.....x.....!!!!!!X.....x.....x350
  .....x.....x.....x.....x.....x400
  .....x.....x.....x.....x.....x450
  !.....x.....x.....x.....x.....x500
  !!!!!!!!!X!!!!!!!!X!.....x.....x.....x550
  .....x.....x.....x.....x.....x600
  .....x.....x.....x.....x.....x650
  .....!!x.....x.....x.....x.....x700
  .....x.....x.....x!.....x.....x750
  .....x.....x.....x!!!!!!..x.....x800
  .....x.....x.....x.....
  match=68 of total=835
QUERY:cacheDB repetitive:"f:2:PM"
  .....x.....x.....x.....!..x.....x50
  .....!X.....
  match=3 of total=68
QUERY:cacheDB repetitive:"tst103"
  ...
  match=0 of total=3
QUERY:cacheDB repetitive:"tst00"
  !!!
  match=3 of total=3
QUERY:cacheDB repetitive:""
  !!!
  match=3 of total=3
QUERY:cacheDB repetitive:"tst00"
  !!!
  match=3 of total=3
QUERY:cacheDB repetitive:""
  !!!!!
  match=6 of total=6
```



```

!!!!!!
  match=6 of total=6
label |styp|accel|dist|distrel|os  |osrel|PM          |if|TCP
-----+-----+-----+-----+-----+-----+-----+-----+-----+-----
tst003|VMW |      |SuSE|9.3   |Linux|2.6   |delphi.soho|0 |172.20.2.133
tst003|VMW |      |SuSE|9.3   |Linux|2.6   |delphi.soho|0 |172.20.2.133
tst005|VBOX|HVM  |      |      |Windo|      |lab02.soho | |
tst005|VBOX|HVM  |      |      |Windo|      |lab02.soho | |
tst005|VBOX|      |      |      |      |lab02.soho | |
tst005|VBOX|      |      |      |      |lab02.soho | |

real 0m1.176s
user 0m0.696s
sys 0m0.532s

```

which requires less steps due to early AND reduction of the metadata. The second call requires much more steps, this is due to the late AND reduction of the metadata, which causes any OR part to be AND checked on the whole initial set.

```

time ctys-vhost -I 2 -o sort:1 \
  not f:2:PM and tst103 or tst00 or tst00 and tst00

START R-Methods
CHECK                =/homen/acue/.ctys/db/default
START R/W-Methods
RM RTCACHE           =/tmp/ctys.acue/ctys-vhost.20101106184201.cdb
QUERY:cachedB repetitive:"f:2:PM"
  .....x.....x.....x.....x.....x50
  .....x.....x.....x.....x.....x!!!!!!X100
  !!!!!!!!!X!!!!!!!!X!!!!!!!!X!!!!!!!!X!!!!!!!!X150
  !!!!!!!!!X!!!!!!!!X!!!!!!!!X!!.....x.....x200
  .....x.....x.....x.....x.....x250
  .....x.....x.....!!!!!!X!!!!!!!!X!!!!!!!!X300
  !!!!!!!!!X!!!!!!!!.x.....x.....x.....x350
  .....x.....x.....x.....x.....!!!!!!X400
  !!!!!!!!!X!!!!!!!!X!!!!!!!!X!!!!!!!!X!!!!!!!!X450
  !!!!!!!!!X!!!!!!!!X!!!!!!!!X!!!!!!!!X!!.....x500
  .....x.....x.....x.....x.....x550
  .....x.....x.....x.....x.....x600
  .....x.....x.....x.....!!!!!!X!!!!!!!!X650
  !!!!!!!!!X!!!!!!!!X!!!!!!!!X!!!!!!!!X!!!!!!!!X700
  !!!!!!!!!X!!!!!!!!X!!!!.x.....x.....x750
  .....!!X!!!!!!!!X!!!!!!!!.x.....x.....x800
  .....x.....x.....x.....
  match=349 of total=835
QUERY:cachedB repetitive:"tst103"
  .....x.....x.....x.....x.....x50
  .....x.....x.....x.....x.....x100
  .....x.....x.....x.....x.....x150
  .....x.....x.....x.....x.....!.....x200

```

```

.....x.....x.....x.....x.....x250
.....x.....x.....x.....x.....x300
.....x.....x.....x.....x.....
match=1 of total=349
QUERY:cacheDB repetitive:"tst00"
.....x.....x.....x.....x.....x50
.....x.....x.....x.....x.....x100
.....x.....x.....x.....x.....x150
.....x.....x.....x.....x..!.....x200
.....x.....x.....x.....x.....x250
.....x!!.....x.....x.....x.....x300
.....x.....x.....x.....x.....
match=3 of total=349
QUERY:cacheDB repetitive:"tst00"
.....x.....x.....x.....x.....x50
.....x.....x.....x.....x.....x100
.....x.....x.....x.....x.....x150
.....x.....x.....x.....x..!.....x200
.....x.....x.....x.....x.....x250
.....x!!.....x.....x.....x.....x300
.....x.....x.....x.....x.....
match=3 of total=349
QUERY:cacheDB repetitive:"tst00"
.!!!!!!
match=6 of total=7
QUERY:cacheDB repetitive:""
!!!!!!
match=6 of total=6
!!!!!!
match=6 of total=6
label |styp|accel|dist|distrel|os    |osrel|PM          |if|TCP
-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----
tst003|VMW |      |SuSE|9.3    |Linux|2.6   |delphi.soho|0 |172.20.2.133
tst003|VMW |      |SuSE|9.3    |Linux|2.6   |delphi.soho|0 |172.20.2.133
tst005|VBOX|HVM  |     |        |Windo|        |lab02.soho |  |
tst005|VBOX|HVM  |     |        |Windo|        |lab02.soho |  |
tst005|VBOX|     |     |        |     |        |lab02.soho |  |
tst005|VBOX|     |     |        |     |        |lab02.soho |  |

real 0m1.256s
user 0m0.836s
sys 0m0.480s

```

Another effect which could be seen here is the almost equal processing duration, which shows the startup and display overhead as the dominating factor. The times are in the same range for half of the data as shown for 835 entries.

The seemingly double-display is due to the multiple OR in this inverted-precedence semantics, which is resulting in additional display.

The seemingly double-display is due to the hidden display of the owner, here one is the user 'root.root', the other is 'acue.ldapusers'.

For the following rules and operators the search-and-match strings are case sensitive, the operators are not.

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 match is **checked literally**, this means that 'pm' is NOT EQUAL 'PM'.

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.

8.24 ctys-vmw2utils

SYNTAX

ctys-vmw2utils

```

[-d <debug-options>]
[-h]
[-H <help-options>]
[-n]
[-V]
[-X]
[
  (
    (FETCH|F) (
      VMWPATH4OBJID|P40
      |VMWOBJID4PATH|O4P
      |DATASTORE|D
    )
  )
  |
  (
    (CONVERT|C) (TODATASTORE|2D)
  )
  |
  (
    (LIST|L) (
      INVENTORY|I
      |DATASTORES|D
    )
  )
]

```

DESCRIPTION

ctys-vmw2utils is a collection of function calls for display and conversion of Inventory data form VMWare Server 2.x(TM). The call could be executed on localhost or on a remote host with the tool 'ctys-beamer'.

OPTIONS

-d <debug-args>

Refer to "ctys" generic options for additional information.

-h

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

-H <help-option>

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

installed manuals.

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

-V

Version.

-X

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

ARGUMENTS

The commands to be executed on the localhost or remote machine.

CONVERT <subargs>

- **TODATASTORE <pathname>**
Converts a pathname to a datastore.

FETCH <subargs>

- **VMWPATH4OBJID <OID>**
Transforms the provided object ID to a pathname.
- **VMWOBJID4PATH <pathname>**
Transforms the provided pathname to an object ID.
- **DATASTORE <pathname>**
Displays the datastore for a given pathname.

LIST <subargs>

Lists the requested entries.

- **INVENTORY**
Lists the contents of the inventory as defined by the variable:
VMW_INVENTORY_DEFAULT=
{VMW_INVENTORY_DEFAULT:~/etc/vmware/hostd/vmInventory.xml}
- **DATASTORES**
Lists the present datastores.
VMW_DATASTORE_DEFAULT=
{VMW_DATASTORE_DEFAULT:~/etc/vmware/hostd/datastores.xml}

EXIT-VALUES

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

Chapter 9

CTYS Call interface

9.1 Common Elements and Behaviour

9.1.1 Sessions Namebinding

The current version binds the session to its plugin type as an internal selector for required runtime components. The data identifying a specific session is required for the identification of the individual instances.

The combination of the plugin type and the session identifier selects a specific session and the related runtime components.

Even though the key pair allows the usage of address elements within multiple session types, this should be avoided.

9.1.2 Cache for Performance and Collections

The name resolution for static and dynamic data could be performed basically by two approaches:

- from cached data by ctys-vhost and related tools
- by polling of actual runtime data with "ctys -a LIST" call.

Both approaches have their advances and impacts as any cache system has.

When using the cache, this obviously has to be up to date, when a data match occurs. When no match could be get, the realtime polling will be used by default.

The performance impact of the realtime poll vs. cached data is frequently a factor more than 10, which is 5-15 seconds vs. 0.5 seconds. The usual case is even more dramatic, usually 30 to 60 seconds, for CREATE, could be even much longer, depending from the file system to be scanned.

In cases of runtime systems scan via LIST it depends on the actual load. This approach is normally somewhat different from the pure CREATE call, because just a part of the data could be used from the cache DB, which just represents "potential candidates", the actual runtime state has to be polled in any

case, but could be done much more efficient of course.

But anyhow, in specific situations the usage of the poll approach is the only one applicable. The systems bootstrap for example is normally performed with an

empty cache DB, so no decision is required. This is particularly the case, due to the usage of ENUMERATE itself for evaluation and collection of the cache DB.

The cache behaviour could be commonly controlled by the following two keyword:

NOCACHE

Disables the usage of cached data for address resolution. Therefore the call of "ctys-vhost" and related utilities is disabled.

The default behaviour is to used CACHE first, which resolves for the low-medium reference system frequently within less than 0.5 seconds to a match. When no match found POLL is used, which is an internal ENUMERATE call and may have an performance impact of multiples of 10 and even more, depending on the size and structure of the scanned file system. The response time is frequently in the range of 30-60 seconds, and more.

When activated, at least polling has to be allowed, otherwise no resolution is be possible.

NOPOLL

Disables the polling of data, thus CACHE is required. A missing hit on cached data will lead to failing of resolution.

9.1.3 Access VM Stacks

The utilization of VM stacks, which are nested and therefore encapsulate each level requires for several methods some specific recognition and stack-awareness.

Almost each action requires it's own specific handling.

CREATE

The CREATE action is almost only effected specifically, when a stack-level is requested to be activated, for which the containing (n-1)-level stack entity is not yet present, this statement includes any lower missing entity too. Therefore it has to be decided whether the request should be rejected or an implicit stack-recovery has to be performed. When the decision is made for implicit recovery, the algorithm could be said to

"process-upward-from-highest-existent".

CANCEL

The CANCEL action is effected by stack-awareness when a stack entity is requested to be canceled, which has active contained upper stack entities. In this case the decision to be made are one or a combination of

- propagate CANCEL request to upper stack
- use hypervisor only, ignore upper stack
- "kill" if supported (e.g. not Xen) the current level hypervisor
- CANCEL the execution target, which is the execution level of stack "itself", e.g. essential for PMs.

When the decision is made for state-change-forward-propagation, the algorithm could be said to

- (a) "walk-upward-to-all-highest-existent"
- (b) "process-downward-to-request-target"

LIST

This action has to be stack-aware on request, it is the most beneficial dynamic information collector for the user in order to manage VM stacks.

SHOW

This action should be stack-aware on request, it is a beneficial dynamic information collector for the user in order to manage VM stacks.

ENUMERATE

This action must not be stack aware, what is due to the practical circumstance, that the destination VMs has to be active, when to be enumerated for the executable VMs. Thus, as long, as not the offline VM images could be scanned (which will be obviously never the case for all VMs), this call would have lead to the activation of ALL VMs executable, thus it probably could be said, will never finish due to bulk-start-performance-impact!

INFO

Basically almost the same as ENUMERATE.

Due to the impact of stack awareness for execution, this behaviour could be controlled for the methods. Almost each action requires it's own specific handling.

FORCE

Performs the requested action immediately by calling the hypervisor. No forward propagation is performed. The specific range of features for the various hypervisors could and do vary, thus will be specialized within the description of the specific plugin.

STACK

Performs the requested action first by state-propagation "deeper into the nested stack", which means upward due to ISO-like stack model. The behaviour when failing with state-change-request propagation depends on the actual request.

The hypervisor might be called as final anchor in case the propagation fails or timeouts for a CANCEL request, because otherwise the request may fail, even though it could have been finished by usage of the hypervisor. A decision whether or not to perform a "final and ultimate kill" on the stack is not obvious due to the hidden upper part, which basically could include any highly sensitive task, resulting in crucial failure when canceled abrupt. This risk will remain for generic actions, and has to be handled by the application and it's close systems within the encapsulated stack entities.

For the CREATE action - which is in case of STACK an iterative upward execution - the call of the hypervisor is necessarily the first call on each level. Thus in case of CREATE in contrast to the CANCEL action the transparency of the stack is system immanent. The call of the hypervisor could be followed by an state-change-propagation request after success only. The remaining question for partial execution is here what to do with an incomplete VM stack.

The answer for now is "let him live". This has to be monitored and handled appropriately by the user when the "-b on" option for asynchronous background operations is chosen. In case of synchronous operations the job will be canceled, but also no roll-back is performed. Just the monitoring is implied due to synchronicity. The dynamic information poll methods LIST and SHOW just ignore a propagation failure, this is the same behaviour as in the case that nothing was found.

SELF

The SELF keyword is currently for CANCEL action used only, but will of such basic level of understanding, that it is shown here. SELF causes the inclusion of the execution target for requested operations too.

This is due to the technical requirement, that any method for an stacked entity has to be initially performed within it's container entity, which hosts the bottom controller instance of the targeted stack-level - it's hypervisor - for ultimate superposing action. The basic call structure

```
ctys -a <action>=<action-target> <execution-target>
```

of ctys is splitted into the <execution-target> where the first communications peer entity resides, and the <action-target>, which is the destination of <action> to be performed.

The Stack Essentials - CREATE and CANCEL

One of the most beneficial features for stacked VMs would be the implicit creation of missing sessions within the execution stack. Anyhow, even though it is basically possible to do this with some current VM implementations, this feature is shifted.

The reason is simply the lack of a common strategy within the used VMs, where at least XEN and QEMU would be suitable, but others not for an unattended and SERVERONLY iterative CREATE chain. In avoidance of a quick-shot resulting in some "savage-variants" when coming to details, the decision for this version is not to implement the automatic and implicit creation of inactive intermediate stack entities.

Therefore this feature is currently shifted and will be implemented within a later version. For now each stack entity has to be CREATED seperately and explicitly by the user, in manual iteration. For CANCEL anything stays as it is, stacked operations are supported as described.

9.2 ctys

ctys

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

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

9.2.1 Installation and Setup

The installation and configuration is described 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.

9.2.2 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.

9.2.3 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.

9.2.4 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.

9.3 OPTIONS

9.3.1 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**

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>|-)] [%<domain>]]]
)

```

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>|-)][%<domain>]]

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

CREATE

```

CREATE=<machine-address>{1}
[
  CONNECT
  |REUSE
  |RECONNECT
  |RESUME
] [,]
[USER:<user>[%[(<credentials>|-)] [%<domain>]]] [,]
[CONSOLE:<console-type>] [,]
[BOOTMODE:<boot-mode> [%<boot-img-pathname>]] [,]
[(CHDIR|CD):<working-directory-change-to>] [,]
[INSTMODE:
  <boot-mode>%<inst-boot-source>%<insttargetmode>%<insttarget>%<inststa>
  [,]
[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 described 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 plugging. 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 UnifiedSessionsManager 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 9.1: 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 9.2: Supported Install-Targets

<boot-img-pathname>

Optional pathname to an alternative 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 accessible 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 9.3: 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 9.4: 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 pre-configured 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 during 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 and 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 separate 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>|-)][%<domain>]]

The account to be used for native access to the <action-target>. In addition to the user name and credentials an optional domain name for the user account - e.g. fo MS-Windows(TM) based accouts - could be provided. Missing parameters are requested during authentication.

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.

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 ad-

ditionally 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 templates 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 statically 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 separate 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 separated 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 9.5: 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 capabilities supported.	-
22	STACKREQ	The list of capabilities 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-CPU's.	-
42	CONTEXTSTRG	A private context storage for the plugin	-
43	USERSTRING	A string to be customized by the user.	-

Table 9.6: Output Record-Format for MACHINE suboption

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"
```

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 9.1: INFO Display

```

LIST
LIST[=
  (
    (
      (
        [ACCELERATOR|ACCEL] [,]
        [ARCH] [,]
        [CONTEXTSTRG|CSTRG] [,]
        [CPORT] [,]
        [DISPLAY] [,]
        [EXECPTH|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>|-)] [%<domain>]]] [,]
]

```

LIST displays 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-persistent 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 performance 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 separated 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 separator, 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 have 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 9.7: 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 9.8: 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 separated 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.

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 9.2: LIST Display

9.3.2 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 of 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 DETACHED 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 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.

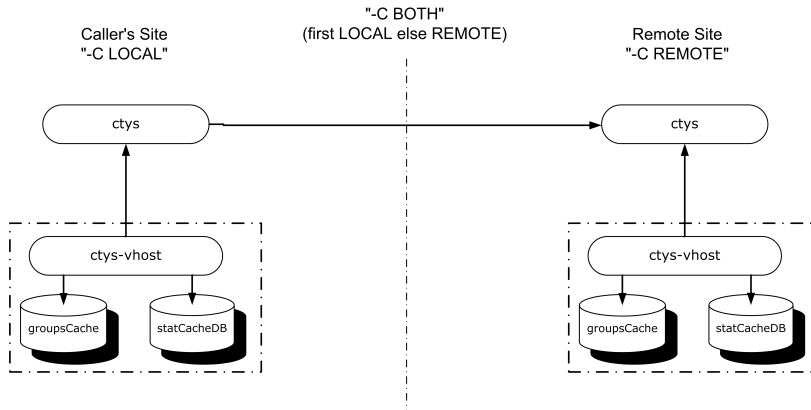


Figure 9.3: Distributed Caches

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 its 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 its 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 handled 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 on 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 scheduler, 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 caching, 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 post-processed 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#1000000000000000=16#4000** Traces sync-points for regression tests.
- **D_BULK=32768=2#10000000000000000=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 reasons 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 separate 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 man-pages, 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 explicitly 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>

Restricts/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 they 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 these 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, at least 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 preferred.

- **OFF**

No remote resolution is performed. The application should provide for appropriate output.

- **CHAIN**

Chained access, which is the immediate forwarding of requests from the site where these occur.

- **STAR**

STAR access, which is the forwarding of requests from 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 authenticated 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 "**requiretty**" 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.

9.4 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 permuted 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.

9.5 ctys-wakeup

SYNTAX

ctys-wakeup

```

[-d <level>]
[-h]
[-H <help-options>]
[-i <interface>]
[-n]
[-p <port>]
[-t <TCP/IP address>]
[-V]
[-X]
<mac-address>

```

DESCRIPTION

ctys-wakeup ctys-wakeup sends a so called MagicPacket(TM) to a given destination. In case a remote segment is addressed by setting the "-t" option, a UDP message is sent to port "9". The script consists of a few lines of bash code with a call to Netcat. So it could be adapted easily.

```

mac=$1;shift
ip=$1;shift

function buildWOLMagicPacket () {
    local _mac=$1;shift
    declare -a _pdu;

    function macAsc2Hex () {
        for i in ${@};do
            printf "\x$i"
        done
    }

    #frame
    _pdu=(ff ff ff ff ff ff);

    #MAC addr in "little endian"
    addr=(${mac: })
    addrLitteEndian=(${addr[0]} ${addr[1]} \
                    ${addr[2]} ${addr[3]} \
                    ${addr[4]} ${addr[5]});

```

```

#add 16 duplications
for((i=0;i<16;i++));do
  for m in ${addrLitteEndian[@]};do
    size=${#_pdu[@]}
    _pdu[${size}]=$m
  done
done

macAsc2Hex ${_pdu[@]}
}

if [ -n "$ip" ];then
  printf "buildWOLMagicPacket $mac'\n'|nc -u -w 1 $ip 9
else
  buildWOLMagicPacket $mac
fi

```

OPTIONS

- d** <level> Sets debug.
- h** Print help, refer to "-H" for additional information.
- H** <help-option> The extended help option is based on system interfaces for display of manpages, PDF and HTML documents. This comprises the man pages and installed manuals.

For additional help refer to the documents or type **ctys -H help**.
- i** <broadcast-interface> The ethernet-broadcast interface for local segment. When used "-t" option is not applicable.
- n** No execution, evaluation only.
- p** <port> The port for destination, which is by default "9" for a hard-coded UDP message. It is not really relevant, might be used only for adaptation to firewall rule.
- t** <TCP/IP address> When provided the packet is send to destination. The destination address could be a "directed-broadcast" address, which has to be supported by the router.
E.g. OpenBSD requires "net.init.ip.directed-broadcast=1", and the appropriate pf-rules.
For 1.2.3.4:

```
<wol>    {1.2.3.255}
```

```
pass in on \${ifX} from <wolExec> to <wol>
```

```
pass out on \${ifY} from <wolExec> to <wol>
```

Alternatively a "rdr" - Redirection could be set. When this option is not provided, a UDP packet will be sent to "255:255:255:255:9".
When used "-i" option is not applicable.
- V** Version.
- X** See ctys, terse for machine output.

. ARGUMENTS

The MAC address of the targeted NIC, this could be evaluated by call of "ctys-macmap" and/or "ctys-vhost"

.

EXIT-VALUES

0: OK: Result is valid.

1: NOK: Erroneous parameters.

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

9.6 ctys-xdg

SYNTAX

ctys-xdg

```

[-d]
[-e]
[--force]
[--helpEx|-helpEx]
[-h|--help|-help]
[--menu-cancel]
[--menu-cancel=<config-template-root-or-home-pathname>]
[--menu-create]
[--menu-create=<config-template-root-or-home-pathname>]
[--menu-edit]
[--menu-edit=<files-or-directories>]
[--menu-gui]
[--menu-private]
[--menu-private=<target-base>]
[--menu-shared]
[--menu-shared=<target-base>]
[--no-user]
[--user]
[-V]
[-X]

```

DESCRIPTION

UNDER DEVELOPMENT: The current interface may change and the functionality will be expanded. For now basic features for template based private and shared menus are included. The validated support is mainly for Gnome desktops. KDE works almost too, some icon problems still remain, shifted due to priorities.

ctys-xdg is the main interface for the creation of menus and integration of the UnifiedSessionsManager into a graphical desktop. This includes the creation of the entries as well as removing them.

The Gnome desktops are provided by default, others require the **-force** option. This is due to limited support with some restrictions, though the user may decide case-by-case to apply or not. The installation of menus could be performed during installation by the **ctys-distribute(1)** utility too, but this does not provide a '-force' flag.

The following examples show the creation of the standard menu.

```
ctys-xdg --menu-create
```

When applied to KDE desktops the '-force' option is required.

```
ctys-xdg --menu-create --force
```

Whereas the following removes the entries but keeps the empty directories.

```
ctys-xdg --menu-cancel
```

OPTIONS

-d <debug-args>

Refer to "ctys" generic options for additional information.

-e

Calls the configured editor by **CTYS_MENUEDIT** with the search path

MENU_DEFAULT_PRIVATE_LST, which is set by default to

```
$HOME/.config/menus:$HOME/.local/share/applications:\
$HOME/.local/share/desktop-directories
```

-force

Forces the application to additional desktops.

-h

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

-H <help-option>

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

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

--menu-cancel=<config-template-root-or-home-pathname>

Removes the entries as currently present within the templates. Directories are not removed.

--menu-create

Creates the standard menu entries as provided by the pre-defined templates. The templates are stored within the configuration directory in **xdg.d**.

--menu-create=<config-template-root-or-home-pathname>

The same as **--menu-create** with additional alternative path for installation.

menu-edit=<files-or-directories>

The same as **-e** option.

menu-gui

Sets GUI based messages and dialogues as preference.

menu-private

Creates a private menu. This is the default.

menu-private=<target-base>

The same as **--menu-private** with alternative directory.

menu-shared

Creates a shared menu within standard path, this requires root permissions.

menu-shared=<target-base>

The same as **-menu-shared** with an alternative base path.

-no-user

This is required when a menu entry without a user local installation is going to be used. This stores the required environment for shared executables within a init file and prepends the call into the menu entries.

-user

The default for user based installation, which assumes the environment is set during each login.

-V

Version.

-X

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

. ARGUMENTS

None.

.

EXIT-VALUES

0: OK: Result is valid.

1: NOK: Erroneous parameters.

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

9.7 ctys-xen-network-bridge

SYNTAX

ctys-xen-network-bridge

```
ctys-xen-network-bridge [options] [<action-arguments>]
```

DESCRIPTION

ctys-xen-network-bridge is copied and adapted from the original script of Xen-3.0.3-25.0.4.el5 from CentOS-5.0 distribution for the UnifiedSessionsManager. This is preferred as an exceptional case, instead of a patch, because almost any second line is changed, even though for trace mainly. The intention is not to take ownership or claim any rights. The adaptation was necessary, mainly due to debugging issues as an integral part for setting up WoL by PM, requiring a previous re-ordering and re-configuration of the IF targeted by WoL packets. The usage of the provided standard tool for Xen-ified machines was just the natural step. Anyhow, should be obsolete, when the so called MAGIC-PACKET(TM) could be set on a vif, other WoL-attributes, such as broadcast, seems to work already.

- Copyright (C) by Owners of Xen
- Copyright (C) by Owners of CentOS
- Copyright (C) by RedHat Inc.

It is adapted in order of simplification of access grant for userland access. For ctys the "stop" case only is required when WoL has to be activated, thus this is the only verified call interface for now: “network-bridge stop bridge=<bridge-to-stop> netdev=<netdev>

This will be compatible for an immediate switch-over to the original script, when preferred. Some minor patches for suppression of debugging flags might be required. For application and manual call the following has to be supported:

- if appropriate: call only as root, than anything might be ok.
- if user land required:
 - set access bits for each PATH part of group for
/etc/xen/scripts/network-bridge
 - assure ksu/sudo for at least:
ip, ifup, brctl

Because this is just an error-workaround, for erroneous setting of "wol g" by ethtool, it may be removed later. For additional information refer to Xen documentation.

OPTIONS

All options are "optional".

bridge=<virtual-bridge> The name of the virtual bridge to stop, for Xen this is frequently "xenbr0."

netdev=<**original-network-device**> The name of the interface to stop, for Xen this is frequently "eth0" or "bond0".

vifnum=<**virtual-netdev-num**> Normally not required, else frequently this is "0".

. ARGUMENTS

Standard init argument.

- **status**
Standard init argument.
- **stop**
Standard init argument.

. EXIT-VALUES

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

9.8 ctys-utilities

SYNTAX

ctys-utilities

```

ctys-getMasterPid

getCPUinfo
getCurArch
getCurCTYSRel
getCurDistribution
getCurGID
getFSinfo
getGeometry
getHDDinfo
getHDDtemp
getMEMinfo
getCurOS
getCurOSRel
getCurDistribution
getCurRelease
getPerfIDX
getSolarisUUID
getVMinfo

pathlist

```

DESCRIPTION

ctys-utilities is a collection of tiny byt tools, which encapsulate some basic system information.

getCPUinfo

Displays CPU information including the HVM and PAE/PARA capabilities.

getCurArch

Displays the current architecture.

getCurCTYSRel

Displays the current release of the UnifiedSessionsManager.

getCurDistribution

Displays the name of the current distribution.

getCurGID

Displays the GID of current user.

getFSinfo

Displays the list and size of current home filesystems on separate partitions/slices.

getGeometry

Converts the extended format geometryExt for Xinerama addressing to standard geometry values for usage with X11 executables.

```
getGeometry
  [-g <geometryExt>]
  [--geometry=<geometryExt>]

  [-d <debug-args>]
  [-H|--helpEx]
  [-h|--help|-help]
  [-V]
  [-X]
```

The <geometryExt> represents the extended addressing as described by the standard generic parameters for ctys. The output is the calculated absolute address as required for standard executables.

getHDDinfo

Displays a list of current HDDs.

getMEMinfo

Displays a list of available memory including available swap space.

getCurOS

Echoes the name of current OS. Will be used widely for internal execution variant selection during runtime.

getCurOSRelease

Echoes the release of current OS. Will be used widely for internal execution variant selection during runtime.

getCurDistribution

Echoes the name of current distro. Will be used widely for internal execution variant selection during runtime.

getCurRelease

Echoes the release of current distro-release. Will be used widely for internal execution variant selection during runtime.

getCurOSRel

Displays the release of current OS.

getPerfIDX

Displays a standardized "ctys-index", for comparison of machine performance.

getSolarisUUID

Gets the UUID on Solaris.

getVMinfo

Displays the VM information, which is either VM or PM.

pathlist

Shell interface for "pathSplit" library function, lists each element of common colon separated search paths line by line. Could be used for trouble shooting support when PATH resolution fails.

ctys-getMasterPid

Internal call for evaluation of the master pid for a process. This works for detached processes and already running processes too. For additional

explanation refer to PROC-FIND-TOP-BOTTOM.

It is a quick-hack, do not rely on it.

Part III

Appendices

Chapter 10

Current Loaded Plugins

This section enumerates the current loaded static libraries and the dynamic loaded plugins. Which will be partly detected automatically and loaded as predefined or On-Demand.

The following list is generated with the call:

```
"ctys -T all -v"
```

REMARK: For limited environments this could produce errors due to memory exhaustion. The error messages are not obvious!!!

UnifiedSessionsManager Copyright (C) 2007, 2008, 2010 Arno-Can Uestuensoez

This program comes with ABSOLUTELY NO WARRANTY; for details refer to provided documentation.

This is free software, and you are welcome to redistribute it under certain conditions. For details refer to "GNU General Public License - version 3" <<http://www.gnu.org/licenses/>>.

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

AUTHOR            = Arno-Can Uestuensoez - unifiedsessionsmanager@protonmai
MAINTAINER        = Arno-Can Uestuensoez - unifiedsessionsmanager@protonmai
VERSION           = 01_10_011
DATE              = 2010.03.18

LOC               = 117540 CodeLines
LOC-BARE          = 59063 BareCodeLines (no comments and empty lines)
LOD               = 0 DocLines, include LaTeX-sources

TARGET_OS        = Linux:   CentOS/RHEL, Fedora, ScientificLinux,
                       debian, Ubuntu,
                       (gentoo,) mandriva,
                       (knoppix,) (dsl,)
```

```

SuSE/openSUSE
BSD:      OpenBSD, FreeBSD
Solaris:  Solaris-10, OpenSolaris
Windows:  (WNT/Cygwin), (W2K/Cygwin), (WXP/Cygwin),
           (W2Kx/Cygwin)

TARGET_VM      = KVM, (OpenVZ), QEMU, (VirtualBox,) VMware, Xen
TARGET_WM      = fvwm, Gnome, (KDE,) X11

GUEST_OS       = ANY(some with limited native-access support)
-----
COPYRIGHT      = Arno-Can Uestuensoez - unifiedsessionsmanager@protonmail.com
LICENCE        = GPL3
-----
EXECUTING HOST = ws2.soho
-----
LIBRARIES(static-loaded - generic):

```

Nr	Library	Version
00	bootstrap.01.01.004.sh	01.10.010
01	base.sh	01.07.001b01
02	libManager.sh	01.02.002c01
03	cli.sh	01.07.001b06
04	misc.sh	01.06.001a12
05	security.sh	01.06.001a05
06	help.sh	01.10.002
07	geometry.sh	01.07.001b06
08	wmctrlEncapsulation.sh	01.07.001b06
09	groups.sh	01.11.001
10	network.sh	01.11.001

```

PLUGINS(dynamic-loaded - ctys specific):

```

Nr	Plugin	Version
00	CORE/CACHE.sh	01.07.001b01
01	CORE/CLI.sh	01.07.001b06
02	CORE/COMMON.sh	01.02.002c01
03	CORE/CONFIG/hook.sh	01.06.001a14
04	CORE/DIGGER/hook.sh	01.07.001b06
05	CORE/DIGGER/list.sh	01.02.001b01
06	CORE/ENV.sh	01.02.002c01
07	CORE/EXEC.sh	01.06.001a15
08	CORE/GENERIC.sh	01.10.011
09	CORE/HELP.sh	01.02.002c01
10	CORE/LABELS.sh	01.07.001b05
11	CORE/STACKER/hook.sh	01.07.001b06
12	CORE/VMs.sh	01.02.002c01

13	GENERIC/hook.sh	01.02.001b01
14	GENERIC/LIST/list.sh	01.10.008
15	GENERIC/ENUMERATE/enumerate.sh	01.02.001b01
16	HOSTs/CLI/hook.sh	01.06.001a09
17	HOSTs/CLI/session.sh	01.01.001a01
18	HOSTs/CLI/list.sh	01.10.008
19	HOSTs/CLI/info.sh	01.02.001b01
20	HOSTs/VNC/hook.sh	01.02.001b01
21	HOSTs/VNC/session.sh	01.06.001a15
22	HOSTs/VNC/list.sh	01.07.001b05
23	HOSTs/VNC/info.sh	01.02.001b01
24	HOSTs/X11/hook.sh	01.06.001a09
25	HOSTs/X11/session.sh	01.01.001a01
26	HOSTs/X11/list.sh	01.01.001a00
27	HOSTs/X11/info.sh	01.02.001b01
28	VMs/QEMU/hook.sh	01.10.008
29	VMs/QEMU/config.sh	01.01.001a01pre
30	VMs/QEMU/session.sh	01.10.008
31	VMs/QEMU/enumerate.sh	01.10.008
32	VMs/QEMU/list.sh	01.10.008
33	VMs/QEMU/info.sh	01.01.001a00pre
34	VMs/VMW/hook.sh	01.10.009
35	VMs/VMW/session.sh	01.10.009
36	VMs/VMW/enumerate.sh	01.06.001a09
37	VMs/VMW/list.sh	01.10.009
38	VMs/VMW/info.sh	01.02.001b01
39	VMs/XEN/hook.sh	01.10.008
40	VMs/XEN/config.sh	01.01.001a01
41	VMs/XEN/session.sh	01.07.001b06
42	VMs/XEN/enumerate.sh	01.01.001a01
43	VMs/XEN/list.sh	01.10.008
44	VMs/XEN/info.sh	01.01.001a00
45	PMs/PM/hook.sh	01.10.008
46	PMs/PM/session.sh	01.01.001a00
47	PMs/PM/enumerate.sh	01.01.001a01
48	PMs/PM/list.sh	01.10.008
49	PMs/PM/info.sh	01.01.001a00

CTYS-INTERNAL-SUBCALLS:

Nr	Component	Version
00	ctys	01_10_011
01	ctys-callVncserver.sh	01_10_011
02	ctys-callVncviewer.sh	01_10_011
03	ctys-createConfQEMU.sh	01_10_011

04	ctys-distribute.sh	01_10_011
05	ctys-dnsutil.sh	01_10_011
06	ctys-extractARPlst.sh	01_10_011
07	ctys-extractMAClst.sh	01_10_011
08	ctys-genmconf.sh	01_10_011
09	ctys-groups.sh	01_10_011
10	ctys-getMasterPid.sh	01_10_011
11	ctys-install.sh	01_10_009
12	ctys-install1.sh	01_10_009
13	ctys-macros.sh	01_10_011
14	ctys-macmap.sh	01_10_011
15	ctys-plugins.sh	01_10_011
16	ctys-vnetctl.sh	01_10_011
17	ctys-smbutil.sh	01_10_011
18	ctys-vdbgen.sh	01_10_011
19	ctys-vhost.sh	01_10_011
20	ctys.sh	01_10_011
21	ctys-wakeup.sh	01_10_011
22	ctys-xen-network-bridge.sh	01_10_011

Helpers:

00	getCPUinfo.sh	01_10_011
01	getFSinfo.sh	01_10_011
02	getHDDinfo.sh	01_10_011
03	getMEMinfo.sh	01_10_011
04	getPerfIDX.sh	01_10_011
05	getVMinfo.sh	01_10_011

Tiny-Helpers:

00	getCurArch.sh	OK
01	getCurCTYSRel.sh	OK
02	getCurDistribution.sh	OK
03	getCurGID.sh	OK
04	getCurOS.sh	OK
05	getCurOSRel.sh	OK
06	getCurRelease.sh	OK
07	getSolarisUUID.sh	OK
08	pathlist.sh	OK

OPTIONAL/MANDATORY PREREQUISITES:

bash:GNU bash, version 3.2.25(1)-release (x86_64-redhat-linux-gnu)

SSH:OpenSSH_4.3p2, OpenSSL 0.9.8e-fips-rhel5 01 Jul 2008

VNC:VNC Viewer Free Edition 4.1.2 for X - built Mar 24 2009 19:52:30

wmctrl:wmctrl 1.07

CURRENT ARG-MEM-USAGE:

ArgList(bytes):"env|wc -c" => 3284

ArgList(bytes):"set|wc -c" => 810677

Chapter 11

File Formats

11.1 Common Tools and Formats

The most common internal file format is the output by MACHINE suboption. This is a semicolon separated file format, which optionally supports a TITLE line for each field. The format is compatible to MS-Excel.

The most important data collector and generator ctys-vdbgen is just a wrapper for the call of ctys with the action ENUMERATE. The replied data is transparently stored in the main file database "enum.fdb", which could be inspected and edited by any ASC-II editor or a common spreadsheet application. The caching boosts the performance beginning with the factor of 10, which could be much more, depending of various specific circumstances. Even though it could theoretically slow down, no practical example occurred.

The generated enum.fdb is just a raw static cache of distributed configuration data for VMs, which may lack some additional information. Therefore another file database is generated in order to map TCP/IP data of the Guest-OSs within VMs to their MAC-Addresses and DNS names. This file is called the macmap.fdb. It could be generated by the usage of the tools ctys-extractMAClst and/or ctys-extractARPlst.

The prefetched database ".fdb" are pre-processed on a second level by aggregating and correlating the data into a common final static cache-database "ctysd-vhost.statcache.cdb". Additional dynamic intermediate data may be cached within tmp directory. The databases could be listed and inspected by the tool ctys-vhost.

`ctys-vhost -S list`

Lists source databases of first level cache "fdb", which includes the non-cached group files.

`ctys-vhost -C list`

Lists statcache caches of second level prefetch, which includes the cached group files.

Some additional options exist within ctys-vhost for listing of members and various output.

11.2 Groups

The groups concept as described within the introduction supports the assembly of arbitrate entities into a common entity, which will be handled as one logical instance. This includes the nesting of includes of groups. Therefore a file has to be supported in a defined subdirectory, with the name as the literal group name to be used within ctys arguments as a replacement for a <machine-address>.

The group file supports the keyword "#include" which has to be left-most on a line followed by a groupname. The number of includes and the level of nesting is not limited. Recursion loops are checked and aborted by a pre-defined level. The subdirectory containing the group will be search from a given list of directories by the path variable "CTYS_GROUPS_PATH", which uses a colon seperated list of directory paths.

The ordinary host entries could be one on each line, or a list of more than one comma-seperated entries on a line. For each entry could be context specific options set the same as on the command line arguments. An example with nested includes is supported in the install sub-directory conf/ctys/groups. Following is an example with context options.

```
#
#This groups contains all machines supporting VMs to be used by
#user acue.
#
#Almost any machine is currently accessible by NFS, thus allows
#a simple means of load-balancing throughout the memgers of
#this group.
#
#The only distinction has to be done by type of VM, which
#depends on the actual running kernel.
#
#
#REMINDER: Any option is dominant from it's first occurance on,
#           until overwritten. This is one of the limits of
#           missing namespaces and smart usable assembled data
#           structures within bash. But anyway the real huge
#           benefit from using bash is the opportunity for almost
#           anyone to add his own modules by simple shell scripts.
```

```

#
#       It has to be recognized, that due to some basic
#       requirements - e.g. grouping of potential
#       Desktop/Workspace display, the jobs are re-ordered.

#       So it is recommended, that when any context-option
#       is required, that ALL might be assigned it's own
#       options.
#
#       So, it's accepted!
#
#
#OpenBSD PM, for tests only. Supports no VMs, Just OpenBSD-PM and
#common HOSTs-(CLI,X11,VNC)
root@host1'(-a enumerate=machine)'

#Multi-Display-WS with local emulation for usage of
#Win-Equipment, such as Dymo-LabelWriter 320 with it's own
#drivers within VMWare and W2K.
host2'(-a enumerate=machine)'

#WMware WS workstation, DualOpteron-244
host3'(-a enumerate=machine)' user5@host0'(-a enumerate=machine)'

#Experimental Dual-PIII-Coppermine, yum-install of
#VMware-server/player-variants

host4'(-a enumerate=machine)' user3@host5'(-a enumerate=machine)'

#Main fileserver with DualP-III-Thualatin
host5'(-a enumerate=machine)' user3@host10'(-a enumerate=machine)'

#Backup and Experimental HVM by Core2-Duo-6300
host6'(-a enumerate=machine)' user2@host9'(-a enumerate=machine)'

#Experimantal Celeron - S420
root@host6'(-a enumerate=machine)'

#Experimental Celeron 2,4GHz
root@host7'(-a enumerate=machine)'

#app1 as primary paravirtualized DualOpteron-Xen-Server
host8'(-a enumerate=machine,b:/homen/userA%/home1/xen)'

```

On the ctys call the group could be given context options, even when it's members have context options. But not the include-statements. Due to the philosophy, that the last wins, the outer . this is the CLI options - will replace options left of it, within the groups. When using ctys-vhost the groups will be cached as a pre-resolved subtree in common database (ENUMERATE) format. This will be recursively done for each group file as described in the following chapters.

11.3 Macros

Thus a macro can contain any part of a call except the command itself. The whole set of required options including the execution target or only a subset of options could be stored within a macro. The macro and its content are stored within a file which could be edited by each user or provided as a common defaults file. A macro is defined within the default file "default" which is searched in the order:

1. "\$HOME/.ctys/macros/default"
2. "<actual-call-conf-path>/macros/default"

The <actual-call-conf-path> is evaluated from the resolved symbolic link of the call. The following call syntax is provided:

```
MACRO:(
<macro-name>
[%<macro-file-db>]
  [(
    ECHO
    |EVAL]
  )
]
)
```

MACROs could be nested and chained as required. Even though the recursion depth could be arbitrary a counter is implemented, which sets a threshold limiting recursive processing. This is set by the configuration variable CTYS_MAXRECURSE. The variable protects all recursion depths, thus should be handled carefully. Default is 15 levels. The keyword "MACRO" prefixes the actual macro alias with the following parts.

<macro-name>

The actual name of the alias to be replaced.

<macro-file-db>

The default macro file could be altered by this new filename. The "macros" directories will be scanned for a file with given name.

ECHO

The given macro is inserted by "echo" command into the replacement position, which is the default behaviour.

EVAL

The macro is evaluated on the callers site by "eval" call and the result is inserted into the insertion position.

11.4 Static Import File Databases - fdb

11.4.1 macmap.fdb

This file contains the output of the standard call to one of the tools "ctys-extractMAClst" or "ctys-extractARPlst". Which is a three colon semicolon seperated table, callee file-database. The record format is

```
<DNS-name>;<MAC-address>;<TCP/IP-address>
```

The default format is expected by the post-processing tools. The tools `ctys-extractMAClst` and `ctys-extractARPlst` could be used by themselves for search and output to `stdout` too.

`ctys-vhost` allows by the flag "MACMAP" the optional usage of this database when only output suitable is selected for the "-o" option. Or allows the forced usage by "MACMAPONLY". `ctys-macmap` works natively on `macmap.fdb`.

11.4.2 enum.fdb

The `enum.fdb` file is the first level cached raw output from the `ENUMERATE=MACHINE` call. Therefore the call of `ctys-vdbgen` is used, which is a wrapper for the `ctys` call. Any option of `ctys` could be provided for `ctys-vdbgen` and will be passed through transparently. Therefore the flag "-T" and the given targets, e.g. as a group, could be used to constrain the collected and stored data. Using this features several databases could be used independently with different scopes of network view.

The record format is as shown with the call "ENUMERATE=MACHINE,TITLE":

```
"ContainingMachine;SessionType;Label;ID;UUID;MAC;TCP/IP;\
VNCAccessPort;VNCBasePort;VNCDisplay;Distribution;OS;\
VersionNr;SerialNr;Category"
```

```
host.soho;QEMU;tst100;\
/homen//tst100-01.01.001.x86\_64/tst100-inst.conf;\
;00:50:56:13:11:40;;;;;;
```

Space are not allowed, multiple entries will be ignored, just the first will be used.

11.5 Static Pre-Fetch Cache Databases - cdb

The Pre-Fetch Caches are the second level, where some time consuming pre-processing and correlation is performed. This also includes the finale resolution of the group files by replacing each <machine-address> entry with the appropriate `ENUMERATE` result, and resolving the whole resulting include-tree.

The common `enum.fdb` entries are stored within one database file, whereas the groups are resolved with their whole tree for each defined group file.

The record format is the common `ENUMERATE` format.

11.5.1 statcache.cdb

This is the cached `enum.fdb`. In addition to `enum.fdb` the `macmap.fdb` is correlated to any ITCP relevant field. Therefore - if available - any item should now contain a MAC-address and the related TCP/IP-address. This allows by simple search operations the evaluation of the GuestOSs TCP/IP-address from the VMs configuration file. Thus the native access to the GuestOS.

11.6 Dynamic Runtime Cache Databases

Some internal data is cached into files, which could be controlled by flags. This is due to reuse of data spanning more than one call.

11.7 Configuration Entries for ctys

11.7.1 Actual Processing vs. Administrative Display

The first point to mention is the actual functionality supported by the various configuration options. The native configuration files of the various hypervisors support values in order to actually effect their managed VM. The parameters displayed with ENUMERATE option are read out from the native configuration with highest priority, where possible. When this fails or is simply not provided, the manual stored data is used.

11.7.2 Configuration File Variants

The various supported VMs have partly quite different configuration options for their VM files. Therefore some addons tho the existing files and additional specific files for ctys are defined. The basic convention for the naming and location of VMs configuration files is defined as:

- (a) Any VM has it's own directory, wherein all related files are stored. These are the bld/build/output/doc-tmp/en/images and the configuration files. This is commonly pre-required for all VMs.
- (b) The name of the directory is the same as the VM name, which could be the display name as well as a DomU name. For the directory name some variations could be set, e.g. due to versioning or backups. The tools like ctys-vhost search by default for the first match by alphabetical order only, which eliminates the usage of backup files.
- (c) The configuration file will be named the same as the containing directory, but with the appropriate post-fix, which is e.g. 'vmx' or 'conf'.
- (d) Additional files, to be optionally used for ctys information when required are:
 - i. Same filenamepath as the VMs config-file, but with the post-fix 'ctys'.
 - ii. The same file stored in the parent directory.
 - iii. When common definitions to be used, these finally are searched within the CTYSCONF file.

The scan for resolution will be performed until a match occurs, thus the first match wins.

The order of search is:

- (a) <vmx-file>
 - i. The native configuration file of the VM. The scanning subsystem first tries to match standard information provided by the native syntax of the VM.
 - ii. Second the ctys specific entries will be scanned.
- (b) \${<vmx-file>%.*}.ctys
The ctys-file coallocated within the same directoy as the plugin specific configuration file.

- (c) 'dirname <vmx-file>'.ctys
The parent directory of the plugins configuration directory.
- (d) \${CTYSCONF}
The predefined ctys configuration file.

11.7.3 Keywords

The following keywords are defined:

- `##ARCH`
The supported architecture of the VM. For additional description refer to Section 11.7.7 '[Virtual Hardware-Platform](#)' on page 347 .
- `##CTYSRELEASE`
The identified of the release, which was used to create the current record, helpful for distributed access.
- `##CATEGORY`
The category of this Machine. Currently VM or PM.
- `##CSTRG`
A private context string supported for the plugin.
- `##DIST`
The distribution name of the GuestOS.
- `##DISTREL`
The release of the distribution of the GuestOS.
- `##EXECLOCATION`
- `##GATEWAY`
The gateway to be used by the GuestOS. This could be registered specific to each interface.
- `##HWCAP`
The provided hardware capacity.
- `##HWREQ`
The required hardware capacity.
- `##HYPERREF`
The release of the hypervisor used to create the VM.
- `##IP[0-9]*`
The IP parameter is tightly correlated with the MAC parameter. For additional description refer to Section 11.7.4 '[Interface Keywords](#)' on page 344 .
- `##INST_EDITOR`
Responsible person installed the VM.
- `##INST_DATE`
Date of installation.
- `##INST_CTYSREL`
CTYS-Release of installation.
- `##INST_VERNO`
Version number of VM.
- `##INST_SERNO`
Serial number of VM.

`#@#INST_UID`
 User ID.

`#@#INST_GID`
 Group ID.

`#@#INST_HOST`
 Host of installer execution.

`#@#INST_HOST_DIST`
 Distribution running on host.

`#@#INST_HOST_DISTREL`
 Distribution release running on host.

`#@#INST_HOST_OS`
 OS on host.

`#@#INST_HOST_OSREL`
 OS version on host.

`#@#INST_QEMUBASE`
 QEMU executable.

`#@#INST_QEMUBASE_HYPERREL`
 QEMU executable release.

`#@#INST_QEMUBASE_MAGICID`
 QEMU magic ID.

`#@#INST_QEMUBASE_ACCELERATOR`
 Type of supported QEMU accelerator.

`#@#INST_QEMUKVM`
 QEMU executable.

`#@#INST_QEMUKVM_HYPERREL`
 QEMU with KVM executable release.

`#@#INST_QEMUKVM_MAGICID`
 QEMU with KVM magic ID.

`#@#INST_QEMUKVM_ACCELERATOR`
 Type of supported KVM accelerator.

`#@#LABEL`
 Label of VM, which is used as unique name. Could be e.g. the display name or a DomU name.

`#@#MAC[0-9]*`
 The MAC parameter is tightly correlated with the IP parameter. For additional description refer to Section 11.7.4 ‘[Interface Keywords](#)’ on page 344 .

`#@#MAGICID-<plugin>`
 A magicID to for classification of the managing plugin for this type. For additional description refer to Section 11.7.5 ‘[MAGICID](#)’ on page 346 .

`#@#OS`
 The name of the OS.

`#@#OSREL`
 The release of the GuestOS.

#@#PLATFORM

The supported virtual HW as execution base for the GuestOS. Advanced support for this option is provided by QEMU. For additional description refer to Section 11.7.7 ‘**Virtual Hardware-Platform**’ on page 347 .

#@#RELOCCAP

The provided relocation capacity. For additional description refer to Section 11.7.9 ‘**Execution Location and Relocation**’ on page 349 .

#@#SERNO

A free serial number for administration by the user. The tools generate the recommended format by utilising the "date" tool with the call:

```
date +%Y%m%d%H%M%S,
```

which might be sufficiently unique, when seen within it’s context.

#@#SESSIONTYPE

The type of session. This field will be evaluated by the plugin itself, passed through to the main dispatcher.

#@#SPORT

The serverport for management access to the hypervisor. This is supported by Xen and QEMU.

#@#STACKCAP

The offered stacking capacity of the VM. For additional description refer to Section 11.7.8 ‘**Stacking Entries**’ on page 349 .

#@#STACKREQ

The required stacking capacity by the VM. For additional description refer to Section 11.7.8 ‘**Stacking Entries**’ on page 349 .

#@#USTRG

An arbitrary user string. For now just trusted, means no specific validation is done. Particularly the size should be kept small, which means some bytes only, as a reminder for the VMs task for example.

#@#UUID

The UUID of the machine.

#@#VCPU

The number of configured virtual CPUs. For Xen called within Dom0 with the "PM" argument used the actual present CPUs, which is the sum of all CPU cores, is additionally displayed.

```
<Dom0-cpus>/<total-number-of-cores>
```

#@#VERSION

The release of the OS, which is the kernel release for Linux.

#@#VMSTATE

The state of the VM. For additional description refer to Section 11.7.6 ‘**VMSTATE**’ on page 346 .

#@#VNCACCESSPORT

The port number of VNC access for the GuestOS, if required to be defined explicitly.

#@#VNCACCESSDISPLAY

The DISPLAY when required to be fixed.

#@#VNCBASEPORT

The baseport when to be calculated with usage of display.

#@#VRAM

The configured RAM for the GuestOS, when called with "VM".

For Xen called within Dom0 with the "PM" argument used the total amount of physical RAM is additionally displayed.

<Dom0-RAM>/<total-physical-RAM>

Additional data could be provided for and by the specific plugin. This is required for example in the specific configuration file defined for the QEMU plugin by ctys. Which has to be wrapped due to it's call-option only interface, to be unified in accordance to the common usage with the remaining VMs.

11.7.4 Interface Keywords

Specific consideration is required for the interface configuration and it's representation within the GuestOS, particularly when multiple interfaces are configured.

As mentioned before, the first attempt of the ENUMERATE action is to read-out the native values of the VM. This is commonly for the MAC address only, where in case of XEN "arbitrary" default values are set when values are not present. In case of VMware the hypervisor eventually resets these values dynamically if not configured to be static.

Thus the first requirement of the UnifiedSessionsManager is the static configuration of MAC addresses.

The next point is the missing interface index within Xen and the varying prefix of the interfacenames within OpenBSD guests for example. Therefore ctys numbers the interface by the order they were found. This is done for the native configured interfaces within the configuration files, where no index is provided. In case of names containing a numbering part, this number is kept. When redundancy occurs, the first match wins, and a warning is displayed for the second, which is dropped. When applicable the binding of interface names to the actualy HW devices should be fixed, this is particularly true, when hotplug Ethernet devices are used. This could be done by configuring MAC addresses into ifcfg-scripts and/or by usage of such tools as "ifrename".

The next step is to evaluate eventually configured TCP/IP parameters, like the IP address from the configuration files. This is currently "somehow" supported by Xen only, but is not indexed too. Thus ctys adds an "IP[0-9*]" options, which provides the whole set of required parameters for the VMs as depicted within the following description. When redundancy occurs, the behaviour is the same as for MAC addresses.

The correlation between the MAC address representing the hardware item, and the TCP/IP parameters representing the upper communications protocol layers is given by the index values evaluated before. Thus the first TCP/IP entity with the address "0" is assigned to the first AMC address with the same index. The rest is worked out as might be expected. For TCP/IP addresses without an assigne MAC interface a warning is displayed.

The following syntax is available to be applied.

```
<INDEXEDENTRY>[0-9]* = <entity>
```

```
<entity> =:
  <elements>[,<entity>]
```

```
<elements> =:
  <entry0>[%<entry1>[%<entry2>[%<entry3>[...]]]]
```

The specific adaption for MAC addresses:

```
MAC[0-9]* = <mac-entity>
```

Where a missing index is equal to 0. The specific adaption for IP:

```
IP[0-9]* = <ip-entity>
```

```
<ipentity> =:
  <ip-elements>[,<ip-entity>]
```

```
<ip-elements> =:
  [<dotted-IP-addr>] [%<netmask>] [%<relay>] \
  [%<ifname>] [%<ssh-port>] [%<gateway>]
```

Where a missing index is equal to 0. The values mostly are edited independent from the actual configuration. Thus particularly do not necessarily represent a static configuration. In case of DHCP these should be configured too, but DHCP might use fixed address assignments.

This is also the style the `ctys-extractMAClst` utility relies on.

<dotted-IP-addr>

TCP/IP address in numerical form should be preferred. Netmask - if present - has to be provided within it's own field.

<netmask>

The netmask.

<relay> This is the local interconnection device, which could be a virtual bridge, switch/hub or a router. This version supports bridges and switches/hubs only. Support for shorewall will follow soon, check it out.

<ifname>

The name of the interface. Multiple IP addresses on the same interface are supported, thus could have the form "`<ifname>:#ifnum`". The consistency of the interface names is within the responsibility of the user.

<ssh-port>

An alternate port for SSH. OpenSSH supports different ports for each interface.

<gateway>

An individual gateway for networks to be accessed by current interface. Additional settings are required, else the default is used.

The following is an example for configuration of the IP addresses only.

```
#@#IP2 = "11.0.0.11,11.0.11.0,11.11.0.0"
```

The next example provides the full scope of possible information to be stored.

```
#@#IP3 = \
  "11.0.0.1%255.255.255.0%xenbr0%eth3%222,\
  11.0.1.0%255.255.255.0%xenbr1%eth3:1%223"
```

Yes, the <CR> is for LaTeX only.

11.7.5 MAGICID

The MAGICID defines specific behaviour for the filesystem scanners how to recognize the found configuration. When the plugin value only is found, it will immediately accept the file as a valid configuration. The values could be used for <plugin> in order to control the scan behaviour of ENUMERATE.

- <plugin>
The SESSIONTYPE of the owning plugin.
- IGNORE
Ignores the file without any further processing.
- NOENUM
ffs.

11.7.6 VMSTATE

The state of the VM, currently ACTIVE is supported only.

11.7.7 Virtual Hardware-Platform

The virtual hardware platform is closely correlated to the architecture supported by the VM. These are dependent on the type and capabilities of the utilized hypervisor.

Currently the following non-comprehensive list of supported architectures and platforms is available, which depends partly on the actually used physical hardware base. For additional information refer to the specific documentation of the VMs.

- Architecture - ARCH
 - i386
VMWARE, XEN, QEMU
 - x86_64, amd64
VMWARE, XEN, QEMU
 - arm9
QEMU, for an example with debian installation refer to Section ?? ‘??’ on page ?? , for the provided example by QEMU refer to Section ?? ‘??’ on page ?? .
 - coldfire, PowerPC, MIPS, SPARC, SH3, ...
For display of a list of the current installed release of QEMU the INFO action could be used. An example display is provided within Section ?? ‘??’ on page ?? . Preconfigured VM configuration files requiring less adaption to local filesystem are available for the provided examples by QEMU, for additional information refer to .
- Platform - PLATFORM
 - pc-standard-platform
A standard PC platform with widely available emulated hardware components is supported by VMware, Xen, and QEMU. This includes particularly the option to configure the available RAM and the number of present CPUs.
 - ...
QEMU supports a variety of architectures of standard and embedded devices. For additional information refer to user-manual.

The hardware relevant entries describe the present hardware. Therefore the following generic format to each entry is applied.

```
<entity> =:
[<#cnt>x]<elements>[%<add-elements>] [,<entity>]

<elements> =:
<component>-<version>-<architecture>[-<opt-attr>]
<opt-attr> =: <attr>[-<opt-attr>]
<#cnt>      =: number present

<add-elements> =: <elements>[%<add-elements>]
```

The overall order of the entries contain the following elements, which could be either physical or virtual.

- (a) CPUs:

```
<vendor>-<family>-<model>-<stepping>-<frequency>-<cache>\  
[-<VM-cap>]
```

```
<VM-cap> := (SVM|VMX)|PAE
```

This format could vary for non-Linux OSs. Non applicable or available but mandatory fields are padded with dots('.').

(b) RAM:

```
<RAM>,<SWAP>
```

(c) HDD:

```
<device-1st> := <device>[%<device-1st>]
```

(d) FS:

```
<home-1st> := <home>[0-9]*-<size>[%<home-1st>]
```

Currently the home-partitions onyl are displayed by "ctys-genmconf".

11.7.8 Stacking Entries

The stack relevant entries describe the offered VM and PM capacity as well as the required base. Therefore the following generic format to each entry is applied.

```
<entity> =: <elements>[%<add-elements>] [,<entity>]
```

```
<elements> =:
```

```
<component>-<version>-<architecture>[-<opt-attr>]
```

```
<opt-attr> =: <attr>[-<opt-attr>]
```

```
<add-elements> =: <elements>[%<add-elements>]
```

The overall order of the entries contain the following elements.

- (a) VM plugin.
- (b) A list of present hypervisor capabilities.
- (c) Release of present stack support.
- (d) Additional information.

11.7.9 Execution Location and Relocation

The execution locations and the supported and/or allowed relocation behaviour. Currently just for display. For current supported values refer to RELOCCAP and EXECLOCATION .

Chapter 12

Miscellaneous

12.1 Basic EXEC principle

The basic execution principle of ctys is first to analyse the given options and build up an call array. Therefore several distinctions have to be made as resulting from permutation and superposing of the expanded CLI arguments. These array is finally executed in sets dependent from multiple criteria. Some examples are:

- Grouping of common sessions for each desktop, due to reliability and addressing gaps when shifting windows between desktops.
- Grouping of sessions for each remote server, but only if not ConnectionForward is choosen, because the current OpenSSH release does not support MuxDemux of multiple XSessions with different Displays.
- Splitting of Remote Server and Local Client execution of ctys, for VMs even though the VM-configuration is available on the server site only, which requires a remote component of ctys to be executed.
- ...and so on.

The implementation of ctys is pure bash with usage of the common shell components such as awk and sed. The whole design is based on unique set of sources which will be executed as the local initial call and client starter part as well as the remote server execution script. In case of DISPLAYFORWARDING the local component just initiates the remote co-allocated execution of Client and Server component. For the case of LOCALONLY both will be locally executed, thus the ctys acts locally as initial caller, server starter and client starter script.

To assure consistency and compatibility the remote and local versions will be checked and the execution is proceeded only in case of an match of both versions.

12.2 PATH

First of all, this is normally just required when handling different versions during development. This is particularly true, when during development a version is executed which is not contained within the standard PATH. This is particularly to be recognized, when executing the remote component which relies on the PATH mechanism too. Therefore the two environment variables are defined:

R_PATH: Replaces path for remote execution.
 L_PATH: Replaces path for local execution.

The local component L_PATH is required for local execution too, because the following subcalls of ctys will be executed based on PATH mechanism, which is most often different to initial path-prefixed test-call. For example this is for calling a test version for starting a local client and remote server from a test path without changing PATH:

```
V=01\_01\_007a01;\
export R\_PATH=/mntbase/ctys/src/ctys.\$V/bin:\$PATH;\
export L\_PATH=\$R\_PATH;\
ctys.\$V/bin/ctys -t vmw \
  -a create=b:\$HOME/vmware/dir2%\$HOME/vmware/dir3,\
    l:"GRP01-openbsd-4.0-001",REUSE
  -g 800x400+100+300:3 \
  -L CF \
  -- '(-d 99)' app2
```

The same for common user with standard install will be:

```
ctys -t vmw \
  -a create=\
    base:\$HOME/vmware/dir2%\$HOME/vmware/dir3\
    ,label:"GRP01-openbsd-4.0-001"\
    ,REUSE
  -g 800x400+100+300:3 \
  -L CF \
  app2
```

12.3 Configuration files

The configuration of ctys is performed in 4 steps, first has highest priority.

- (a) Environment Variable If an environment variable is set, it dominates other settings and it's value is kept.
- (b) \$HOME/.ctys/ctys.conf Config-File sourced: \$HOME/.ctys/ctys.conf
- (c) <install dir>/conf/ctys.conf Config-File sourced: <install dir>/conf/ctys.conf
- (d) Embedded defaults in ctys.

Chapter 13

LICENSES

Additionally a separate document including all licenses is contained within the package.

ctys-Licenses-01.11-print.pdf

13.1 CCL-3.0 With Attributes

License

THE WORK (AS DEFINED BELOW) IS PROVIDED UNDER THE TERMS OF THIS CREATIVE COMMONS PUBLIC LICENSE ("CCPL" OR "LICENSE"). THE WORK IS PROTECTED BY COPYRIGHT AND/OR OTHER APPLICABLE LAW. ANY USE OF THE WORK OTHER THAN AS AUTHORIZED UNDER THIS LICENSE OR COPYRIGHT LAW IS PROHIBITED.

BY EXERCISING ANY RIGHTS TO THE WORK PROVIDED HERE, YOU ACCEPT AND AGREE TO BE BOUND BY THE TERMS OF THIS LICENSE. TO THE EXTENT THIS LICENSE MAY BE CONSIDERED TO BE A CONTRACT, THE LICENSOR GRANTS YOU THE RIGHTS CONTAINED HERE IN CONSIDERATION OF YOUR ACCEPTANCE OF SUCH TERMS AND CONDITIONS.

1. Definitions

1. "Adaptation" means a work based upon the Work, or upon the Work and other pre-existing works, such as a translation, adaptation, derivative work, arrangement of music or other alterations of a literary or artistic work, or phonogram or performance and includes cinematographic adaptations or any other form in which the Work may be recast, transformed, or adapted including in any form recognizably derived from the original, except that a work that constitutes a Collection will not be considered an Adaptation for the purpose of this License. For the avoidance of doubt, where the Work is a musical work, performance or phonogram, the synchronization of the Work in timed-relation with a moving image ("synching") will be considered an Adaptation for the purpose of this License.

2. "Collection" means a collection of literary or artistic works,

such as encyclopedias and anthologies, or performances, phonograms or broadcasts, or other works or subject matter other than works listed in Section 1(f) below, which, by reason of the selection and arrangement of their contents, constitute intellectual creations, in which the Work is included in its entirety in unmodified form along with one or more other contributions, each constituting separate and independent works in themselves, which together are assembled into a collective whole. A work that constitutes a Collection will not be considered an Adaptation (as defined above) for the purposes of this License.

3. "Distribute" means to make available to the public the original and copies of the Work through sale or other transfer of ownership.

4. "Licensor" means the individual, individuals, entity or entities that offer(s) the Work under the terms of this License.

5. "Original Author" means, in the case of a literary or artistic work, the individual, individuals, entity or entities who created the Work or if no individual or entity can be identified, the publisher; and in addition (i) in the case of a performance the actors, singers, musicians, dancers, and other persons who act, sing, deliver, declaim, play in, interpret or otherwise perform literary or artistic works or expressions of folklore; (ii) in the case of a phonogram the producer being the person or legal entity who first fixes the sounds of a performance or other sounds; and, (iii) in the case of broadcasts, the organization that transmits the broadcast.

6. "Work" means the literary and/or artistic work offered under the terms of this License including without limitation any production in the literary, scientific and artistic domain, whatever may be the mode or form of its expression including digital form, such as a book, pamphlet and other writing; a lecture, address, sermon or other work of the same nature; a dramatic or dramatico-musical work; a choreographic work or entertainment in dumb show; a musical composition with or without words; a cinematographic work to which are assimilated works expressed by a process analogous to cinematography; a work of drawing, painting, architecture, sculpture, engraving or lithography; a photographic work to which are assimilated works expressed by a process analogous to photography; a work of applied art; an illustration, map, plan, sketch or three-dimensional work relative to geography, topography, architecture or science; a performance; a broadcast; a phonogram; a compilation of data to the extent it is protected as a copyrightable work; or a work performed by a variety or circus performer to the extent it is not otherwise considered a literary or artistic work.

7. "You" means an individual or entity exercising rights under this License who has not previously violated the terms of this License

with respect to the Work, or who has received express permission from the Licensor to exercise rights under this License despite a previous violation.

8. "Publicly Perform" means to perform public recitations of the Work and to communicate to the public those public recitations, by any means or process, including by wire or wireless means or public digital performances; to make available to the public Works in such a way that members of the public may access these Works from a place and at a place individually chosen by them; to perform the Work to the public by any means or process and the communication to the public of the performances of the Work, including by public digital performance; to broadcast and rebroadcast the Work by any means including signs, sounds or images.

9. "Reproduce" means to make copies of the Work by any means including without limitation by sound or visual recordings and the right of fixation and reproducing fixations of the Work, including storage of a protected performance or phonogram in digital form or other electronic medium.

2. Fair Dealing Rights. Nothing in this License is intended to reduce, limit, or restrict any uses free from copyright or rights arising from limitations or exceptions that are provided for in connection with the copyright protection under copyright law or other applicable laws.

3. License Grant. Subject to the terms and conditions of this License, Licensor hereby grants You a worldwide, royalty-free, non-exclusive, perpetual (for the duration of the applicable copyright) license to exercise the rights in the Work as stated below:

1. to Reproduce the Work, to incorporate the Work into one or more Collections, and to Reproduce the Work as incorporated in the Collections; and,

2. to Distribute and Publicly Perform the Work including as incorporated in Collections.

The above rights may be exercised in all media and formats whether now known or hereafter devised. The above rights include the right to make such modifications as are technically necessary to exercise the rights in other media and formats, but otherwise you have no rights to make Adaptations. Subject to 8(f), all rights not expressly granted by Licensor are hereby reserved, including but not limited to the rights set forth in Section 4(d).

4. Restrictions. The license granted in Section 3 above is expressly made subject to and limited by the following restrictions:

1. You may Distribute or Publicly Perform the Work only under the terms of this License. You must include a copy of, or the Uniform

Resource Identifier (URI) for, this License with every copy of the Work You Distribute or Publicly Perform. You may not offer or impose any terms on the Work that restrict the terms of this License or the ability of the recipient of the Work to exercise the rights granted to that recipient under the terms of the License. You may not sublicense the Work. You must keep intact all notices that refer to this License and to the disclaimer of warranties with every copy of the Work You Distribute or Publicly Perform. When You Distribute or Publicly Perform the Work, You may not impose any effective technological measures on the Work that restrict the ability of a recipient of the Work from You to exercise the rights granted to that recipient under the terms of the License. This Section 4(a) applies to the Work as incorporated in a Collection, but this does not require the Collection apart from the Work itself to be made subject to the terms of this License. If You create a Collection, upon notice from any Licensor You must, to the extent practicable, remove from the Collection any credit as required by Section 4(c), as requested.

2. You may not exercise any of the rights granted to You in Section 3 above in any manner that is primarily intended for or directed toward commercial advantage or private monetary compensation. The exchange of the Work for other copyrighted works by means of digital file-sharing or otherwise shall not be considered to be intended for or directed toward commercial advantage or private monetary compensation, provided there is no payment of any monetary compensation in connection with the exchange of copyrighted works.

3. If You Distribute, or Publicly Perform the Work or Collections, You must, unless a request has been made pursuant to Section 4(a), keep intact all copyright notices for the Work and provide, reasonable to the medium or means You are utilizing: (i) the name of the Original Author (or pseudonym, if applicable) if supplied, and/or if the Original Author and/or Licensor designate another party or parties (e.g., a sponsor institute, publishing entity, journal) for attribution ("Attribution Parties") in Licensor's copyright notice, terms of service or by other reasonable means, the name of such party or parties; (ii) the title of the Work if supplied; (iii) to the extent reasonably practicable, the URI, if any, that Licensor specifies to be associated with the Work, unless such URI does not refer to the copyright notice or licensing information for the Work. The credit required by this Section 4(c) may be implemented in any reasonable manner; provided, however, that in the case of a Collection, at a minimum such credit will appear, if a credit for all contributing authors of Collection appears, then as part of these credits and in a manner at least as prominent as the credits for the other contributing authors. For the avoidance of doubt, You may only use the credit required by this Section for the purpose of attribution in the manner set out above and, by exercising Your rights under this License, You may not implicitly or explicitly assert or imply any connection with,

sponsorship or endorsement by the Original Author, Licensor and/or Attribution Parties, as appropriate, of You or Your use of the Work, without the separate, express prior written permission of the Original Author, Licensor and/or Attribution Parties.

4.

For the avoidance of doubt:

1. Non-waivable Compulsory License Schemes. In those jurisdictions in which the right to collect royalties through any statutory or compulsory licensing scheme cannot be waived, the Licensor reserves the exclusive right to collect such royalties for any exercise by You of the rights granted under this License;

2. Waivable Compulsory License Schemes. In those jurisdictions in which the right to collect royalties through any statutory or compulsory licensing scheme can be waived, the Licensor reserves the exclusive right to collect such royalties for any exercise by You of the rights granted under this License if Your exercise of such rights is for a purpose or use which is otherwise than noncommercial as permitted under Section 4(b) and otherwise waives the right to collect royalties through any statutory or compulsory licensing scheme; and,

3. Voluntary License Schemes. The Licensor reserves the right to collect royalties, whether individually or, in the event that the Licensor is a member of a collecting society that administers voluntary licensing schemes, via that society, from any exercise by You of the rights granted under this License that is for a purpose or use which is otherwise than noncommercial as permitted under Section 4(b).

5. Except as otherwise agreed in writing by the Licensor or as may be otherwise permitted by applicable law, if You Reproduce, Distribute or Publicly Perform the Work either by itself or as part of any Collections, You must not distort, mutilate, modify or take other derogatory action in relation to the Work which would be prejudicial to the Original Author's honor or reputation.

5. Representations, Warranties and Disclaimer

UNLESS OTHERWISE MUTUALLY AGREED BY THE PARTIES IN WRITING, LICENSOR OFFERS THE WORK AS-IS AND MAKES NO REPRESENTATIONS OR WARRANTIES OF ANY KIND CONCERNING THE WORK, EXPRESS, IMPLIED, STATUTORY OR OTHERWISE, INCLUDING, WITHOUT LIMITATION, WARRANTIES OF TITLE, MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, NONINFRINGEMENT, OR THE ABSENCE OF LATENT OR OTHER DEFECTS, ACCURACY, OR THE PRESENCE OF ABSENCE OF ERRORS, WHETHER OR NOT DISCOVERABLE. SOME JURISDICTIONS DO

NOT ALLOW THE EXCLUSION OF IMPLIED WARRANTIES, SO SUCH EXCLUSION MAY NOT APPLY TO YOU.

6. Limitation on Liability. EXCEPT TO THE EXTENT REQUIRED BY APPLICABLE LAW, IN NO EVENT WILL LICENSOR BE LIABLE TO YOU ON ANY LEGAL THEORY FOR ANY SPECIAL, INCIDENTAL, CONSEQUENTIAL, PUNITIVE OR EXEMPLARY DAMAGES ARISING OUT OF THIS LICENSE OR THE USE OF THE WORK, EVEN IF LICENSOR HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

7. Termination

1. This License and the rights granted hereunder will terminate automatically upon any breach by You of the terms of this License. Individuals or entities who have received Collections from You under this License, however, will not have their licenses terminated provided such individuals or entities remain in full compliance with those licenses. Sections 1, 2, 5, 6, 7, and 8 will survive any termination of this License. 2. Subject to the above terms and conditions, the license granted here is perpetual (for the duration of the applicable copyright in the Work). Notwithstanding the above, Licensor reserves the right to release the Work under different license terms or to stop distributing the Work at any time; provided, however that any such election will not serve to withdraw this License (or any other license that has been, or is required to be, granted under the terms of this License), and this License will continue in full force and effect unless terminated as stated above.

8. Miscellaneous

1. Each time You Distribute or Publicly Perform the Work or a Collection, the Licensor offers to the recipient a license to the Work on the same terms and conditions as the license granted to You under this License.

2. If any provision of this License is invalid or unenforceable under applicable law, it shall not affect the validity or enforceability of the remainder of the terms of this License, and without further action by the parties to this agreement, such provision shall be reformed to the minimum extent necessary to make such provision valid and enforceable.

3. No term or provision of this License shall be deemed waived and no breach consented to unless such waiver or consent shall be in writing and signed by the party to be charged with such waiver or consent.

4. This License constitutes the entire agreement between the parties with respect to the Work licensed here. There are no understandings, agreements or representations with respect to the Work not specified here. Licensor shall not be bound by any

additional provisions that may appear in any communication from You. This License may not be modified without the mutual written agreement of the Licensor and You.

5. The rights granted under, and the subject matter referenced, in this License were drafted utilizing the terminology of the Berne Convention for the Protection of Literary and Artistic Works (as amended on September 28, 1979), the Rome Convention of 1961, the WIPO Copyright Treaty of 1996, the WIPO Performances and Phonograms Treaty of 1996 and the Universal Copyright Convention (as revised on July 24, 1971). These rights and subject matter take effect in the relevant jurisdiction in which the License terms are sought to be enforced according to the corresponding provisions of the implementation of those treaty provisions in the applicable national law. If the standard suite of rights granted under applicable copyright law includes additional rights not granted under this License, such additional rights are deemed to be included in the License; this License is not intended to restrict the license of any rights under applicable law.

Creative Commons Notice

Creative Commons is not a party to this License, and makes no warranty whatsoever in connection with the Work. Creative Commons will not be liable to You or any party on any legal theory for any damages whatsoever, including without limitation any general, special, incidental or consequential damages arising in connection to this license. Notwithstanding the foregoing two (2) sentences, if Creative Commons has expressly identified itself as the Licensor hereunder, it shall have all rights and obligations of Licensor.

Except for the limited purpose of indicating to the public that the Work is licensed under the CCPL, Creative Commons does not authorize the use by either party of the trademark "Creative Commons" or any related trademark or logo of Creative Commons without the prior written consent of Creative Commons. Any permitted use will be in compliance with Creative Commons' then-current trademark usage guidelines, as may be published on its website or otherwise made available upon request from time to time. For the avoidance of doubt, this trademark restriction does not form part of this License.

Creative Commons may be contacted at <http://creativecommons.org/>.

Copyright (C) 2008,2009,2010,2011 by
Ingenieurbuero fuer Telekommunikations und Software-Systemloesungen
Arno-Can Uestuensoez

<https://arnocan.wordpress.com/>

<http://ctys.sourceforge.net/>

Licenses:

Software: GPL3

Basic-Documents: FDL-1.3 - with invariant sections - The whole document!

Concepts+Interfaces+Documents: Creative Commons License-3.0 - cc
BY-NC-ND