m23 software distribution

m23 Development Guide
for m23 rock 16.2

Hauke Goos-Habermann

June 4, 2016

Contents

Chapter 1

introduction

Welcome to the m23 development guide. This is a not (yet) finished document because m23 isn't completed yet. You will find useful information about the m23 interna. If you want to develop for m23 this is the right document for you ;).
If you don't know what m23 is, you'll get a short answer. m23 will help you to set up hundreds of clients from one place. m23 can partition and format clients, install an operating system and additional programs. With m23 you can manage your clients and keep them up to date. For more information have a look at the m23 user guide.
This guide is meant for developers and people who want to know how m23 works only.

1.1 What you can expect from this document:

- an API reference about all functions used in the m23admin GUI and packages. This will be useful if you want to make changes to m23, build addons or plugins.
- information about serveral tools developed for m23. The little tools called "m23 helpers" make m23 work. Without them m23 can't do its job. You will learn how these tools work and how to use them.

1.2 What you can't exspect from this document:

- a 100% description of all functionality of m23. m23 is still in development, things are changing rapidly, so don't expect too much actuality.
- correct english ;) But I think it is written in a way most people will be able to understand. Don't expect a poem ;)

Have fun ;)

Chapter 2

m23 license: The GNU General Public License

Version 2, June 1991

59 Temple Place - Suite 330, Boston, MA 02111-1307, USA

Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change free software—to make sure the software is free for all its users. This General Public License applies to most of the Free Software Foundation's software and to any other program whose authors commit to using it. (Some other Free Software Foundation software is covered by the GNU Library General Public License instead.) You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

The precise terms and conditions for copying, distribution and modification follow.

TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The "Program", below, refers to any such program or work, and a "work based on the Program" means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term "modification".) Each licensee is addressed as "you".

   Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.

1. You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.

   You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

   (a) You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.

   (b) You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.

   (c) If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

   These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

   Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.

   In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:

(a) Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

(b) Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

(c) Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.

6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

   If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

   It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of

the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

8. If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

9. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

   Each version is given a distinguishing version number. If the Program specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.

10. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

Appendix: How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

one line to give the program's name and a brief idea of what it does.
Copyright (C) yyyy name of author

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 2 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, write to the Free Software Foundation, Inc., 59 Temple Place - Suite 330, Boston, MA 02111-1307, USA.

Also add information on how to contact you by electronic and paper mail.
If the program is interactive, make it output a short notice like this when it starts in an interactive mode:

Gnomovision version 69, Copyright (C) yyyy name of author
Gnomovision comes with ABSOLUTELY NO WARRANTY; for details type 'show w'.
This is free software, and you are welcome to redistribute it under certain conditions; type 'show c' for details.

The hypothetical commands `show w` and `show c` should show the appropriate parts of the General Public License. Of course, the commands you use may be called something other than `show w` and `show c`; they could even be mouse-clicks or menu items—whatever suits your program.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a "copyright disclaimer" for the program, if necessary. Here is a sample; alter the names:

Yoyodyne, Inc., hereby disclaims all copyright interest in the program
'Gnomovision' (which makes passes at compilers) written by James Hacker.

signature of Ty Coon, 1 April 1989
Ty Coon, President of Vice

This General Public License does not permit incorporating your program into proprietary programs. If your program is a subroutine library, you may consider it more useful to permit linking proprietary applications with the library. If this is what you want to do, use the GNU Library General Public License instead of this License.

Chapter 3

MDK menue system

3.1 Introduction

The MDK has got a menue system to make it easier for you to change things and to create your own m23 version, boot CDs etc.. In earlier versions of the MDK there were some widespread scripts without an user interface. The new menues make it easier to find what you are searching for. The menue system should be self-explanatory now.

Some of the things you can do with the MDK menues:

- Create server installation CDs: Build the operating system data file, the m23 programm file, the MDK file and make an installable boot CD from it.
- Create bootimages for the clients: Build netboot images for PXE and Etherboot standard and create bootable client installation CDs.
- Build special Debian packages: These packages are needed to add missing functionalities to the clients.
- Build documentation: Tools for generating the "m23 manual" in different languages and the "Development guide". This includes generation of screenshots, PDF and HTML files. The text is fetched from the online help, source codes and other sources.

3.2 Starting the MDK menue system

All you have to do is to run the following command from a console:

```
/mdk/bin/menuStart
```

Hint: If this doesn't work install the **dialog** package, that is needed to draw the menues.

Chapter 4

database

4.1 overview

m23 stores all information about the clients, their status, pending jobs and hardware infos in a database. This chapter will show you how the m23 database is organized. The m23 database is divided in several tables:

- clientjobs: stores waiting and done jobs for each client. If you install a package to a client the information about the install job is stored in this table with the status waiting. When the package is installed the install job will be marked as "done".
- clientlogs: Here are stored error and success of the installation procedure. You can see the output of whole installation procedure.
- clientpackages: here are stored all packages installed on the clients with status, version and action.
- clientpreferences: to make adding a new client more easy you can save preferences for new clients. These preferences are stored here.
- clients: information about all clients, ip, ram size, cpu, etc.
- ftpusers: will later be used for access to the ftp server on the m23 server.
- groups: clients can be organized in groups to manage them more easy.
- packages: will be used for storing available packages that can be installed on the clients.
- plugins: information about installed plugins.
- recommendpackages: here you can store package selections. With a package selection you can install a bundle of software with one click on a client.
- remotevar: table to store the remove vars.

4.2 the tables

4.2.1 clientjobs

If you install or deinstall software on the client, this (de)install job will be saved in the clientjobs table. The new job is added with the status "waiting" and after succesful finish it will get the status "done". Every job can have serveral parameters, e.g. if you want to format a partition, the format job should know which partition to format. This infomation is stored as the parameter. And of course each job should know the name of the client it is for. To get the jobs in the correct execution order, there are two values: id and priority. Priority has to do with the type of the job, a special job like the "hardware scan" gets the priority 0 and should be executed before all other jobs. With the priority the order of execution is set, lower priorities

are executed earlier. The second value "id" is the order of job creation. Jobs of the same priority will get executed in the order of creation. Remember: priority is mightier than the id value. If the priority is lower the job will be executed earlier while the id might be higher.

- id: the id of the job
- client: name of the client, the job is for.
- package: name of the package
- priority: the priority of the package
- status: status of the job: waiting, done
- params: the parameters for the job

4.2.2 clientlogs

clientslogs saves the information about the installation of clients. The input is generated by the log2db tool and PHP scripts.

- client: name of the client, the log information is saved for.
- logtime: time the log event was.
- status: contains the logged information.

4.2.3 clientpackages

- clientname: name of the client, the package is installed on
- package: name of the package
- version: version of the package
- status: is status status and can be "install ok installed", "deinstall ok config-files" and every other status, debian packages can have.
- action: action tells what should be done with the package. Possible actions can be: none, remove and reinstall.

4.2.4 clientpreferences

A preference can store multiple variables with its values, all of these variables are stored with the same preference name.

- name: name of the preference
- var: name of preference variable
- value: value for the variable

4.2.5 clients

The clients table stores information about hardware, the network settings, the username, email, etc. .

- client: the name of the client
- office: here you can leave information about the place where the client stands.
- name: name of the user
- familyname: familyname of the user

- eMail: eMail address for the user
- mac: mac address of the network card
- ip: ip address
- netmask: netmask for the ip
- gateway: gatewas address
- dns1: ip address of the first domain name server
- dns2: ip address of the second domain name server
- groupname: name of the group the client is in
- firstpw: the password for the first login
- rootPassword: root password
- memory: size of the installed memory in MB
- hd: size of harddisk in MB
- partitions: data about the partitions
- cpu: type of cpu
- MHz: speed of the cpu
- netcards: product names of the installed network cards
- graficcard: information about the grafic card
- soundcard: name of the sound card
- isa: information about ISA components
- dmi: DMI information
- dhcpBootimage: name of the currently used bootimage
- installdate: date the clients first was set up
- lastmodify: date the client was last modified
- status: actual status of the client. 0: client has not finished the hardware detection sequence. 1: client has finished hardware detection and waits for partition/format job 2: the client is partitioned and formated and has installed the base system.

4.2.6 ftpusers

This table can store information about ftpusers with their permissions.

- Password: password for th user
- Uid: user ID
- Gid: group ID
- Dir: directory for the user
- QuotaFiles: quota for the amount of files.
- QuotaSize: quota for the whole size of all files
- ULRatio / DLRatio: ratio for upload to download, you have to set both values

- ULBandwidth: max speed for upload
- DLBandwidth: max speed for download
- User: name of the FTP user

4.2.7 groups

With groups you will be able to organize your clients more efficient. You can add a client to agroup and install software on all clients of a group without selecting each of the clients. It just saves some clicks ;)

- groupname: name for the group

4.2.8 plugins

With plugins you can enrich you m23 admin console with additional functionalities. Plugins are a bundle of PHP/Bash and other files showing one or more dialogs in the m23 admin, that are designed for a special purpose. E.g. you can write a backup plugin that lets you backup all m23 clients. For more information abou plugins see the "How to develop plugins for m23?" chapter.

- name: name of the plugin
- author: who had done it?
- version: version number
- updateurl: where to get the update file. An update file contains information about the new plugin and the plugin data itself.
- clientRequires: packages that have to be installed on the client before you can use the plugin. E.g. if you install a backup plugin there shoulb be installed the backup software on the client.
- deinstall: here is stored the uninstall script, this is normaly a Bash script.
- files: the file names included in the plugin, this is saved for clean uninstall.
- installdate: when was it installed?

4.2.9 recommendpackages

In the recommendpackages table are stored package selections for reuse at a later moment. E.g. you may save a selection containing OpenOffice, Mozilla and Gimp for office usage. Now you can install these three packages with the selection and don't have to install each of them.

- name: name of your selection (e.g. office)
- package: the name of the package in the selection (e.g. openoffice.org)
- version: may be used later if we have to select between different versions of a package.
- priority: the priority of a package selects when to install the package among other packages in the whole installation process. Packages with lower numbers are installed earlier.
- params: special parameters for the package.

4.2.10 remotevar

With remote variables you can store values server side. The variables are stored for a special ip.

- ip: the ip address the variable is stored for.
- var: name of the variable
- value: the value for the variable
- addtime: the time the value was changed / added

Chapter 5

HowTos

5.1 Using new Debian/Ubuntu releases with m23

To fully support a new Debian/Ubuntu release a few steps are needed. The following guide shows a generic approach to not miss an important step. The total time and difficulty of a new release depends on the changes the distribution made between the last release supported by m23 and new release and if there are new desktops etc. that should be supported.

5.1.1 Test and development

- Build a compressed root file system for the new release via /mdk/m23helper/compressedDebootstrap and put it (for testing) into the directory /m23/data+scripts/packages/baseSys on the m23 server.
- Create an empty file (touch) with the release name of the new distribution release under /m23/data+scripts/distr/debian/d
- Add the new release to the list in DISTR$_{r}eleaseVersionTranslator$.

- Search all occurrences with the release name of the old version in the PHP files to get hints for finding places where new hacks or adaptions need to be made.
- Check, if all desired desktop environments are present in the file /m23/inc/distr/<distribution>/info.txt and add missing desktops.
- Write a new package source list (based on a previous release) in the m23 webinterface and choose the release with the name of the previously touched file name. Hook all desktops that should be deployed with this release.
- Make a base client install, see errors and fix them ;-)
- Develop missing desktop installation scripts under /m23/inc/distr/<distribution>/packages/m23<desktop>Install.php.
- Then make an installation/test/fix run with all desktop envirtonments that should be supported by the release.

5.1.2 Building the packages

- Generate the package template files with /mdk/m23helper/getDebianTemplates and run /mdk/m23helper/template2confp in the directory containing the template files. Afterwards move the *OptionPage.php files from the subdirectory "out" to /m23/data+scripts/m23admin/packages/<distribution>/<release>/.
- Add the sources list name in /mdk/bin/exportDBsourceslist.php.
- Build new packages via the MDK.

5.2 Releasing a new m23 version

5.2.1 Switch to relase

- Run /mdk/bin/menuStart $\Rightarrow fork \Rightarrow moveDevel2Release$

5.2.2 Documentation

- Translate all new and changed texts (/m23/inc/help/* and /m23/inc/i18n/*)
- Check for missing I18N variables. Run /mdk/doc/manual/bin/checkForMissingi18n.sh
- Manual
  - Check, if all parts seem to be there. Run: /mdk/doc/manual/bin/checkForMissingHlp.sh
  - Check, if all HTML entities are in the HTML to LaTeX translation index. Run: /mdk/doc/manual/bin/checkForMissingHT
  - Check, if there are all screenshots present. Run: /mdk/doc/manual/bin/checkMissingScreenshots.sh
  - Generate and update the screenshots
  - Run /mdk/bin/menuStart $\Rightarrow doc \Rightarrow manualChoosethelanguage :\Rightarrow lang.Chooseall$

– Generate the tex files: $\Rightarrow textOptimisethePNGs. \Rightarrow optimisePNGs$
– Generate PDF and HTML version of the manual: $\Rightarrow pdf - htmlUpload :\Rightarrow upload$
– Go back

- Development guide
  - Add (maybe existing new) LaTeX files
  - Build the development guide: $\Rightarrow devguideGenerateit :\Rightarrow generate$
- Upload: $\Rightarrow upload$

5.2.3 Building the files

- Debian packages

– /mdk/bin/menuStart $\Rightarrow debs \Rightarrow buildUploadthepackages, ifallseemstoberight :\Rightarrow directuplinst$

- Server installation ISO

– /mdk/bin/menuStart $\Rightarrow serverISO \Rightarrow iso.Exitafterbuilding.Upload : cd/mdk/server; ./upiso.file$

- VirtualBox appliance
  - Create a new VM (Linux/Debian, 512 MB Ram, growing VDI 16GB HDD
  - Change the Network card to Network bridge
  - Start the VM and "insert" the ISO into the virtual CD drive
  - Select "English".
  - Use "m23s" as server name, "test" as password and local network settings.
  - Choose the automatic partitioning and formating.
  - Call /mdk/bin/prepareOSForCompression after reboot.
  - Shutdown the VM
- Export it (m23server$_x x.y_r ock.ova$)$andcompress(7zm/7zr-t7z-m0 = lzma-mx = 9-mfb = 64 - md = 32m - ms = onam23server$_x x.y_r ock.7zm23server_x x.y_r ock.ova)theexportedfiles.$

- – Write a SD card with Raspbian.
  - Configure /etc/network/interfaces to static IP for eth0.

  - Boot the Raspberry Pi.
  - Log into it via SSH (user: pi, password: raspberry)
  - Run "sudo raspi-config".
  - Call in the menu "Expand Filesystem", set "Advanced Options" "Memory split" to 16 MB, call "Update".
  - Reboot.
  - Enable root access (set password to test): "sudo passwd"
  - Enable root login for SSH in /etc/ssh/sshd$_{c}onfigLogoutandlogintoitviaSSH(user : root, password : test)$
- Remove the user pi: userdel -r -f pi
- Add "192.168.1.77 m23debs" to /etc/hosts
- Create "/etc/apt/sources.list.d/m23debs.list" with the contents "deb http://m23debs ./"
- Update the package index: apt-get update
- Install the m23 server: apt-get install m23
- Choose yes on all options and use "test" for all passwords.
- After the failure edit /etc/default/tftpd-hpa: Adjust TFTP$_{O}PTIONS = "-4--secure"Continuetheinstallationwith : apt - getinstall - f$
- Run /mdk/m23helper/Raspbian-RemoveUnneededPackages.sh
- /mdk/bin/prepareOSForCompression.

5.2.4 CMS

- Write an article
- Create new link files and move the old to the old directories.
- Check in into Bazaar
- Upload the files.

5.2.5 Announcement

- Prepare the newsletter with /mdk/doc/newsletterGenerator3.sh and send it.

5.3 How to translate m23?

m23 uses a system to make translation to other languages easy. All language specific text is stored in files. One big file (m23base.php) contains the text shown in the m23admin menus, messages, on buttons, ... The help texts are stored in single files containing a topic each. To make your translation available, you have to give the new language a name and store it in a lang file.

5.3.1 make directories

You should think about a good abreviation for the language. All m23 languages have a short 2 letter name that is used for directory name (e.g. de=german, en=english, fr=french).

Make the directories:

```
/m23/inc/i18n/<your language short name>
/m23/inc/help/<your language short name>
```

if you use a console to create the directories it may look like this:

```
mkdir /m23/inc/i18n/de
mkdir /m23/inc/help/de
```

5.3.2 generate the language file

To give your language a name and make it available to m23 create a *language.info* file.
The language.info has to contain the following lines:

```
language: <the full name of your language>
shortlanguage: <the short 2 letter name of the language>
```

m23 uses the word for the language that is used in the origin country. e.g. Deutsch and not German, Francais and not French.

Your language.info may look like this:

```
language:Deutsch
shortlanguage:de
```

5.3.3 translating the messages

Simply copy the m23base.php file from your preferredly understood language to the directory of your translation. e.g. if you want to translate the english version to german: copy /m23/inc/i18n/en/m23base.php to /m23/inc/i18n/de/m23base.php.
Now translate the text between the ' " ' s. e.g.

```
$I18N_help="Help";
```

becomes

```
$I18N_help="Hilfe";
```

Please don't delete any other text and make changes only between the ' " ' letters. Don't translate something like $I18N_help to $I18N_hilfe. If you do so, m23 can't find it and will leave the place, the text should be appear empty. For your information: m23 stores the texts as variables that are inserted at the right places in the m23admin interface.

5.3.4 translating the help texts

Copy all *.hlp from the /m23/inc/help/<lang> directory to your new help directory. e.g. : copy /m23/inc/help/en/*.hlp to /m23/inc/help/de/. Translate the text in each file. If you want, you can use HTML characters. e.g. "«" are used in french texts, these characters are interpreted by HTML as begin of tag. You have to replace these characters with the HTML equivalent: "«" becomes "<<".

5.4 How to generate a new base system?

5.4.1 install necessary system files

```
debootstrap --arch=i386 woody .
```

This may run in to an error, but it doesn't matter. Simply follow this guide ;) .

5.4.2 editing files

fstab

Copy your existing /etc/fstab to the new /etc/ directory. Adjust the line for your boot device, e.g. your new boot device is hda3 so you have to change */dev/hda2 / ext3 defaults 0 0* to */dev/hda1 / ext3 defaults 0 0* .
Your new fstab may look like this:

```
# /etc/fstab file system info, created by the m23 project
#  <file system> <mount point>  <type> <options>     <dump>  <pass>
proc             /proc          proc   defaults      0       0
/dev/hda1        none           swap    sw           0       0
/dev/hda3       /               ext3   defaults           0       0
/dev/cdrom      /mnt/cdrom      auto    ro,noauto,user,exec     0 0
/dev/fd0        /mnt/floppy     auto    user,noauto             0 0
```

lilo.conf

You can use your current lilo.conf and edit some values like *root*. You may choose a different kernel than in your current system so you have to adjust the *image* parameter too. But first it is essetial to have a lilo.conf with the correct *root* value, *image* doesn't matter at the moment.

```
boot=/dev/hda
map=/boot/map
install=/boot/boot-menu.b
compact
prompt
vga=normal
delay=10
timeout=20

default=m23Server3

image=/boot/vmlinuz-2.4.20-1-386-sec
        root=/dev/hda3
        label=m23Server3
        append="hdb=scsi-ide"
        initrd=/initrd.img
```

sources.list

Your sources.list should look like this:

```
deb http://security.debian.org/ woody/updates main contrib non-free
deb http://ftp.de.debian.org/debian/ stable main non-free contrib
deb http://ftp.szczepanek.de/ftp/trusteddebian/ stable main contrib
```

You can add more sources if preferred.

/etc/network

Simply copy your old /etc/network directory to the new /etc.

5.4.3 step in to your new system

You can enter your new system (without reboot) with:

```
chroot .
```

5.4.4 source update and cleanup

I think we don't need mailx :) .

```
apt-get update
apt-get remove mailx
```

5.4.5 installing a kernel

You can list all available kernels with the following command:

```
apt-cache search kernel | grep kernel-image
```

Pick one out adjust the *image* value in your new *lilo.conf*. To install your new kernel type the following:

```
apt-get install myChosenKernel-Image
```

in case of an error

If you get an error with the *mkinitrd* command try it manualy. Adjust */boot/initrd.img-2.4.20-1-386-sec* to the filename you entered in the *lilo.conf*. You have to adjust the */lib/modules/...* and the root device after *-r* also.

```
mkinitrd -r /dev/hda3 -o /boot/initrd.img-2.4.20-1-386-sec /lib/modules/2.4.20-1-386-sec
```

5.4.6 system update and installation of additional packages

```
apt-get dist-upgrade -u
apt-get install ssh
```

5.4.7 booting your new system

To see if your new system is ready to boot, copy your new kernel from the new boot/ directory to /boot. This kernel will be called like *vmlinuz-2.4.20-1-386-sec*. Edit your **current** /etc/lilo.conf and add a new section like this:

```
image=/boot/vmlinuz-2.4.20-1-386-sec
        root=/dev/hda3
        label=m23Server3
        initrd=/initrd.img
```

vmlinuz-2.4.20-1-386-sec should be replaced with the name of your kernel and the section above has to be the same as in your new *lilo.conf*.

5.5 How to create a bootable server installation CD?

5.5.1 Introduction

If you want to change the server and generate a new installation CD you simply have to execute the makeCD.sh script from the /mdk/bootCD directory.

5.5.2 what does the makeCD script?

- generates a new initrd files used for booting from the CD
- compresses the server operating system
- compresses the m23 files
- generates the ISO image to burn on a CD

initrd

While booting from the CD the kernel is started, if the kernel starts up it loads an initial file system from a file, this is the initrd (here called rescue). This file is compressed and will be extracted to the memory and mounted as root file system. All needed files like basic commands (ls,cp) and the install script are contained in this root file system.
In the /mdk/bootCD/root are all files needed for the initrd. If you want to make changes to the start up process of the boot CD you should edit the /mdk/bootCD/root directory. The mkRescue.sh script makes an empty initrd file, formats it with ext2 file system, copies needed files to initrd and compresses it to rescue.gz. For further information have a look at the mkRescue.sh script.

compress operating system

The whole current operating system is compressed mith bzip2 to a file called m23image.tb2. Some directories aren't saved, because they aren't needed or will mess up the image with useless files. E.g. /proc, /tmp, ... should not be saved. The image is stored in the /mdk/bootCD/iso directory. If there is allready a m23image.tb2 the operating isn't compressed. To regenerate the m23image.tb2 you have to delete the file.

compresses the m23 files

The whole /m23 directory is compressed to /mdk/bootCD/iso/m23.tb2, if this file allready exists the /m23 system isn't recompressed. To force recompress simply delete /mdk/bootCD/iso/m23.tb2. Before compression you should clean the /m23 directory from test scripts, dhcp config files and database from all entries.
For cleaning the m23 database: Delete all *.MYD files in /m23/db/m23/ and replace them with empty files. (touch tablename.MYD).

generating the ISO image

After these steps the ISO image will be created automatical in /mdk/bootCD directory. Now you can write the iso file to a CD. This should generate a bootable CD. Good luck ;)

5.6 How to make network bootimages?

5.6.1 Introduction

What are bootimages? Bootimages are used to boot up a client over the network. The bootimage contains a Linux kernel, an initial ramdisk and a script for fetching the jobs for the client.
A client runs for first start up thru the following sequence:

- get IP, netmask, bootimage name, ... from the m23 DHCP server
- fetch the bootimage from the server with TFTP protocoll
- load and extract the image to the client memory
- start the contained kernel
- kernel loades the included initrd
- start up script fetches IP, netmask, etc. from the server again
- the script fetches the first job for the client and saves the job script file
- the script is executed and the next job will be fetched from the server

This technology makes it possible to install angelOne Linux on an empty computer. No installed operating system is needed for partition, format or installation.

5.6.2 creating a new bootimage

In the /mdk/bootimage directory you can find the mkBootImage.sh script that generates the bootimage for PXE and Etherboot standard. The bootimages will be stored in /m23/tftp/ as m23pxeinstall (PXE kernel), initrd.gz (initial ramdisk for PXE) and m23install (Etherboot). To generate the Etherboot files you need to install **mknbi**.

How it works?

it creates a blank file of 17MB size

```
> dd if=/dev/zero of=initrd bs=1k count=23000
```

sets up a loop device

```
> losetup /dev/loop1 initrd
```

the 23MB file will be formated with EXT2

```
> mke2fs -m 0 -N 10000 /dev/loop1
```

mount the file

```
> mount /dev/loop1 mnt
```

copy needed files from root2 and hardware informations to the mounted image

```
> cp -rdpR root2/* mnt/
> cp -rdpR /usr/share/hwdata/*  root2/usr/share/hwdata/
```

umnount the image

```
> umount mnt
```

set "down" the loop device

```
> losetup -d /dev/loop1
```

set correct file permissions

```
> chmod 0555 bzImage
> chown root:root bzImage
```

set boot device in kernel

```
> rdev bzImage /dev/ram0
```

generate bootimage for Etherboot

```
> mknbi-linux bzImage --first32pm --output=/m23/tftp/m23install --ip=dhcp
--rootdir=/dev/ram0 initrd
```

generate files for PXE

```
> cp bzImage /m23/tftp/m23pxeinstall
> gzip initrd
> mv initrd.gz /m23/tftp/initrd.gz
```

5.6.3 What can you do with this SDK?

Modify all files in root2. These files are the files for a kind of Mini Linux distribution. You should be familar to Linux, if you want to change a thing. linurc will be the first script executed after network boot. If you want to do automatic execution this is the right place.

Build a new kernel

Copy your new kernel to bzImage in the SDK directory and don't forget to copy the modules to root2/lib/modules. In the m23client-Install*.conf file you get the configuration for the kernel we used.

5.7 How to generate an Etherboot bootimage for booting from harddisk?

If the clients is set up, we want to boot it from harddisk. To do so, we send a little bootimage that tells the client to boot from harddisk directly. This seems the easiest way right now.

5.7.1 exchange epic100.c with the special boot code

All Etherboot drivers have the option to boot (in case of an error or user intervention) from harddisk. To make harddisk booting the default starting method and remove the network card specific code, we have to patch one of the Etherboot drivers. I choosed the epic100 driver because of its size ;)

Here you can see how its done:

```
#ifdef ALLMULTI
#error multicast support is not yet implemented
#endif
/* epic100.c: A SMC 83c170 EPIC/100 fast ethernet driver for Etherboot */

#define LINUX_OUT_MACROS

#include "etherboot.h"
#include "nic.h"
#include "cards.h"
#include "timer.h"
#include "epic100.h"

#undef  virt_to_bus
#define virt_to_bus(x)  ((unsigned long)x)

#define TX_RING_SIZE    2       /* use at least 2 buffers for TX */
#define RX_RING_SIZE    2

#define PKT_BUF_SZ      1536    /* Size of each temporary Tx/Rx buffer.*/

/*
#define DEBUG_RX
#define DEBUG_TX
#define DEBUG_EEPROM
*/

#define EPIC_DEBUG 0    /* debug level */

/* The EPIC100 Rx and Tx buffer descriptors. */
struct epic_rx_desc {
    unsigned short status;
    unsigned short rxlength;
    unsigned long  bufaddr;
    unsigned short buflength;
    unsigned short control;
```

```
    unsigned long  next;
};

/* description of the tx descriptors control bits commonly used */
#define TD_STDFLAGS     TD_LASTDESC

struct epic_tx_desc {
    unsigned short status;
    unsigned short txlength;
    unsigned long  bufaddr;
    unsigned short buflength;
    unsigned short control;
    unsigned long  next;
};

#define delay(nanosec)   do { int _i = 3; while (--_i > 0) \
                                  { __SLOW_DOWN_IO; }} while (0)

static void     epic100_open(void);
static void     epic100_init_ring(void);
static void     epic100_disable(struct nic *nic);
static int      epic100_poll(struct nic *nic);
static void     epic100_transmit(struct nic *nic, const char *destaddr,
                                 unsigned int type, unsigned int len, const char *data);
static int      read_eeprom(int location);
static int      mii_read(int phy_id, int location);

static int      ioaddr;

static int      command;
static int      intstat;
static int      intmask;
static int      genctl ;
static int      eectl  ;
static int      test   ;
static int      mmctl  ;
static int      mmdata ;
static int      lan0   ;
static int      rxcon  ;
static int      txcon  ;
static int      prcdar ;
static int      ptcdar ;
static int      eththr ;

static unsigned int     cur_rx, cur_tx;         /* The next free ring entry */
#ifdef  DEBUG_EEPROM
static unsigned short   eeprom[64];
#endif
static signed char      phys[4];                /* MII device addresses. */
static struct epic_rx_desc      rx_ring[RX_RING_SIZE];
static struct epic_tx_desc      tx_ring[TX_RING_SIZE];
#ifdef  USE_LOWMEM_BUFFER
#define rx_packet ((char *)0x10000 - PKT_BUF_SZ * RX_RING_SIZE)
#define tx_packet ((char *)0x10000 - PKT_BUF_SZ * RX_RING_SIZE - PKT_BUF_SZ * TX_RING_SI
#else
static char             rx_packet[PKT_BUF_SZ * RX_RING_SIZE];
```

```
static char             tx_packet[PKT_BUF_SZ * TX_RING_SIZE];
#endif

/*************************************************************************/
/*                      Externally visible functions                     */
/*************************************************************************/

    static void
epic100_reset(struct nic *nic)
{
    /* Soft reset the chip. */
    outl(GC_SOFT_RESET, genctl);
}

    struct nic*
epic100_probe(struct nic *nic, unsigned short *probeaddrs)
{
 return 0;
}

    static void
epic100_open(void)
{
}

/* Initialize the Rx and Tx rings. */
    static void
epic100_init_ring(void)
{
}

/* function: epic100_transmit
 * This transmits a packet.
 *
 * Arguments: char d[6]:          destination ethernet address.
 *            unsigned short t:   ethernet protocol type.
 *            unsigned short s:   size of the data-part of the packet.
 *            char *p:            the data for the packet.
 * returns:   void.
 */
    static void
epic100_transmit(struct nic *nic, const char *destaddr, unsigned int type,
                 unsigned int len, const char *data)
{
}

/* function: epic100_poll / eth_poll
 * This receives a packet from the network.
 *
 * Arguments: none
 *
 * returns:   1 if a packet was received.
 *            0 if no pacet was received.
 * side effects:
 *            returns the packet in the array nic->packet.
 *            returns the length of the packet in nic->packetlen.
```

```
 */

    static int
epic100_poll(struct nic *nic)
{
    return 0;
}


    static void
epic100_disable(struct nic *nic)
{
}


#ifdef  DEBUG_EEPROM
/* Serial EEPROM section. */

/*  EEPROM_Ctrl bits. */
#define EE_SHIFT_CLK    0x04    /* EEPROM shift clock. */
#define EE_CS           0x02    /* EEPROM chip select. */
#define EE_DATA_WRITE   0x08    /* EEPROM chip data in. */
#define EE_WRITE_0      0x01
#define EE_WRITE_1      0x09
#define EE_DATA_READ    0x10    /* EEPROM chip data out. */
#define EE_ENB          (0x0001 | EE_CS)

/* The EEPROM commands include the alway-set leading bit. */
#define EE_WRITE_CMD    (5 << 6)
#define EE_READ_CMD     (6 << 6)
#define EE_ERASE_CMD    (7 << 6)

#define eeprom_delay(n) delay(n)

    static int
read_eeprom(int location)
{
 return 0;
}
#endif


#define MII_READOP      1
#define MII_WRITEOP     2

    static int
mii_read(int phy_id, int location)
{
 return 0;
}
```

Exchange the code from the epic100.c file (in the **src** directory) with the code above. Now you have tho change the default boot type. Exchange in *etherboot.h*

```
#ifndef ANS_DEFAULT /* in case left out in Makefile */
#define ANS_DEFAULT ANS_NETWORK
#endif
```

with

```
#define ANS_DEFAULT ANS_LOCAL
```

and

```
#define TIMEOUT (10*TICKS_PER_SEC)
```

with

```
#define TIMEOUT (1*TICKS_PER_SEC)
```

Now execute the following commands:

```
make clean
make
```

5.7.2 Making the bootimage

You should find the epic100.rom file in the **src/bin32**. With the help of the *mknbi-rom* program you can convert this from file to an Etherboot bootimage. To convert and save the bootimage as new "harddisk boot image" for m23 execute the following command:

```
mknbi-rom epic100.rom > /m23/tftp/hdboot
```

5.8 How to create the server update files?

m23 has a mechnism that makes it easy for the user to update the m23 server. The information for the update is fetched from the internet. There is a php script that generates the needed update information from single files. This files contain information about the new codename, the new version number, a describing text, a script that is exectued at the beginning and at the end of the update.

5.8.1 The files

All files for an update have to begin with the version number (e.g. 0.4.5):

- 0.4.5.begin: conatins the bas script that is executed a the beginning
- 0.4.5.end: conatins the bas script that is executed a the end
- 0.4.5.info: contains only the codename of the m23 release (e.g. shiver)
- 0.4.5.text: helds the decribing text of the update. There you should put a changelog

xxx.begin

This is only an example, don't assume it will work.

```
wget http://m23.sf.net/newdata.tb2
tar xfvj newdata.tb2
```

xxx.end

```
rm newdata.tb2
```

xxx.info

```
shiver
```

xxx.text

You can use html tags in your release information.

```
This is the new shiver update. There will be the following changes:
<ul>
 <li> bigger</li>
 <li> louder</li>
 <li> ...
</ul>
```

5.9 debugging m23 scripts

If you write your own m23 scripts for special jobs, you will need a kind of debugging solution.

5.9.1 Activate PHP debugging

You can activate the PHP warning messages if you call:

```
/m23/bin/phpDebug
```

to deactivate the debugging mode simply call:

```
/m23/bin/phpNormal
```

5.9.2 Turn on m23 script debugging

If you want to see the messages behind the blue status screens of the normal m23 client installation, you can turn on the debugging mode. You can do it by setting the debugging status for your client IP.

- call http://serverIP/m23admin/phpMyAdmin/ with your webbrowser
- select the database "m23"
- select the table "remotevar"
- click on "Browse"
- click on "Insert new row"
- enter the ip of your client in the "ip" field, "debug" in the "var" field and "1" in the "value" field.

5.9.3 Turn off m23 script debugging

You have two possibillities to deactivate the script debugging: the first solution is temporary the second is usefuly to stop debugging for a longer time.

1. Edit the row for your client in the "remotevar" table. Set the value from the "value" field to "0".
2. Delete the row of your client in the "remotevar" table.

5.9.4 Set script status

After the script is executed and has produced an error, you have to set the status of this script to "waiting" to repeat the execution.

- call http://serverIP/m23admin/phpMyAdmin/ with your webbrowser
- select the database "m23"
- select the table "clientjobs"

- click on "Browse"
- search for your job script
- click on "Edit"
- enter "waiting" at the "status" field
- click on "Go" to save your changes

This will work if the script you want to debug has the lowest priority and id among all waiting jobs of the client only.

5.9.5 get the output of your script

You should turn off your client and overtake its ip. Now you can get the generated script by:

```
wget http://serverIP/work.php
```

5.10 Additional installation parameters for normal packages

There are some cases where you want to make it possible to enter some additional parameters for normal Debian packages. Maybe you want to set the DocumentRoot directory of your Apache webserver. Wouldn't it be much easier to enter this value in a dialog box in the m23 webinterface and automate the Apache installation job? m23 has a possibillity to make it easy for the administrator with a little work of the programmer.

5.10.1 The *OptionPage.php

If you want to enable this feature for a special package (e.g. apache) you have to design an OptionPage. This page shows and writes/reads possible options of this package. The file is called <packagename>OptionPage.php. In our example it's **apacheOptionPage.php**. The file is placed in the */m23/data+scripts/distr/<distributionname>/packages directory*. It is important to put the *OptionPage.php in the correct directory, otherwise the it can't be found.

5.10.2 Filling the *OptionPage.php

To make it easier for you there are two functions that help you to generate your option page. To use these function you have to include */m23/inc/packages.php*

- *PKG_OptionPageHeader($title)*: For starting the page. It generates all necessary HTML code (colors, styles, tables, form) and expects a title for your option page. This title is shown in the window label of your webbrowser and in the window itself. It returnes an associative array with all parameters of the selected package. These values are used to initalise the OptionPage if it is opened for the first time.
- *PKG_OptionPageTail($layout)*: Renderes the layout of the page, adds a save button and closes the HTML page.
- *PKG_OptionPageGetValue($variable,$params)*: gets a value from the $_GET array or falls back to the params values. This is used if there havn't been entered values in the OptionPage.

Your OptionPage needs to have the following elements:

```
include ('/m23/inc/packages.php');

$params = PKG_OptionPageHeader("My OptionPage title");

$layout[0]...

PKG_OptionPageTail($layout);
```

5.10.3 Layoutoptions

The layout is stored in an array. Every element gets an numeric entry with several options. These entries have to be counted beginning by 0. The elements are marked by the **type** attribute. There are some different types of elements:

- *text* Is a simple text: In the exaple you can see the text is selected by the *type* of **text**. All types are case sensitive. If you want a text as first element, give it the index 0.

```
$layout[0]['type']="text";
$layout[0]['text']="HalloText";
```

- *line*: A line is even simpler, because it has no attributes. It only draws a horizontal line. Lines are graphical objects to make your OptionPage look better ;)

```
$layout[1]['type']="line";
```

- *inputline*: An inputline is a editable text field with the height of one. The inputline has a lot of attributes:
  - text: a text that is shown before the element. It is a good idea to put a name or a descriptive text here.
  - value: is the text, that stands in the inputline. You should follow the example and use the PKG_OptionPageGetValue function to get the value. Otherwise the value can't be gotten after saving and will disappear.
  - name: The name of the element, that's the same name the entered value will be stored under the params column in the packagejobs table. **This name has to be the same as in the PKG_OptionPageGetValue function under value. Otherwise the values can't be stored!!!**
  - size: The width of the inputline in characters.
  - maxlength: The maximum of characters that can be entered.

```
$layout[2]['type']="inputline";
$layout[2]['text']="documentRoot";
$layout[2]['value']=PKG_OptionPageGetValue('documentRoot',$params);
$layout[2]['name']="documentRoot";
$layout[2]['size']=10;
$layout[2]['maxlength']=100;
```

- *selection*: A selection gives the user a list with options to choose. As before the text attribute describes the element, name is the variable name to store it in the database, value the value to store. New is the option* attribute. The selectable options are stored under the attributes option0, option1, ... **It is important to start by 0 and to left no number out.** Otherwise the renderer will stop by the first hole in the count.

```
$layout[3]['type']="selection";
$layout[3]['text']="Desktop";
$layout[3]['name']="desktop";
$layout[3]['value']=PKG_OptionPageGetValue('desktop',$params);
$layout[3]['option0']="gnome";
$layout[3]['option1']="kde3";
$layout[3]['option2']="kde2";
$layout[3]['option3']="kde4";
```

- *textarea*: A text area with multiple colums and rows. The attributes type, name and value as usual. Cols (colums) and rows are the size parameters of the text area in characters.

```
$layout[4]['type']="textarea";
$layout[4]['cols']=20;
$layout[4]['rows']=20;
$layout[4]['name']="textedit";
$layout[4]['value']=PKG_OptionPageGetValue('textedit',$params);
```

Chapter 6

m23 helpers

6.1 mdoc

6.1.1 what is mdoc?

mdoc creates documentation in latex format out of comments in your source files. Special marked lines in the source code are extracted and converted to a documentation file. For an example of the generated documentation see the "m23 API reference" included in this document. mdoc can scan PHP and C/C++ files and other files that allow "/**" and "**/" for begin and end of comments or don't care about these kind of strings. If you want to use mdoc for files that don't allow these strings, put the comment sequence used for the file type before the mdoc lines. e.g. for BASH scripts you will put a '#' in front of each mdoc line:

```
#/**
#**name helloworld.sh
#**description shows a hello world
#**parameter none
#**/
echo hello world
```

6.1.2 how to make your source code mdoc compatible?

To tell mdoc that it should search for comments mark the begin of the search area with "/**" and the end with "**/". You have tree comment tapes:

- **n : for the name, you should leave the name of the function with all parameters here
- **d : this is the description of the function. here you can write longer comments about the usage, restrictions, ...
- **p : deals with a single parameter used for the function. you should describe all parameters used to call the function with a "**p" line each.

6.1.3 mdoc info block

In the mdoc info block you can leave all information you want, e.g. you can write down your name, the function of the file etc. . This block is parsed by mdoc first and will appear in the documentation at the beginning of the chapter. A mdoc info block begins with "/*$mdocInfo" and ends with "$*/". All between these lines will be treated as a comment and is copied 1 to 1 to the documentation file. If you make a line break this line break will appear in the documentation too.
Attenction: For the 'I' in mdocInfo you have to use an upcase letter. Otherwise the mdoc info block will be ignored.
Here an example for a mdoc info block:

```
/*$mdocInfo
 Author: Daniel Kasten (DKasten@gss-netconcepts.de)
 Description: a lot of routines for client handling.
$*/
```

6.1.4 example for a mdoc comment

```
/**
**n CLIENT_listPackages($client, $key)
**d lists all packages on the client
**p client: name of the client
**p key: keyword for searching for packages
**/
```

6.1.5 using mdoc

```
usage:  mdoc <start directory> <tex output file>
```

- start directory: directory to start search for files that should be scanned for comments.
- tex output file: filename the latex output file should be saved to

6.1.6 example

```
mdoc /m23/data+scripts /tmp/m23api.tex
```

will scan the /m23/data+scripts directory and store the documentation in /tmp/m23api.tex.

Chapter 7

m23customPatch

The patch system "m23customPatch" makes it easy to change parts of m23 with user specific code. In the m23 source code are some m23customPatch range markers who define that this portion of code may be deleted or changes by a m23customPatch file.

If you need additional patchable areas in m23 feel free to contact me via http://goos-habermann.de or http://m23.sf.net .

Here is a short example of the file "/m23/data+scripts/m23admin/head/head.php" where the logo and link are replaced.

7.1 Indicating patchable areas in the source code

The start and end position of a patchable area are marked by comments (as used in the programming language the source file is written in). "m23customPatchBegin" is the keyword for the start of the patchable area, "m23customPatchEnd" for its end. Both keywords must be in different lines with "m23customPatchBegin" before "m23customPatchEnd". Patchable areas may not overlap.

7.1.1 Start position of a patchable area

- HTML notation: "<!−−m23customPatchBegin type=change id=logo−−>"
- PHP notation: "/*m23customPatchBegin type=change id=logo*/"
- PHP notation (alternativ): "//m23customPatchBegin type=change id=logo"
- BASH notation: "m23customPatchBegin type=change id=logo"

7.1.2 End position of a patchable area

- HTML notation: "<!−−m23customPatchEnd id=logo−−>"
- PHP notation: "/*m23customPatchEnd id=logo*/"
- PHP notation (alternativ): "//m23customPatchEnd id=logo"
- BASH notation: -Notationm23customPatchEnd id=logo"

The parameter "type" defines how the contents between start and end position may be changed:

- change: By running the m23customPatch file here, all code lines between the start and end position of a patchable area are replaced by the code lines of the m23customPatch file.
- del: By running the m23customPatch file here, all code lines between the start and end position will be deleted.

The parameter "id" is a unique identifier to find the correct patchable area. The ID may be uses only once in each source file and is written in the m23customPatch file too. This way, the patchable area and m23customPatch file are "linked".

7.1.3 Example (/m23/data+scripts/m23admin/head/head.php)

```
<!--m23customPatchBegin type=change id=logo-->
                        <a HREF="http://m23.goos-habermann.de" target="_blank">
                                <img src="head/m23.png" border="0">
                        </a>
<!--m23customPatchEnd id=logo-->
```

7.2 m23customPatch file format

The m23customPatch defines the ID to find the correct patchable area in the source file. For each patchable area a distinct m23customPatch file is required. The first line of a m23customPatch file contains the string "!m23customPatch" only. Lines 2 and 3 are containing the name of the source file (with full path) and the unique identifier (paramter "id"). The following lines are copied to a patchable area if its type is "change". In case of a "del" type area all lines in a m23customPatch file from the 4th on are ignored.

7.2.1 Example (logo.php.m23customPatch)

```


        <a href="http://mysite.local/" target="_blank"><img src="head/My-Logo.png"></a>
```

7.3 /m23/bin/m23customPatch

The script "m23customPatch" does the actual patching. The only command line parameter is the name of the m23customPatch file (with full path). If the patching worked well, a return code of 0 is given back. In case of an error a different return code is given back. Hint: The posting of your own Debian packages may be a good place to run "m23customPatch".

7.3.1 Return/error codes

- 1: Wrong parameter amount (!= 1)
- 2: m23customPatch file invalid
- 3: Source code file does not exist
- 4: The unique ID could not be found

Chapter 8

m23 API reference

In this chapter you will get an introduction to all m23 functions used in the m23admin user interface and for packages. If you want to write addons or plugins you should use the existing functions for faster programming. All functions are described with information about usage and parameters. This refernce is created by the mdoc tool that strips comments out of the source files and creates a documentation file. If you make changes to existing code please comment it in the mdoc way. So it is easy to generate documentation automatical. For introduction to mdoc see the chapter in this guide.

8.1 ./inc/assimilate.php

Author: Hauke Goos-Habermann (HHabermann@pc-kiel.de)
Description: Contains functions for assimilation of clients

8.1.1 ASSI_showClientAddDialog

Description: Shows a dialog for adding a client to assimilate.

8.1.2 ASSI_addClient

Description: Adds needed data for assimilating a client.
Parameter:

- client: name of the client
- ip: IP of the client
- password: root password on Debian systems or combines user/root password on Ubuntu systems
- ubuntuuser: name of the Ubuntu user or empty if a Debian system is meant.
- clientUsesDynamicIP: if set to true, the client uses a dynamic IP address

8.1.3 ASSI_addUbuntuRoot

Description: Enables the root account in Ubuntu if a Ubuntu installation is found.

8.1.4 ASSI_prepareClient

Description: Prepares a client for assimilisation.

8.2 ./inc/autoTest.php

Author: Hauke Goos-Habermann (HHabermann@pc-kiel.de)
Description: Functions for test automation.

8.2.1 AUTOTEST_getKey

Description: Generates the needed scan codes to produce a given character.
Parameter:

- charIn: Input character.

Returns:

- Needed scan codes to produce a given character.

8.2.2 AUTOTEST_calcScancodes

Description: Converts an input string that may contain special keys into scancodes (e.g. for usage with VirtualBox)
Parameter:

- in: Input string with normal and special keys.

Returns:

- Scancodes that represent the input string.

8.2.3 AUTOTEST_keyAndRelease

Description: Generates (Shift press,) key, key release (and Shift release) codes.
Parameter:

- keyCode: Key (scan) code.
- pressShift: true, when Shift should be pressed.

Returns:

- (Shift press,) key, key release (and Shift release)

8.2.4 AUTOTEST_VM_create

Description: Creates a new VM with virtual hard drive in VirtualBox.
Parameter:

- vmName: Name of the VM.
- diskSize: Size of the virtual HD in MB.
- ramSize: Size of RAM in MB.
- VMCreationMessage: Variable where the VirtualBox (error) messages will be written to.

Returns:

- true, when the creation was successfully, otherwise false.

8.2.5 AUTOTEST_VM_enableCapture

Description: Enables capturing the screen of a VM to a movie file.
Parameter:

- vmName: Name of the VM.
- movieFile: File to store the capturing in.
- VMenableCaptureMessage: Variable where the VirtualBox (error) messages will be written to.

8.2.6 AUTOTEST_VM_delete

Description: Deletes a VM and its virtual hard drive from VirtualBox.
Parameter:

- VMDeletionMessage: Variable where the VirtualBox (error) messages will be written to.
- vmName: Name of the VM.

8.2.7 AUTOTEST_VM_start

Description: Starts a virtual machine in an existing X session.
Parameter:

- VMStartMessage: Variable where the VirtualBox (error) messages will be written to.
- vmName: Name of the VM.

8.2.8 AUTOTEST_VM_insertBootISO

Description: Inserts a bootable ISO into a VM.
Parameter:

- vmName: Name of the VM.
- iso: ISO file with full path.
- VMinsertBootISOMessage: Variable where the VirtualBox (error) messages will be written to.

8.2.9 AUTOTEST_VM_rebootFromHD

Description: Stops the VM, disables booting from ISO and enables HDD booting and starts the VM again.
Parameter:

- vmName: Name of the VM.

8.2.10 AUTOTEST_executePHPFunction

Description: Executes a PHP function with (optionall) parameters.
Parameter:

- vmName: Name of the VM.
- params: [0] function name, [1...] parameters for the PHP function.

8.2.11 AUTOTEST_VM_keyboardWrite

Description: Emulates the keystrokes into a VM.
Parameter:

- vmName: Name of the VM.
- toType: Input string with normal and special keys.

8.2.12 AUTOTEST_VM_ocrScreen

Description: Uses gocr to convert the contents of the VirtualBox VM display to text.
Parameter:

- vmName: Name of the VM.

Returns:

- The recognised text of the display.

8.2.13 AUTOTEST_VM_getStatus

Description: Parses the complete status of a VM.
Parameter:

- vmName: Name of the VM.

Returns:

- Array with the current state of the VM.

8.2.14 AUTOTEST_VM_isRunning

Description: Checks if a VM is switched on.
Parameter:

- vmName: Name of the VM.

Returns:

- true, when the VM is powered on, otherwise false.

8.3 ./inc/backup.php

Author: Hauke Goos-Habermann (HHabermann@pc-kiel.de)
Description: functions for controling BackupPC

8.3.1 BACKUP_showClientSettings

Description: Shows the dialog for starting and configuring BackupPC for a special client
Parameter:

- client: name of the client

8.3.2 BACKUP_getBackupDirs

Description: Returns the comma seperated list of directories that should be backupped on the client
Parameter:

- client: name of the client

8.3.3 BACKUP_saveBackupDirs

Description: Saves the list of backup diretories oo the client in the BackupPC file
Parameter:

- client: name of the client
- dirs: comma seperated list of all directories to backup on the client

8.3.4 BACKUP_getAdmins

Description: Stores informations about known administrators in the BackupPC configuration file in variables.
Parameter:

- adminLine: The current line in config.pl that stores the dsmin informations.
- admins: Array with all admins.

8.3.5 BACKUP_addAdmin

Description: Adds an admin to the config.pl configuration file of BackupPC.
Parameter:

- admin: Name of the admin.

8.3.6 BACKUP_delAdmin

Description: Deletes an admin from the config.pl configuration file of BackupPC.
Parameter:

- admin: Name of the admin.

8.4 ./inc/burn.php

Author: Hauke Goos-Habermann (HHabermann@pc-kiel.de)
Description: functions for burning CDs

8.4.1 BURN_listBurners

Description: returnes a selection of the available burners
Parameter:

- first: entry that should be shown first

8.4.2 BURN_getDevice

Description: returnes the device name for cdrecord from a specific burner
Parameter:

- burner: device name and burner name

8.4.3 BURN_blank

Description: blanks a CD-RW
Parameter:

- dev: device name of the burner
- method: blanking method (fast, all);

8.4.4 BURN_burn

Description: burns an ISO
Parameter:

- dev: device name of the burner
- iso: name of the ISO file
- speed: the write speed

8.4.5 BURN_getStatus

Description: returns the status of the burner (BURNSTATE_IDLE, BURNSTATE_BLANK, BURNSTATE_BURN)

8.4.6 BURN_showLog

Description: shows a status info window about the current burner state

8.4.7 BURN_checkISO

Description: checks, if the client ISO exist and create i otherwise
Parameter:

- arch: Architecture of the ISO (32 bits = i386, 64 bits = amd64).

8.4.8 BURN_getISOSize

Description: Gets the size of an ISO.
Parameter:

- arch: Architecture of the ISO (32 bits = i386, 64 bits = amd64).

Returns:

- : Size of the ISO or error message, if the ISO could not be found.

8.5 ./inc/capture.php

Author: Hauke Goos-Habermann (HHabermann@pc-kiel.de)
Description: routines storing and loading POST and GET values in forms

8.5.1 CAPTURE_getKeys

Description: gets all POST or GET variables and returnes all keys and values as an assiciative array. Values of buttons are filtered out.
Parameter:

- var: set to $_POST or $_GET
- allowBut: set to true, if button should be captured too

8.5.2 CAPTURE_captureAll

Description: stores all POST and GET variables to the DB
Parameter:

- step: number of the step, this is used, if there are "subpages" of a page e.g. clientcdistr.php
- comment: comment to add to the entry
- allowBut: set to true, if button should be captured too

8.5.3 CAPTURE_load

Description: loads all POST and GET variables for a special page from the DB to emulate the user input while makeing a screenshot
Parameter:

- GET[page]: has to be set to the name of the page
- GET[captureLoad]: has to be set to "1" to activate loading of the saved values

8.5.4 CAPTURE_deActivate

Description: (de)activates capturing the POST, GET values
Parameter:

- activate: true, if you want to activate capturing. otherwise false

8.5.5 CAPTURE_isActive

Description: returnes true, if capturing of POST, GET values is activated. otherwise false

8.5.6 CAPTURE_captureImg

Description: returnes the status image URL of the current capture state

8.5.7 CAPTURE_toggle

Description: toggles the current capture state

8.5.8 CAPTURE_showMessageBox

Description: shows a message box, if capturing is enabled

8.5.9 CAPTURE_showEntries

Description: shows a table of the captured pages with the possibility to delete entries.

8.5.10 CAPTURE_deleteById

Description: deletes a capture entry.
Parameter:

- id: the id of the capture entry to delete

8.5.11 CAPTURE_showMarker

Description: Shows a new column with a marker that is used for autodetecting the screenshot size by khtml2png.

8.5.12 CAPTURE_showTableWith

Description: Adds a width element if in captureLoad mode.

8.6 ./inc/CAutoTest.php

Author: Hauke Goos-Habermann (HHabermann@pc-kiel.de)
Description: Class for test automation.

8.6.1 CClient::__construct

Description: Constructor for new CClient objects. The object holds all information about a single client and loads the values from the DB.
Parameter:

- in: ID of an existing client (to load), name of an existing or nonexisting (to create) client or associative array of parameters.
- checkMode: Check for the input variable.

8.6.2 CAutoTest::readAndDefineOrDieXML

Description: Checks, is a given variable has a value that is not NULL (it was read via XML) or let the programm die with an error.
Parameter:

- constant: Name of the constant to define.
- val: Value read from the XML to check.

8.6.3 CAutoTest::readSettings

Description: Reads basic settings from settings.m23test and sets them as constants.

8.6.4 CAutoTest::VMCreate

Description: Creates a new VM with virtual hard drive in VirtualBox and (optionally) inserts a bootable ISO into a VM.

8.6.5 CAutoTest::VMStart

Description: Starts a virtual machine in an existing X session.

8.6.6 CAutoTest::setISO

Description: Sets the filename and path of the ISO image for booting.
Parameter:

- isoFile: Filename and path of the ISO image for booting.

8.6.7 CAutoTest::getISO

Description: Get the filename and path to the ISO image for booting.
Returns:

- Filename and path to the ISO image for booting.

8.6.8 CAutoTest::isVM

Description: Check, if the the test should be run in a VM.
Returns:

- true, when the test should be run in a VM, otherwise false.

8.6.9 CAutoTest::getVM_hdsize

Description: Get the hard disk size of the virtual machine (to create).
Returns:

- Hard disk size of the VM.

8.6.10 CAutoTest::getVMRam

Description: Get the ram size of the virtual machine (to create).
Returns:

- Ram size of the VM.

8.6.11 CAutoTest::getMachine

Description: Get the name of the (real or virtual) machine, the test is run on.
Returns:

- Name of the (real or virtual) machine, the test is run on.

8.6.12 CAutoTest::getMovieFileName

Description: Get the movie file name for capturing the VM's screen.
Returns:

- Movie file name (with full path) for capturing the VM's screen.

8.6.13 CAutoTest::getLogFileName

Description: Get the file name for the log file.
Returns:

- Log file name (with full path).

8.6.14 CAutoTest::addToLogFile

Description: Appends lines to the log file.
Parameter:

- lines: The lines to add.

8.6.15 CAutoTest::getTimestampString

Description: Returns the date and time in human readable form.
Returns:

- Date and time in human readable form.

8.6.16 CAutoTest::setTriggered

Description: Sets the trigger state ofthe current sequence event.
Parameter:

- triggered: true, when the current sequence event was triggered, otherwise false.

Returns:

- true, when the current sequence event was triggered.

8.6.17 CAutoTest::isTriggered

Description: Checks, if the current sequence event was triggered.
Returns:

- true, when the current sequence event was triggered.

8.6.18 CAutoTest::setTimeout

Description: Sets the timeout for the current sequence element.

8.6.19 CAutoTest::decTimeout

Description: Decrements the remaining time for the timeout.

8.6.20 CAutoTest::addToSequence

Description: Adds an element to the sequence.
Parameter:

- triggerType: Type of the trigger (CAutoTest::TRIGGER_*) or the type event, that should happen to begin with the given element of the sequence.
- triggerParam: Parameter for the trigger (e.g. string that should be read from the screen when in CAutoTest::TRIGGER_OCR mode).
- answersA: Associative array with the answers and parameters.
- execType: Type of action (CAutoTest::EXEC_*), that will be executed when the trigger is hit.
- execParam: Parameter for the action (e.g. keys to press, when in CAutoTest::EXEC_KEY mode).
- timeout: Time to wait (in seconds) until the element of sequence will become a failure.
- description: Description for the test.

8.6.21 CAutoTest::matchArray

Description: Checks, if the search text is found in one of the texts contained in the array.
Parameter:

- search: Text to search in the array elements.
- array: Array with texts as element values.

8.6.22 CAutoTest::getAnswersA

Description: Returns the array with the answers of the current sequence element.
Returns:

- Array with the answers of the current sequence element.

8.6.23 CAutoTest::getTriggerTypes

Description: Returns the trigger type of the current sequence element.
Returns:

- Trigger type of the current sequence element.

8.6.24 CAutoTest::getTriggerParams

Description: Returns the trigger parameter of the current sequence element.
Returns:

- Trigger parameter of the current sequence element.

8.6.25 CAutoTest::getExecTypes

Description: Returns the type of execution of the current sequence element.
Returns:

- Type of execution of the current sequence element.

8.6.26 CAutoTest::getExecParams

Description: Returns the parameter for execution of the current sequence element.
Returns:

- Parameter for execution of the current sequence element.

8.6.27 CAutoTest::checkTriggerResult

Description: Checks, if the result (e.g. from AUTOTEST_VM_ocrScreen) is found in the good, warn or bad array and executes the matching element finish handler.

8.6.28 CAutoTest::executeTriggerAction

Description: Executes the action of the current sequence element.

8.6.29 CAutoTest::waitForTrigger

Description: Waits for a trigger event, to execute the action.

8.6.30 CAutoTest::getCurElement

Description: Returns the current sequence element.
Returns:

- Trigger type of the current sequence element.

8.6.31 CAutoTest::nextCurElement

Description: Increments the current sequence element number.

8.6.32 CAutoTest::showAndLogMessage

Description: Shows a message and logs it to the log file.
Parameter:

- msg: Message to show in the console and the log file.
- prefix: A prefix show before the message to indicate the type of the message.

8.6.33 CAutoTest::elemOk

Description: The current sequence elements was finished sucessfully.
Parameter:

- msg: Message to show in the console and the log file.

8.6.34 CAutoTest::elemWarn

Description: The current sequence elements was finished with a warning.
Parameter:

- msg: Message to show in the console and the log file.

8.6.35 CAutoTest::elemBad

Description: There was an error in the current sequence element, so the execution must bestopped.
Parameter:

- msg: Message to show in the console and the log file.

8.6.36 CAutoTest::setVariableFromXML

Description: Returns the input value when it is not NULL or exists the script with an error message.
Parameter:

- val: Input value.
- descr: Description for the value.

Returns:

- Input value when it is not NULL or exists the script with an error message.

8.6.37 CAutoTest::triggerTypeToConstant

Description: Tries to convert the trigger type (string) to a trigger type constant.
Parameter:

- type: Trigger type (string).

Returns:

- Trigger type constant.

8.6.38 CAutoTest::parseTriggerFromXML

Description: Parses the trigger and its type from the XML.
Parameter:

- xmlO: Prased XML structure pointing to the trigger.
- testDescription: Description of the test.
- testTrigger: Trigger parameter. (Result is written to this pointer)
- testTriggerType: Trigger type. (Result is written to this pointer)

8.6.39 CAutoTest::parseActionFromXML

Description: Parses the action and its type from the XML.
Parameter:

- xmlO: Prased XML structure pointing to the action.
- testDescription: Description of the test.
- testAction: Action parameter. (Result is written to this pointer)
- testActionType: Action type. (Result is written to this pointer)

8.6.40 CAutoTest::parseAnswersFromXML

Description: Parses an (good, warn, bad) array from the XML.
Parameter:

- xmlO: Prased XML structure pointing to the array.
- goodWarnBad: Answer type (CAutoTest::GWB_GOOD, GWB_WARN or GWB_BAD).

Returns:

- Associative array with the good, warn or bad answers, how to fetch the answer from the client/webbrowser/etc. and the answer type (GWB_GOOD, GWB_WARN or GWB_BAD).

8.6.41 CAutoTest::parseXML

Description: Parses the XML test description file.
Parameter:

- xmlFile: File name (with full path) of the XML test description file.
- argv: Array with the command line parameters.

8.7 ./inc/CFDiskAlles.php

Author: Hauke Goos-Habermann (HHabermann@pc-kiel.de)
Description: functions for partition and format of the client. print partition information and do the format itself.

8.7.1 FDISK_mdToEndOfArray

Description: Orders all MD devices from the input array to the end of the output array.
Parameter:

- in: Associative array with devices as keys and values (e.g Array ([/dev/md0] => /dev/md0 [/dev/sda1] => /dev/sda1 [/dev/sdb2] => /dev/sdb2)).

Returns:

- Associative array with devices as keys and values where the MDs are at the end (e.g. Array ([/dev/sda1] => /dev/sda1 [/dev/sdb2] => /dev/sdb2 [/dev/md0] => /dev/md0)).

8.7.2 FDISK_fdiskSessionReset

Description: Sets back all session variables (client name optionally) for partitioning and formating a client.
Parameter:

- resetClientName: If set to true, the name of the client will be deleted too (and re-set by FDISK_fdiskSessionClient).

8.7.3 FDISK_fdiskSessionPartMethod

Description: Stores the partitioning method in the session.
Parameter:

- newMethod: The new method to set or false for not changing.

Returns:

- The current partitioning method.

8.7.4 FDISK_fdiskSessionTitle

Description: Stores the partitioning title in the session.
Parameter:

- newTitle: The new title to set or false for not changing.

Returns:

- The current partitioning title.

8.7.5 FDISK_getPartitionByType

Description: Gets the FIRST partition matching a partition type.
Parameter:

- param: parameter string containing status informations about the harddisks
- vDev: Virtual (internally used) device number.
- type: type of the partition (primary, extended, logical)

Returns:

- Virtual partition number of the FIRST partition matching a partition type or false, if no partition was found.

8.7.6 FDISK_listDrivesAndPartitions

Description: Generates a selection that contains all drives and partitions of a given client.
Parameter:

- param: parameter string containing status informations about the harddisks
- default: the drive to show first
- selName: the name the selection is called in PHP and HTML
- pathFilter: Set this to another value than false if you want only devices with a given string in it.

Returns:

- String with the HTML selection.

8.7.7 FDISK_selectDrives

Description: creates a selection list of all drives
Parameter:

- param: parameter string containing status informations about the harddisks
- selName: the name the selection is called in PHP and HTML
- first: the drive to show first

8.7.8 FDISK_getFirstDrive

Description: return the first drive as installation drive
Parameter:

- param: parameter string containing status informations about the harddisks

8.7.9 FDISK_delFstab

Description: Removes an entry from the fstab array.
Parameter:

- fstab: Array that contains the fstab information. The changed fstab will be written to this parameter too.
- fstabNr: Number of the fstab entry to delete.

8.7.10 FDISK_fstabAddDialog2

Description: Dialog for adding fstab entries. This version uses the param and fstab parameters from the session.

8.7.11 FDISK_adjustFstabParam

Description: Adjust the parameter block of a fstab line to make it use an supported FS.
Parameter:

- param: The parameter block of a fstab line
- sourceName: The name of the package source list

Returns:

- Adjust the parameter block of a fstab line

8.7.12 FDISK_genManualFstab

Description: Generates commands to edit a given fstab, add new entries and remove old ones before.
Parameter:

- param: parameter string containing status informations about the harddisks
- mntPrefix: Prefix to set before the mountpoint (e.g. /mnt/m23root/)

8.7.13 FDISK_getAllDrives

Description: gets all drives of the client
Parameter:

- partitions: associative array containing status information about the harddisks

8.7.14 FDISK_listSupportedFS

Description: lists all supported fileSystems for the menu.
Parameter:

- selName: name the selection list, used for the html form
- showFirst: name of file system shown first

Returns:

- The currently choosen file system.

8.7.15 FDISK_listInstPartSelector

Description: lists all partitions to select for swap and install partition
Parameter:

- param: parameter string containing status informations about the harddisks
- default: partition that should be shown as selected
- type: array with filesystems that are possible for installation or swap
- selName: name of the selection

8.7.16 FDISK_defineDrive

Description: defines drive information for the clientBuilder
Parameter:

- client: client name
- path: path to the drive (/dev/hda, /dev/hdb, ...)
- size: size of the drive in MB
- upperI: upper tolerance border for disks with identical type
- lowerI: lower tolerance border for disks with identical type
- upperO: upper tolerance border for disks with other type
- lowerO: lower tolerance border for disks with other type
- asSpeciefied: use the speciefied disk, if it exists (is set to "yes" or empty)
- sizeAdjustmentType: defines how the partitions should be adjusted, if there is more or less space on the client that the defined one. "system" increases or tries to keep the size of the system partition. "percentage" makes a percentage adjustment of all partitions.

8.7.17 FDISK_getDiskType

Description: returnes the type of the drive (CFDiskBasic::DISK_TYPE_IDE, CFDiskBasic::DISK_TYPE_SCSI)
Parameter:

- path: the path to the device (e.g. /dev/hde)

8.7.18 FDISK_getDrivePartitionSize

Description: Calculates the size of a drive or partition.
Parameter:

- vDev: Virtual (internally used) device number.
- vPart: Virtual (internally used) partition number. This is normally another number than the physical number (e.g. 0 on /dev/hda1)
- param: parameter string containing status information about the harddisks

Returns:

- Size of the drive or partition in MB.

8.7.19 FDISK_deleteDriveFromParam

Description: Deletes all drive and partition parameters of a drive from param without correcting any order.
Parameter:

- vDev: Virtual (internally used) device number of the drive to delete.
- param: parameter string containing status informations about the harddisks.

Returns:

- Changed param without the drive.

8.7.20 CFDiskIO::setClientName

Description: Sets the client name, if given via constructor.
Parameter:

- clientName: String to check, if it is a valid client name.

Returns:

- : true, if it is a valid client name, otherwise false.

8.7.21 CFDiskIO::getClientName

Description: Gets the client name, if set via constructor.
Returns:

- : Client name or dies, when no client name was set.

8.7.22 CFDiskIO::getClientObject

Description: Gets the client object, of the client the partitioning belongs to.
Returns:

- : Client object.

8.8 ./inc/checks.php

Author: Daniel Kasten (DKasten@pc-kiel.de), Hauke Goos-Habermann (hauke@goos-habermann.de)
Description: different checks for validation of email, ip, netmasks, etc.

8.8.1 CHECK_safeFilename

Description: Make sure, the file/directory name is safe and doesn't contain evil characters.
Parameter:

- fileName: File/directory name to make safe.

Returns:

- The safe made file/directory name.

8.8.2 CHECK_text2db

Description: Makes a text safe for using it in the database.
Parameter:

- val: Text to use;
- like: Set to true if the text should be used as the LIKE parameter.

Returns:

- The safe made text.

8.8.3 CHECK_db2text

Description: Converts a string from the DB format to a normal string.
Parameter:

- val: String to use.

Returns:

- The safe made text.

8.8.4 CHECK_FW

Description: Variable checking firewall, that checks a bunch of variables if they contain only valid characters.
Parameter:

- List of parameters, where the first of two is the checking parameter and the second the value to check.

8.8.5 CHECK_deviceName

Description: Checks if the input value is a valid device name for a HD drive or partition or a RAID.
Parameter:

- devName: Device name to check.
- partition: Set to true if you want to check for a partition.
- diskOrPartition: Set to true, if a disk or partition should be valid. This overwrites the parameter "partition".
- raidAllowed: Set to true, if RAIDs are allowed too.

Returns:

- The input value is a valid drive, partition or RAID or false on an error.

8.8.6 CHECK_mointPoint

Description: Checks if the input value is a valid mountpoint.
Parameter:

- mountpoint: Mountpoint to check.

Returns:

- The input value is a valid mountpoint or false on an error.

8.8.7 CHECK_int

Description: Checks if the input value is an integer and shuts down the application if not.
Parameter:

- val: Input value to check.
- allowEmpty: Set to true if you want to allow empty strings.
- returnNoDie: Set to true if you want to return (instead of aborting the program) when an error in the input is found.

Returns:

- The input value is an integer or false on an error.

8.8.8 CHECK_float

Description: Checks if the input value is a float number and shuts down the application if not.
Parameter:

- val: Input value to check.
- allowEmpty: Set to true if you want to allow empty strings.
- returnNoDie: Set to true if you want to return (instead of aborting the program) when an error in the input is found.

Returns:

- The input value if it's a float number or false on an error.

8.8.9 CHECK_strAlpha

Description: Checks if the input value is a string that contains only characters and shuts down the application if not.
Parameter:

- val: Input value to check.
- maxlen: The maximal length of the the string or 0 if the string length doesn't matter.
- allowEmpty: Set to true if you want to allow empty strings.
- returnNoDie: Set to true if you want to return (instead of aborting the program) when an error in the input is found.

Returns:

- The input value if it contains only characters or stops (or false, if $returnNoDie is true) the program on an error.

8.8.10 CHECK_strAlphaNum

Description: Checks if the input value is a string that contains only characters and digits and shuts down the application if not.
Parameter:

- val: Input value to check.
- maxlen: The maximal length of the the string or 0 if the string length doesn't matter.
- allowEmpty: Set to true if you want to allow empty strings.
- returnNoDie: Set to true if you want to return (instead of aborting the program) when an error in the input is found.

Returns:

- The input value if it contains only characters and digits or stops (or false, if $returnNoDie is true) the program on an error.

8.8.11 CHECK_letFWDie

Description: Lets the variable checking firewall die with error message and info why and where it stopped executing the script.
Parameter:

- dieMessage: Message to show if the script should be stopped.

8.8.12 CHECK_str

Description: Checks if the input string only contains valid characters and is not longer than the maximum length and shuts down the application if not.
Parameter:

- val: String value to check.
- maxlen: The maximal length of the the string or 0 if the string length doesn't matter.
- allowEmpty: Set to true if you want to allow empty strings.
- returnNoDie: Set to true if you want to return (instead of aborting the program) when an error in the input is found.

Returns:

- The input string or stops the program on an error.

8.8.13 countLinesInFile

Description: counts the lines of a file, return value is the amount of lines
Parameter:

- dateiname: file name
- ignoreEmpty: If set to true, empty lines are ignored.

8.8.14 checkIP

Description: checks if an ip is valid
Parameter:

- string: ip value to check

Returns:

- true if IP is valid, else false

8.8.15 checkMAC

Description: Checks if a MAC address is valid.
Parameter:

- mac: MAC address to test.

Returns:

-

8.8.16 checkNetmask

Description: checks if a netmask is valid
Parameter:

- string: netmask value to check

Returns:

- true if netmask is valid, else false

8.8.17 checkEmail

Description: checks if a email address is valid, returns 1 if it is a valid netmask otherwise 0
Parameter:

- string: email address value to check

8.8.18 checkFQDN

Description: Checks if a string contains only characters that are allowed in a FQDN.
Parameter:

- string: string to check for special characters

8.8.19 checkNormalKeys

Description: checks if a string doesn't contain any special letters, returns 1 if it doesn't contain special characters otherwise 0
Parameter:

- string: string to check for special characters

8.9 ./inc/client_details.php

Author: Hauke Goos-Habermann (HHabermann@pc-kiel.de)
Description: Functions for drawing the buttons etc. in client_details.php.

8.9.1 CLIENT_DETAILS_beginCategory

Description: Starts a new named section for icons.
Parameter:

- title: Title of the section.
- anchor: A HTML anchor where the a special URL can jump to.

8.9.2 CLIENT_DETAILS_endCategory

Description: Ends the previously opened icon section.

8.9.3 CLIENT_DETAILS_addIcon

Description: Adds an icon in a section.
Parameter:

- page: The m23 page to link to.
- urlParams: Additional parameters for the URL (e.g. "&action=deinstall").
- icon: File name of the icon placed under /m23/data+scripts/gfx.
- title: Title for the icon shown under it.
- tooltip: The tooltip text that is shown when the mouse is over the icon.

8.9.4 CLIENT_DETAILS_addIcon2

Description: Adds an icon in a section that can link to all URLs.
Parameter:

- url: The URL to link to.
- icon: File name of the icon placed under /m23/data+scripts/gfx.
- title: Title for the icon shown under it.
- tooltip: The tooltip text that is shown when the mouse is over the icon.

8.10 ./inc/client.php

Author: Daniel Kasten (DKasten@pc-kiel.de), Hauke Goos-Habermann (HHabermann@pc-kiel.de)
Description: a lot of routines for client handling. routines for: install and deinstall packages on clients, get information about a special client, list all clients,...

8.10.1 CLIENT_removeServerCache

Description: Removes the client cache on the m23 server.
Parameter:

- client: Name of the client.

8.10.2 CLIENT_touchLiveLogFile

Description: Touches the live log file of a given client and returns the full name of the log file (with directory).
Parameter:

- client: Name of the client.

Returns:

- The full name of the log file (with directory).

8.10.3 CLIENT_getNextFreeIp

Description: Get the next free IP address that can be used as m23 client.
Returns:

- Next free IP address.

8.10.4 CLIENT_getAllAsRes

Description: Creates and executes an SQL statement for getting all values of all clients.
Parameter:

- order: Name of the field to order the results by.

Returns:

- MySQL resource ID.

8.10.5 CLIENT_touchLogFile

Description: Touches a log file in the client's directory and returns the full name of the log file (with directory).
Parameter:

- client: Name of the client.
- base: The base name of the log file.

Returns:

- The full name of the log file (with directory).

8.10.6 CLIENT_liveLogJobName

Description: Generates the job name of the sever's live log job.
Parameter:

- client: Name of the client.

Returns:

- The job name of the sever's live log job.

8.10.7 CLIENT_stopLiveScreenRecording

Description: Stops the screen installation session for real-time client logging.
Parameter:

- client: Name of the client.

8.10.8 CLIENT_startLiveScreenRecording

Description: Saves a screen installation session to a log file on the server in real-time. The server runs a screen for consecutively connecting the client.
Parameter:

- client: Name of the client.

8.10.9 CLIENT_getOverviewSearchLine

Description: Checks all client search dialogs and returns the current search term.
Parameter:

- amount: Amount of client search dialogs to check.

Returns:

- The current client search term.

8.10.10 CLIENT_showOverviewSearchDialog

Description: Shows a client search dialog for the client overview.
Parameter:

- htmlName: Base name for the HTML edit line and the buttons.
- addTable: If set to true, a table structure is build around the dialog.

8.10.11 CLIENT_extraWebAction

Description: Executes extra actions from the client details page.
Parameter:

- action: The action to execute.
- client: Name of the client.

Returns:

- True if the action was executed or false otherwise.

8.10.12 CLIENT_getOption

Description: Returns an option of a client.
Parameter:

- client: Name of the client.
- optionName: Name of the option to ask (e.g. 'distr' for the client's distribution)

Returns:

- Value of the option the client uses.

8.10.13 CLIENT_getDistribution

Description: Returns the distribution of a client.
Parameter:

- client: Name of the client.

Returns:

- Name of the distribution, the client uses.

8.10.14 CLIENT_runDebconf

Description: Generates BASH code to import debconf settings from the DB into the debconf of the client.
Parameter:

- clientName: Name of the client.

8.10.15 CLIENT_copyDebconfDB

Description: Copies all debconf values from one client to another.
Parameter:

- clientName: Name of the source client.
- destClient: Name of the destination client.

8.10.16 CLIENT_setDebconfDB

Description: Sets debconf values for a client and a package.
Parameter:

- clientName: Name of the client.
- package: Name of the package.
- variablesValues[varname][val]: Value for the variable "varname".
- variablesValues[varname][type]: Type of the variable "varname".

Returns:

- debconf for debconf-set-selections.

8.10.17 CLIENT_getDebconfDB

Description: Generates the debconf output as debconf-set-selections expects it from the DB value.
Parameter:

- clientName: Name of the client.

Returns:

- debconf for debconf-set-selections.

8.10.18 CLIENT_getDebconfDBValue

Description: Get the debconf value of a variable of a package.
Parameter:

- clientName: Name of the client.
- package: Name of the package.
- var: Name of the variable to ask the value for.

Returns:

- Value of the package variable.

8.10.19 CLIENT_getAllClientNames

Description: Gets the names of all clients.
Returns:

- Array with the names of all clients.

8.10.20 CLIENT_getClientAmount

Description: Gets the amount of all clients.
Returns:

- Amount of all clients.

8.10.21 CLIENT_getCurrentMemoryUsage

Description: Gets the amount of free and total memory on a client or localhost.
Parameter:

- clientNameOrIP: The name of the client or localhost or an IP.

Returns:

- Associative array with the free memory in $out['free'] and the total memory in $out['all'] in KB.

8.10.22 CLIENT_getCurrentFreeSpaceInDir

Description: Get the amount of free space in a given directory on a client or localhost.
Parameter:

- clientNameOrIP: The name of the client or localhost or an IP.
- dir: The directory to check for.

Returns:

- The amount of free space in the directory in 1K blocks.

8.10.23 CLIENT_getClientID

Description: Returnes the ID of the calling client.

8.10.24 CLIENT_getActiveNetDevices

Description: Checks for active network devices on a client or localhost.
Parameter:

- clientNameOrIP: The name of the client or localhost or an IP.

Returns:

- Associtative array with active network cards (e.g. Array ([0] => Array ([dev] => eth0 [type] => encap:Ethernet [mac] => 00:52:66:23:00:23) [1] => Array ([dev] => venet0 [type] => encap:UNSPEC [mac] => 00-00-00-00-00-00-00-00-00-00-00-00-00-00-00-00))

8.10.25 CLIENT_executeOnClientOrIP

Description: Runs a script with "screen" in the background or under a plain BASH under a given user. The script can be executed on the local machine "localhost" or a remote client that is accessible via SSH with a public key and without a password.
Parameter:

- clientNameOrIP: The name of the client or localhost or an IP.
- jobName: name of the job screen should show
- cmds: the commands of the script
- user: user the script should be run under
- runInScreen: Set to true if the execution should be done in "screen". False executes it under the normal BASH.

Returns:

- The output of the screen (only available on direct output if $runInScreen is false.

8.10.26 CLIENT_isBasesystemInstalledFromImage

Description: Detects if the base system should be installed from an image.
Parameter:

- options: Array with the client options.

Returns:

- true if it should be installed from an image, otherwise false

8.10.27 CLIENT_addClient

Description: adds a new client to the database and prepares the client for the installation
Parameter:

- data['client']: client name
- data['office']: office
- data['name']: name of the user
- data['familyname']: family name of the user
- data['email']: email
- data['mac']: client MAC
- data['ip']: IP of the client

- data['netmask']: netmask of the client
- data['gateway']: gateway of the client
- data['dns1']: DNS server 1
- data['dns2']: DNS server 2
- data['newgroup']: group of the client
- data['language']: client language
- data['firstpw']: password for the first user login
- data['rootpassword']: root password
- options['packageProxy']: the ip of the proxy the packages should be fetched from
- options['packagePort']: the proxy port
- options['netRootPwd']: password for root during network booting
- options['ldaptype']: type of the LDAP server
- options['ldapserver']: name of the LDAP server
- options['nfshomeserver']: NFS home server with full path
- options['login']: login name for the user
- options['userID']: user ID for the LDAP account
- options['groupID']: group ID for the LDAP account
- options['addNewLocalLogin'] = $_POST[addNewLocalLogin];
- options['timeZone']: POSIX timezone
- options['getSystemtimeByNTP']: "yes", if the system time should be set with NTP
- options['installPrinter']: "yes", if printer drivers should be installed and printers detected
- clientAddType: can be CLIENT_ADD_TYPE_add if the client should be added, CLIENT_ADD_TYPE_define if it should be defined for mass installation or CLIENT_ADD_TYPE_assimilate if the client should be imported into the m23 system.
- cryptRootPw: set to true, if the password should be encrypted or false, if it's already encrypted

8.10.28 CLIENT_IPexists

Description: checks if an IP with the selected IP exists and returns true if yes, otherwise false
Parameter:

- ip: IP to check

8.10.29 CLIENT_MACexists

Description: checks if a mac with the selected mac exists and returns true if yes, otherwise false
Parameter:

- mac: MAC to check

8.10.30 CLIENT_exists

Description: checks if a client with the selected name exists and returns true if yes, otherwise false
Parameter:

- clientName: name of the client

8.10.31 CLIENT_getAskingParams

Description: returns database parameters of the asking client. The client is authetified by its m23shared clients name, client ID or ip

8.10.32 CLIENT_getParams

Description: returns database parameters of a special client
Parameter:

- clientName: name of the client

8.10.33 CLIENT_getClientStatus

Description: returnes the current client status
Parameter:

- client: name of the client

8.10.34 CLIENT_getProperty

Description: fetches a property from the client information of the database
Parameter:

- client: name of the client
- var: name of the property

8.10.35 CLIENT_listPackages

Description: lists all packages on the client
Parameter:

- client: name of the client
- key: keyword for searching for packages
- withActions: you can select to draw te action selection button, if you set it to true, you can delete packages and discard package deletion jobs

8.10.36 CLIENT_getPossibleActions

Description: list the possible actions. e.g. if a package is installed it can be uninstalled
Parameter:

- status: actual status of the package
- actionNr: number of the action radio button
- package: name of the package

8.10.37 CLIENT_acceptChanges

Description: removes packages or discards changes
Parameter:

- client: name of zhe client
- amount: amount of packages in the web interface

8.10.38 CLIENT_showHardwareInfo

Description: prints a table with hardware informations
Parameter:

- client: name of zhe client

8.10.39 CLIENT_showGeneralInfo

Description: prints a table with general information
Parameter:

- id: id of the client
- generateEnterKeep: set to true, if you want these values to be generated,entered or kept

8.10.40 CLIENT_showWaitingJobs

Description: shows the waiting jobs for the client
Parameter:

- client: name of the client

8.10.41 CLIENT_setLastmodify

Description: sets the last modified time of a client
Parameter:

- id: id of the client
- client: name of the client

8.10.42 CLIENT_getSubnet

Description: gets the subnet of a given ip and netmask
Parameter:

- ip: ip address
- netmask: netmask

8.10.43 CLIENT_getBroadcast

Description: gets the broadcast of a given ip and netmask
Parameter:

- ip: ip address
- netmask: netmask

8.10.44 CLIENT_convertMac

Description: converts a mac address to a 00:11... or 0011 format
Parameter:

- mac: the mac address
- splitter: select a character to split the mac in couples of two characters, if you leeave it blank, splitting characters will be removed

8.10.45 CLIENT_getIPbyName

Description: returns the ip from a selected clientname
Parameter:

- clientName: name of the client

8.10.46 CLIENT_getNamebyIP

Description: returns the clientname from a selected ip
Parameter:

- ip: ip of the client

8.10.47 CLIENT_getMACbyName

Description: returns the mac from a selected clientname
Parameter:

- clientName: name of the client

8.10.48 CLIENT_sshFetchJob

Description: Connects to the client via SSH and lets the next job fetch and execute it in a screen (named "m23install").
Parameter:

- clientName: name of the client
- ip: Optional parameter for the client's IP (faster than getting the IP by the client name)

8.10.49 CLIENT_backToRed

Description: Sets a client back to red state, as it was just after adding it.
Parameter:

- clientName: name of the client

8.10.50 CLIENT_desasterRecovery

Description: recover a client: all client jobs are done again, status is set to 0
Parameter:

- clientName: name of the client
- addInstallRemovalJobs: If set to true, the names of all installed packages will be combined to a m23normal and all revomed to a m23normalRemove job.
- addShutdownOrRebootPackage: If set to true, a shutdown or reboot package will be added.

8.10.51 CLIENT_wol

Description: wakes a client over the network
Parameter:

- clientName: name of the client

8.10.52 CLIENT_recalculateStatusBar

Description: Recalculates the percent points for the pending jobs on a client.
Parameter:

- clientName: name of the client

8.10.53 CLIENT_resetStatusBar

Description: Resets the percent points to 0 for the pending jobs on a client.
Parameter:

- clientName: name of the client

8.10.54 CLIENT_startInstall

Description: starts the installation on a client
Parameter:

- clientName: name of the client

8.10.55 CLIENT_resetAndInstall

Description: Resets or wakes the client to boot from network and run jobs
Parameter:

- clientName: name of the client

8.10.56 CLIENT_getBootType

Description: gets the type of network boot (pxe, etherboot)
Parameter:

- clientName: name of the client

8.10.57 CLIENT_isrunning

Description: tests out wether a client is up (running) or not
Parameter:

- clientName: name of the client

8.10.58 CLIENT_reset

Description: resets a client
Parameter:

- clientName: name of the client

8.10.59 CLIENT_showLog

Description: prints the log information of the client
Parameter:

- clientName: name of the client

8.10.60 CLIENT_getClientName

Description: returnes the client name of the calling client or the client given by its ID ($_GET['m23clientID']).

8.10.61 CLIENT_getAllOptions

Description: gets all options from the options column of a client as associative array
Parameter:

- clientName: name of the client

8.10.62 CLIENT_setAllOptions

Description: sets all options in the options column of a client
Parameter:

- clientName: name of the client
- options: the options as associative array

8.10.63 CLIENT_getAllAskingOptions

Description: gets all options from the options column of the calling client as associative array

8.10.64 CLIENT_getSetOption

Description: checks if a variable is set and places its value under the variable name in the options array
Parameter:

- options: name of the options array

8.10.65 CLIENT_options2HiddenForm

Description: generates hidden fields with the values of the option array
Parameter:

- options: name of the options array

8.10.66 CLIENT_hiddenForm2options

Description: reads the option values of the hidden fields and adds them to the options array
Parameter:

- options: name of the options array

8.10.67 CLIENT_getStatusimage

Description: return the image name with the correct color
Parameter:

- status: the status that should be converted to an image

8.10.68 CLIENT_showStatusSelection

Description: shows a dialog that lets you select the current status of a client
Parameter:

- client: the name of the client

8.10.69 CLIENT_listCriticalClients

Description: lists clients with critical status'

8.10.70 CLIENT_isInDebugMode

Description: returnes "true", if a client is in debug mode
Parameter:

- clientName: name of the client

8.10.71 CLIENT_toggleDebugMode

Description: en/disables the debug mode of a client
Parameter:

- clientName: name of the client
- enable: set to "true" to activate debug mode or to "false" to disable

8.10.72 CLIENT_getStatusimage

Description: return the image name with the correct color
Parameter:

- status: the status that should be converted to an image

8.10.73 CLIENT_generateHTMLStatusBar

Description: generates HTML code containing the status of the client with links to the pages
Parameter:

- clientName: name of the client

8.10.74 CLIENT_showDebugSelection

Description: shows a dialog that lets you select the current debug state of a client
Parameter:

- client: the name of the client

8.10.75 CLIENT_isInRescueMode

Description: checks if a clients has waiting rescue packages
Parameter:

- clientName: the name of the client

8.10.76 CLIENT_showDirectConnectionHelp

Description: returnes the help file for directConnection and replaces place holders with the correct values
Parameter:

- clientName: the name of the client
- language: language for the help file

8.10.77 CLIENT_isInDebugMode

Description: returnes "true", if the asking client is in debug mode

8.10.78 CLIENT_getToDetailsURL

Description: Generates the link to the client's control center page
Parameter:

- clientName: the name of the client
- id: the id of the client
- section: section to jump on the page

Returns:

- Link to the client's control center page

8.10.79 CLIENT_HTMLBackToDetails

Description: generates HTML code for returning to the client controll center page
Parameter:

- clientName: the name of the client
- id: the id of the client
- section: section to jump on the page

8.10.80 CLIENT_getId

Description: returnes the id of a client
Parameter:

- clientName: the name of the client

8.10.81 CLIENT_query

Description: returnes the result of a query for getting all clients matching selected states and groupNames. Empty values are interpreted as 'all' for this kind of value.
Parameter:

- o1: operator 1 (can be '=', '<', '>') selects of the first state should be equal, smaler or bigger that the state in s1
- s1: first state to compare with the state of the client
- o2: operator 2
- s2: second state to compare
- groupName: if you want to filter for special group, set it to the group name
- o3: operator 3
- s3: third state to compare
- search: Search string to search all clients for and only list matching clients or all if $search is empty.

8.10.82 CLIENT_addChangeElement

Description: Generates a HTML dialog element for changing a client property.
Parameter:

- elem: Name of the element.
- serverOnlyElement: Set to true if the element could only be changed in the DB and not on the server (e.g. a misspelled MAC)

8.10.83 CLIENT_showDelDialog

Description: Shows the dialog for deleting a client.

8.10.84 CLIENT_showAddDialog

Description: shows the dialog for adding, defining or changing a client
Parameter:

- addType: defines the behaviour and appearance of the dialog

8.10.85 CLIENT_deleteClient

Description: deletes a client and shows an optional message
Parameter:

- client: name of the client to delete
- showMsg: set to true, is a success message should me shown
- deleteVM: Set to true to delete the VM too.

8.10.86 CLIENT_getNames

Description: returns an array with all clients
Parameter:

- groupName: if the group is set, only clients in the group are returned, otherwise all clients

8.10.87 CLIENT_getNamesWithPackages

Description: returns an array with all clients having packages installed
Parameter:

- showFakeClients: if set to true, fake clients used to store package lists are shown. false only shows real clients

8.10.88 CLIENT_changeClient

Description: changes values of the clients

8.10.89 CLIENT_setAllParams

Description: Sets all parameters in the columns of a client
Parameter:

- client: name of the client
- data: the options as assiciative array

8.10.90 CLIENT_plinkFetchJob

Description: Connects to a client over the Putty SSH client and executes a command
Parameter:

- clientName: name of the client
- password: Password for root on the client
- jobName: name of the screen job on the server
- ubuntuUser: name of the Ubuntu user or empty if a Debian system is meant.

8.11 ./inc/CScredit.php

Author: Hauke Goos-Habermann (HHabermann@pc-kiel.de)
Description: Functions for the embedded script editor.

8.11.1 CScredit::__construct

Description: Constructor for new CScredit objects.

8.11.2 CScredit::setCurrentScriptFilename

Description: Corrects the given filename to have it a valid prefix and suffix and sets it as current file name.
Parameter:

- filename: The filename to check and correct.

8.11.3 CScredit::getCurrentScriptFilename

Description: Gets the current script filename.
Returns:

- Current script filename.

8.11.4 CScredit::getCurrentScriptFilenameFullPath

Description: Gets the current script filename with full path.
Returns:

- Current script filename full path.

8.11.5 CScredit::getCurrentScriptFilenameWithoutInstallPhp

Description: Gets the current script filename without "Install.php" at its end.
Returns:

- Current script filename without "Install.php" at its end.

8.11.6 CScredit::getLocalScriptFilenames

Description: Gets the filenames of local scripts.
Returns:

- Array with the filenames of local scripts.

8.11.7 CScredit::updateOnlineScriptInfo

Description: Downloads the information about online available scripts.
Returns:

- true, if the information could be fetched, otherwise false.

8.11.8 CScredit::getNextOnlineScriptInfo

Description: Gets an information about all scripts that are available online. Every call of the function fetches the information about one script.
Parameter:

- ts: Timestamp, when the script was uploaded.
- scriptId: Id of the script (needed for download).
- scriptName: Name of the script.
- author: Name of the author.
- description: Description for the script.

Returns:

- true, if an entry could be read otherwise (e.g. if all entries were read) false.

8.11.9 CScredit::loadOnlineScript

Description: Loads an online script into the editor.
Parameter:

- scriptIdToLoad: Id of the script to load.

8.11.10 CScredit::getOnlineScriptDialog

Description: Generates a dialog with JavaScript to get information about online scripts with download option.
Returns:

- Dialog (HTML) with JavaScript to get information about online scripts with download option.

8.11.11 CScredit::isNotSaved

Description: Returns if there is no script in the editor (after submitting).
Returns:

- true, if there is no script in the editor (after submitting), otherwise false.

8.11.12 CScredit::getNewScriptTemplate

Description: Returns a template for a basic script.
Returns:

- Text of the script template.

8.11.13 CScredit::uploadScript

Description: Checks, if all needed information are given before uploading the script.
Parameter:

- author: Name of the script author (or pseudonyme)
- description: Description for the script.
- text: The script code itself.

8.11.14 CScredit::getCurrentScript

Description: Get the text of the editor window.
Returns:

- Current text of the editor window.

8.11.15 CScredit::setCurrentScript

Description: Set the text of the editor window.
Parameter:

- text: Current text of the editor window to set.

8.11.16 CScredit::deleteCurrentScript

Description: Deletes the current script, if one is loaded.
Returns:

- true, if the script could be deleted, otherwise false.

8.11.17 CScredit::getViewScriptOutputDialog

Description: Generates a dialog with JavaScript to choose a client and to open the script output viewer for the currentry saved script.
Returns:

- Dialog (HTML) with JavaScript to choose a client and to open the script output viewer for the currentry saved script.

8.11.18 CScredit::saveScript

Description: Saves the script in the editor to the file.

8.11.19 CScredit::show

Description: Shows a script editor with syntax highlighting.

8.12 ./inc/db.php

Author: Daniel Kasten (DKasten@pc-kiel.de) ,Hauke Goos-Habermann (HHabermann@pc-kiel.de)
Description: database functions, open, close the database, get ip of the calling client

8.12.1 ip2longSafe

Description: Special version of ip2long that is safe on 32 bit machines.
Parameter:

- in: Input v4 IP (e.g. 192.168.1.23) or number.

Returns:

- Unsigned long representation of the input IP or the input numer.

8.12.2 getArchList

Description: Returns an associative array with the supported CPU architectures as key and value.
Returns:

- Associative array with the supported CPU architectures as key and value.

8.12.3 isMySQL3used

Description: checks if MySQL 3 is installed and returnes true if v3 is found, otherwise false

8.12.4 encryptShadow

Description: encrypts a password for adding a user to the client
Parameter:

- userName: the username for the account
- password: the unecrypted password to encrypt

8.12.5 getClientLanguage

8.12.6 getInstDev

Description: fetch the device for installation
Parameter:

- id: package ID

8.12.7 DB_getConnection

Description: Gets the MySQLi connection.
Returns:

- MySQLi connection.

8.12.8 DB_setConnection

Description: Sets the MySQLi connection to use globally.
Parameter:

- conn: MySQLi connection to use globally.

8.12.9 DB_isConnectionValid

Description: Checks, if the MySQLi connection is valid.
Returns:

- true on valid MySQLi connection, otherwise false.

8.12.10 dbConnect

Description: connects to the m23 database

8.12.11 dbClose

Description: closes the connection to the m23 database

8.12.12 getServerIP

Description: returnes the IP of the m23 server

8.12.13 getServerNetmask

Description: returnes the netmask of the m23 server

8.12.14 getServerNetwork

Description: Get the network IP of the m23 server.
Returns:

- Network IP of the m23 server.

8.12.15 getDNSServers

Description: Returnes an array with the DNS servers of the m23 server.

8.12.16 getServerGateway

Description: Returnes the gateway of the m23 server

8.12.17 sendClientStatus

Description: generates a bash script to send a status to the server
Parameter:

- id: package ID
- status: done, waiting; finished jobs should be set to done, waiting should not be used from this place

8.12.18 sendClientStageStatus

Description: generates a bash script to send a stage status to the server
Parameter:

- status: 0: client waiting for hardware detection, 1 hardware detection done, 2 partitionated and formated, base system is installed

8.12.19 returnClientStageStatus

Description: generates a bash script to send a stage status to the server
Parameter:

- status: 0: client waiting for hardware detection, 1 hardware detection done, 2 partitionated and formated, base system is installed

8.12.20 sendClientLogStatus

Description: generates a bash script to send log status to the server
Parameter:

- status: how the line should be named, that is logged to the server
- ok: true: operation sucessful, false: failure
- critical: if it is set to "true" the execution of the script is stopped and a local rescue console is opened

8.12.21 deleteClientLogs

Description: deletes the installation logs
Parameter:

- clientName: name of the client to delete all logs

8.12.22 workPhpName

Description: Generates an unique name for storing the work.php file.
Returns:

- Unique name for storing the work.php file.

8.12.23 executeNextWork

Description: generates a bash script that fetches the next work.php from server

8.12.24 DB_query

Description: makes a query and returns the default error message if an error occurs
Parameter:

- sql: sql query

8.12.25 DB_queryNoDie

Description: Executes a SQL query and returns the resource id to access the result.
Returns:

- Ressource id of the query result and DOESN'T die on an error.

8.12.26 DB_genPassword

Description: generates a random password with a specified length
Parameter:

- length: length of password

8.12.27 getClientIP

Description: returnes the IP of the calling client

8.12.28 implodeAssoc

Description: makes a string from an associative array
Parameter:

- glue: the string to glue the parts of the array with
- arr: array to implode

8.12.29 explodeAssoc

Description: makes an associative array from a string
Parameter:

- glue: the string to glue the parts of the array with
- arr: array to explode

8.12.30 sedSearchReplace

Description: generates BASH code to search and replace a string in a file using sed keeping the ownership an permissions
Parameter:

- pathFile: file with whole path, in that should be searched and replaced
- search: search pattern
- replace: replace string

8.12.31 isProgrammInstalled

Description: returnes true if a programm can be used
Parameter:

- progName: name of the programm

8.12.32 pingIP

Description: tests, if someone is answering the ping on a given IP address. returnes true, if someone answers (needs "iputils-ping" to be installed)
Parameter:

- ip

8.12.33 delFromArray

Description: deletes all entries in the array $arr assigned by the keys stored in the array $delKeys. the new array without the entries in $delKeys is returned.
Parameter:

- arr: array with the entries to filter
- delKeys: array with all keys to delete from $arr

8.12.34 delValuesFromArray

Description: deletes all entries in the array $arr with values stored in the array $delVals.
Parameter:

- arr: array with the entries to filter
- delVals: array with all values to delete from $arr

Returns:

- array without the entries in $delVals.

8.12.35 DB_getLikeableColumns

Description: Returns an associative array that contains all fields of a table that can be searched by LIKE.
Parameter:

- table: Name of the table to search.

Returns:

- associative array that contains all fields of a table that can be searched by LIKE.

8.13 ./inc/dhcp.php

Author: Daniel Kasten (DKasten@pc-kiel.de), Hauke Goos-Habermann (HHabermann@pc-kiel.de)
Description: functions to add and remove a client to/from the dhcp server list.

8.13.1 DHCP_exportDHCPSettingsForExternalDHCPServer

Description: Exports the DHCP settings of m23 clients that are booting over the network and adds settings for external DHCP servers.

8.13.2 DHCP_lineNumberAffterLastClient

Description: Gets the line number with the last client definition in the dhcpd.conf.
Returns:

- Line number with the last client definition in the dhcpd.conf.

8.13.3 DHCP_addSubnetDefinition

Description: Adds the subnet definition to the dhcpd.conf to let the DHCP server give out network information to clients to other subnets.
Parameter:

- subnet: The subnet.
- netmask: netmask for the ip

8.13.4 DHCP_delSubnetDefinition

Description: Removes a subnet definition from the dhcpd.conf.
Parameter:

- subnet: The subnet.
- netmask: netmask for the ip

8.13.5 DHCP_addDynamicRange

Description: Adds a dynamic IP range to the dhcpd.conf and restarts the DHCP server.
Parameter:

- firstIP: The first IP marking the begin of the dynamic IP range.
- lastIP: The last IP marking the end of the dynamic IP range.
- netmask: Netmask for the IPs.
- gateway: The gateway IP.

Returns:

- true, if the DHCP server could be restarted with the new settings.

8.13.6 DHCP_getDynamicRanges

Description: Gets all dynamic IP ranges from the dhcpd.conf.
Returns:

- Associative array with the found IP ranges (e.g. Array ([0] => Array ([netmask] => 255.255.255.0 [firstIP] => 192.168.1.10 [lastIP] => 192.168.1.200 [gateway] => 192.168.1.1) [1] => ...)) or empty array.

8.13.7 DHCP_getNetmaskOfDynamicRanges

Description: Gets the netmask of a dynamic range identified by the first IP of the range.
Parameter:

- firstIP: The first IP marking the begin of the dynamic IP range.

Returns:

- Netmask or false in case of an error.

8.13.8 DHCP_delDynamicRange

Description: Removes a dynamic IP range from the dhcpd.conf and restarts the DHCP server.
Parameter:

- firstIP: The first IP marking the begin of the dynamic IP range.
- lastIP: The last IP marking the end of the dynamic IP range.

Returns:

- true, if the DHCP server could be restarted with the new settings.

8.13.9 DHCP_bootTypeToNewFormat

Description: Converts a boolean boot type to the new string format.
Parameter:

- bootType: Boolean or string format (e.g. "pxe") boot type.

Returns:

- String format (e.g. "pxe") boot type.

8.13.10 DHCP_runScript

Description: Runs the script for controlling an external DHCP server.
Parameter:

- command: 'add' for adding an entry to the DHCP server or 'remove' for removing.
- clientName: name of the client
- ip: ip address of the client
- netmask: netmask for the ip
- mac: mac addresse of the network card
- bootType: Parameter can a string: pxe, etherboot, gpxe, none
- gateway: The gateway for the client.

8.13.11 DHCP_addClient

Description: adds a new client to the dhcpd.conf and restarts the dhcpd-server
Parameter:

- clientName: name of the client
- ip: ip address of the client
- netmask: netmask for the ip
- mac: mac addresse of the network card
- bootType: Parameter can be boolean for backward compatibility: if true use PXE for the client, otherwise use Etherboot
- bootType: Parameter can a string: pxe, etherboot, gpxe, none
- gateway: The gateway for the client.
- updateDB: If set to true, the boot type is set for the client in the DB.

Returns:

- true, if the DHCP server could be restarted with the new settings.

8.13.12 DHCP_addLineToDHCPDConf

Description: Adds a line to the dhcpd.conf file.
Parameter:

- line: Line to add.

8.13.13 DHCP_restartDHCPserver

Description: Restarts the DHCP server.
Returns:

- true if it clould be (re)started otherwise false.

8.13.14 DHCP_rmClient

Description: removes a client from dhcpd.conf and restarts the dhcpd-server
Parameter:

- clientName: name of the client

8.13.15 DHCP_setBootimage

Description: sets the bootimage of a client for EtherBoot
Parameter:

- clientName: name of the client
- bootImage: name of the bootimage (hdboot, ip address for name)

8.13.16 DHCP_activateBoot

Description: switches the network boot on or off
Parameter:

- clientName: name of the client
- on: true activates the network boot, false deactivates
- bootType: The boot type CAN be given here (e.g. pxe or etherboot)

8.13.17 DHCP_calcPXEIP

Description: calculates the ip for the pxe config file
Parameter:

- ip: ip address to convert to the PXE file name

8.13.18 DHCP_writePXEcfg

Description: writes the pxe config file for te client
Parameter:

- clientName: name of the client
- arch: computer architecture (i386 or amd64)

8.13.19 DHCP_removePXEcfg

Description: removes the PXE start file for a special client
Parameter:

- clientName: name of the client

8.13.20 DHCP_isNetworkBootingActive

Description: Checks, if a client has network booting enabled.
Parameter:

- clientName: name of the client

Returns:

- : True, if network booting is active, false otherwise.

8.14 ./inc/distr/halfSister/packages.php

Author: Hauke Goos-Habermann (HHabermann@pc-kiel.de)
Description: Basic package operations (search, add,...) for halfSister distributions

8.14.1 PKG_updatePackageSearchCacheFile

Description: Updates the compressed package search file.
Parameter:

- packagesource: Name of the package sources list.

Returns:

- The name of the package cache.

8.14.2 PKG_fastGetInstalledPackages

Description: Gets a list of all installed packages (faster than dpkg –get-selections).
Parameter:

- storeFile: File name to store the list of installed on the client or empty if the list should be outputted to stdout.

8.14.3 pkgUpdateCacheOnServer

Parameter:

- $1 (packagesourcename): Name der Paketquellenliste
- $2 (packagesource): Inhalt der Paketquellenliste
- $3 (force): set to true (1) if you want to update the package info and the time is not over
- $4 (arch): Architecture (amd64/i386) to get package infos for.

8.14.4 PKG_listPackages

Description: list packages matching the key
Parameter:

- key: search key
- distr: the distribution name
- packagesource: name of the package source
- client: Name of the client the packages are searched for. (Not used here and only for halfSister)

8.14.5 PKG_previewInstall

Description: shows what happens if packages get (de)installed
Parameter:

- clientName: name of the client
- distr: the distribution name
- packagesource: name of the package source
- packages: the packages to be installed
- aptCommand: sets the apt-get command: install, remove

8.14.6 PKG_getKernels

Description: Generates an associative array with the available kernels for an architecture and distribution as keys and values.
Parameter:

- distr: the distribution name
- packagesource: name of the package source
- arch: Architecture to get package infos for.

Returns:

- Asssociative array with the available kernels for an architecture and distribution as keys and values.

8.14.7 PKG_translateClientPackageStatus

Description: translates the package status to human language ;)
Parameter:

- status: status code you want to translate

8.15 ./inc/edit.php

Author: Hauke Goos-Habermann (HHabermann@pc-kiel.de)
Description: routines for editing files with sed

8.15.1 EDIT_writeToFile

Description: Writes a text to a file on a client.
Parameter:

- file: Name of the file.
- text: The text to insert.

Returns:

- Code for writing a text to a file on a client.

8.15.2 EDIT_setOption

Description: Changes an option in a configuration file to a given value.
Parameter:

- file: Name of the file to change.
- option: Name of the option to change.
- value: Value to set.

Returns:

- sed code to change the option in the file to the value.

8.15.3 EDIT_genClientm23Random

Description: Generates BASH code to calculate a client-side random MD5 hash that is stored in the variable $m23Random.
Returns:

- BASH code to generate a random MD5 hash on client-side that is store

8.15.4 EDIT_commentoutInsert

Description: Comments out a matched line and inserts a new line after it.
Parameter:

- file: the name of the file
- search: Search string to match the line to comment out.
- lineToInsert: The text of the line to insert.
- commentStr: string to comment out (e.g. "#" for BASH or "//" for C/C++ style)

8.15.5 EDIT_countMatches

Description: Generates BASH code that counts how many times the search string can be found in the file. This can be used in ‘ ‘.
Parameter:

- file: file name
- search: search text

8.15.6 EDIT_calc

Description: calculates changes of the variable
Parameter:

- var: name of the BASH variable (e.g. $nr)
- calc: calculation that should be done with the var (e.g. incrementation: "+ 1")

8.15.7 EDIT_uncomment

Description: uncomments all with m23 commented lines of a file
Parameter:

- file: the name of the file

8.15.8 EDIT_commentout

Description: comments out lines in range or matching lines
Parameter:

- file: the name of the file
- from: start commenting out from this line
- to: stop commenting out at this line
- commentStr: string to comment out (e.g. "#" for BASH or "//" for C/C++ style)
- match:

8.15.9 EDIT_insertAfterLineNumber

Description: inserts a text AFTER a line number
Parameter:

- file: the name of the file
- lineNumber: reference line number for inserting
- insertText: text to insert
- addIfNotExists: set to true, if the the line should be added only if the line doesn't exist. false, if the line should be added on every execution.

8.15.10 EDIT_insertAtLineNumber

Description: inserts a text AT a line number
Parameter:

- file: the name of the file
- lineNumber: reference line number for inserting
- insertText: text to insert
- addIfNotExists: set to true, if the the line should be added only if the line doesn't exist. false, if the line should be added on every execution.

8.15.11 EDIT_insertLineNumber

Description: inserts a text AT or AFTER a line number
Parameter:

- file: the name of the file
- lineNumber: reference line number for inserting
- insertText: text to insert
- insertMode: "0" insert AT, "1" insert AFTER line number
- addIfNotExists: set to true, if the the line should be added only if the line doesn't exist. false, if the line should be added on every execution.

8.15.12 EDIT_searchLineNumber

Description: searches for the first line that contains "searchLine" and stores the line number in the BASH variable "m23searchLine"
Parameter:

- file: the name of the file
- searchLine: line to search
- startFrom: the line number to start searching from

8.15.13 EDIT_searchLastLineNumber

Description: searches for the last line that contains "searchLine" and stores the line number in the BASH variable "m23searchLine"
Parameter:

- file: the name of the file
- searchLine: line to search

8.15.14 EDIT_searchNextLineNumber

Description: searches for the next line number that contains "searchLine"
Parameter:

- file: the name of the file
- searchLine: line to search

8.15.15 EDIT_replace

Description: replaces $searchLine with $replaceText
Parameter:

- file: the name of the file
- searchLine: line to search
- repaceText: text to replace with
- mode: can be "g" to replace all matching lines. Any other value will only replace the first occurrence.

8.15.16 EDIT_prepareStr

Description: changes the string to make it compatible with sed
Parameter:

- str: string that should be changed. the string is read and written from/to this variable
- forSearch: set to true, if the string should be used as a search string

8.15.17 EDIT_savePerms

Description: saves the permissions and owner of a file
Parameter:

- file: the name to the file

8.15.18 EDIT_restorePerms

Description: restores previously saved file permissions and owner

8.15.19 EDIT_deleteLines

Description: Deletes lines from a given line number to a given line number
Parameter:

- file: the name to the file
- from: start deleting at this line number
- to: end deleting at this line number

8.15.20 EDIT_deleteLinesAmount

Description: Deletes N lines from a given line number
Parameter:

- file: the name to the file
- from: start deleting at this line number
- amount: the amount of lines to delete

8.15.21 EDIT_addIfNotExists

Description: Adds a new line if the search pattern cannot be found.
Parameter:

- file: the name to the file
- search: regular expression to search
- add: line to add

8.15.22 EDIT_deleteMatching

Description: Deletes all lines matching the regular expression
Parameter:

- file: the name to the file
- search: regular expression to search

8.16 ./inc/fdisk.php

Author: Hauke Goos-Habermann (HHabermann@pc-kiel.de)
Description: functions for partition and format of the client. print partition information and do the format itself.

8.16.1 FDISK_showFdiskCombinedGUIFunctions

Description: Shows the menu bar with integrated logic for FDISK_showCombinedFdiskGUIDialog.

8.16.2 FDISK_showCombinedFdiskGUIDialog

Description: Shows the new partition and formating screen.

8.16.3 FDISK_mdToEndOfArray

Description: Orders all MD devices from the input array to the end of the output array.
Parameter:

- in: Associative array with devices as keys and values (e.g Array ([/dev/md0] => /dev/md0 [/dev/sda1] => /dev/sda1 [/dev/sdb2] => /dev/sdb2)).

Returns:

- Associative array with devices as keys and values where the MDs are at the end (e.g. Array ([/dev/sda1] => /dev/sda1 [/dev/sdb2] => /dev/sdb2 [/dev/md0] => /dev/md0)).

8.16.4 FDISK_getFstabArray

Description: Gets the fstab of a client as array.
Parameter:

- client: Name of the client.

Returns:

- Array with the fstab (each line of the fstab as item).

8.16.5 FDISK_findFstabMountPointByDev

Description: Searches a client's fstab for a device and figures out the according mount point.
Parameter:

- fstabA: The fstab as array.
- dev: The device.

Returns:

- Mount point for the device.

8.16.6 FDISK_swapFilesystems

Description: Returns an array with the filesystems usable for swapping.

8.16.7 FDISK_formatInstallAndSwappart

Description: Adds jobs to format the installation and swap partitions and set the boot flag on the installation partition

8.16.8 FDISK_finalChecksAndRealPartitionAndFormatStart

Description: Does some final checks, starts the partitioning and formating and switches to the distribution selection page.

8.16.9 FDISK_installFilesystems

Description: Returns an array with the filesystems usable for installation.

8.16.10 FDISK_getUnusedMDs

Description: Returns an associative array with the unused MDs (e.g. /dev/md0, /dev/md1, ...) as key and value.
Parameter:

- param: parameter string containing status informations about the harddisks

Returns:

- Associative array with the unused MDs (e.g. /dev/md0, /dev/md1, ...) as key and value.

8.16.11 FDISK_listDrivesAndPartitions2

Description: Generates and defines a selection that contains all drives and partitions of a given client.
Parameter:

- param: parameter string containing status informations about the harddisks
- default: the drive to show first
- selName: the name the selection is called in PHP and HTML
- pathFilter: Set this to another value than false if you want only devices with a given string in it.
- filterOutSetRaidLvmLock: If set to true, drives and partitions with set raidLvmLock will not be listed.

Returns:

- String with the HTML selection.

8.16.12 FDISK_printAllBars2

Description: Shows the partition bars of all drives specified for the current client, that is stored in the session.

8.16.13 FDISK_showAllPartTables

Description: Shows the partition tables of all drives specified for the current client, that is stored in the session.

8.16.14 FDISK_fdiskSessionPartJobs

Description: Stores the partition jobs in the session.
Parameter:

- newJobs: The new partition jobs to set or false for not changing.

Returns:

- The current partition jobs.

8.16.15 FDISK_fdiskSessionInstPart

Description: Stores the installation partition in the session.
Parameter:

- newInstPart: The new installation partition to set or false for not changing.

Returns:

- The current installation partition.

8.16.16 FDISK_fdiskSessionSwapPart

Description: Stores the swap partition in the session.
Parameter:

- newSwapPart: The new swap partition to set or false for not changing.

Returns:

- The current swap partition.

8.16.17 FDISK_fdiskSessionSetter

Description: Generic function to store values in the client partition and format session.
Parameter:

- newVal: The value to set or false for not changing.
- varName: The name the value should be stored under in the client partition and format session.

Returns:

- The current value.

8.16.18 FDISK_fdiskSessionClient

Description: Returns the client name to partition and format.
Returns:

- The client name to partition and format.

8.16.19 FDISK_fdiskSessionParam

Description: Stores the partition parameters in the session.
Parameter:

- newParam: The new partition parameters to set or false for not changing.

Returns:

- The current partition parameters.

8.16.20 FDISK_fdiskSessionInstallDrive

Description: Stores the installation drive in the session.
Parameter:

- newDrive: The new installation drive to set or false for not changing.

Returns:

- The current installation drive.

8.16.21 FDISK_fdiskSessionvDevInstall

Description: Stores the internal virtual installation drive number in the session.
Parameter:

- newDrive: The new internal virtual installation drive number to set or false for not changing.

Returns:

- The current internal virtual installation drive number.

8.16.22 FDISK_fdiskSessionFreeSpaces

Description: Stores the free space parts of the installation drive in the session or recalculates them for the current installation drive.
Parameter:

- newSpaces: The new free spaces to set or false for not changing.
- reset: Re-calculate the free spaces, if set to true.

Returns:

- The current free spaces.

8.16.23 FDISK_fdiskSessionReset

Description: Sets back all session variables (client name optionally) for partitioning and formating a client.
Parameter:

- resetClientName: If set to true, the name of the client will be deleted too (and re-set by FDISK_fdiskSessionClient).

8.16.24 FDISK_fdiskSessionPartMethod

Description: Stores the partitioning method in the session.
Parameter:

- newMethod: The new method to set or false for not changing.

Returns:

- The current partitioning method.

8.16.25 FDISK_fdiskSessionPage

Description: Stores the page in the session.
Parameter:

- newPage: The new page to set or false for not changing.

Returns:

- The current page.

8.16.26 FDISK_fdiskSessionHelpPage

Description: Stores the help page in the session.
Parameter:

- newPage: The new help page to set or false for not changing.

Returns:

- The current help page.

8.16.27 FDISK_fdiskSessionTitle

Description: Stores the partitioning title in the session.
Parameter:

- newTitle: The new title to set or false for not changing.

Returns:

- The current partitioning title.

8.16.28 FDISK_fdiskSessionFstab

Description: Stores the fstab in the session.
Parameter:

- newFstab: The new fstab to set or false for not changing.

Returns:

- The current fstab.

8.16.29 FDISK_getPartitionByType

Description: Gets the FIRST partition matching a partition type.
Parameter:

- param: parameter string containing status informations about the harddisks
- vDev: Virtual (internally used) device number.
- type: type of the partition (primary, extended, logical)

Returns:

- Virtual partition number of the FIRST partition matching a partition type or false, if no partition was found.

8.16.30 FDISK_getDrivesAndPartitions

Description: Generates an array that contains all drives and partitions of a given client.
Parameter:

- param: parameter string containing status informations about the harddisks
- pathFilter: Set this to another value than false if you want only devices with a given string in it. If you add an "!" the beginning all is given out that doesn't contains the filter string (without the "!").
- addSizesAndTypes: If set to true, the array will contain the sizes, filesystems and types of the partitions and drives.
- filterOutSetRaidLvmLock: If set to true, drives and partitions with set raidLvmLock will not be listed.

Returns:

- Array with drives and partitions and (optionally) their sizes, filesystems and types.

8.16.31 FDISK_listDrivesAndPartitions

Description: Generates a selection that contains all drives and partitions of a given client.
Parameter:

- param: parameter string containing status informations about the harddisks
- default: the drive to show first
- selName: the name the selection is called in PHP and HTML
- pathFilter: Set this to another value than false if you want only devices with a given string in it.

Returns:

- String with the HTML selection.

8.16.32 FDISK_selectDrives

Description: creates a selection list of all drives
Parameter:

- param: parameter string containing status informations about the harddisks
- selName: the name the selection is called in PHP and HTML
- first: the drive to show first

8.16.33 FDISK_printAllBars

Description: showes the partitions bars of all available drives
Parameter:

- param: parameter string containing status informations about the harddisks
- fstabA: Associative array with fstab information

8.16.34 FDISK_getFirstDrive

Description: return the first drive as installation drive
Parameter:

- param: parameter string containing status informations about the harddisks

8.16.35 FDISK_formatPart

Description: formats a partition
Parameter:

- param: parameter string containing status informations about the harddisks
- dev: partition to format (e.g. /dev/hda1)
- type: type of filesystem
- partJobs: parted commands

8.16.36 FDISK_getBiggestValueOf

Description: gets the biggest value from a special type of partition
Parameter:

- param: parameter string containing status informations about the harddisks
- dev: selected device (e.g. hda)
- partType: type of the partition (logical, primary, extended)
- varType: define part of the key for the associative array (e.g. "type" means $param["dev$vDev"."part$vPart"."_type"])

8.16.37 FDISK_devNrExists

Description: checks if a certain device number exists
Parameter:

- param: parameter string containing status informations about the harddisks
- vDev: Virtual (internally used) device number.
- devNr: device numer you want to check

8.16.38 FDISK_nextLogicalDevNr

Description: gets the next free logical device number
Parameter:

- param: parameter string containing status informations about the harddisks
- vDev: Virtual (internally used) device number.

8.16.39 FDISK_nextPrimaryDevNr

Description: gets the next free primary device number
Parameter:

- param: parameter string containing status informations about the harddisks
- vDev: Virtual (internally used) device number.

8.16.40 FDISK_correctLogical

Description: corrects the order of the logical partitions after deleting $devNr.
Parameter:

- param: parameter string containing status informations about the harddisks
- vDev: Virtual (internally used) device number.
- devNr: the real device number to delete

8.16.41 FDISK_findDevNrPosition

Description: returns the device position for the new device
Parameter:

- start: start position for the search
- end: end position for the search
- param: parameter string containing status informations about the harddisks

- vDev: Virtual (internally used) device number.
- newPartNr: stores the new device number
- type: type of the new partition

8.16.42 FDISK_partCreationSelect

Description: retunes a selection for selecting a partition type to create.
Parameter:

- param: parameter string containing status informations about the harddisks
- vDev: Virtual (internally used) device number.

8.16.43 FDISK_canPartTypeBeCreated

Description: checks if a partition from a certain type can be created
Parameter:

- param: parameter string containing status informations about the harddisks
- vDev: Virtual (internally used) device number.
- type: type of the partition (primary, extended, logical)

8.16.44 FDISK_checkFreeSpace

Description: checks if there is a free space between $start and $end
Parameter:

- param: parameter string containing status informations about the harddisks
- vDev: Virtual (internally used) device number.
- type: type of the partition (primary, extended, logical)
- start: start position for the search
- end: end position for the search
- freeSpaces: array of the free space information

8.16.45 FDISK_installExistingDialog

Description: showes the dialog for installation on existing partitions
Parameter:

- param: parameter string containing status informations about the harddisks

8.16.46 FDISK_addFstab

Description: Adds a new entry to the fstab that is stored in the param array.
Parameter:

- fstab: Array that contains the fstab information. The changed fstab will be written to this parameter too.
- dev: Device to mount (e.g. /dev/hda1)
- mountpoint: Location where to mount the device (e.g. /mnt/hda1)

8.16.47 FDISK_delFstab

Description: Removes an entry from the fstab array.
Parameter:

- fstab: Array that contains the fstab information. The changed fstab will be written to this parameter too.
- fstabNr: Number of the fstab entry to delete.

8.16.48 FDISK_listFstab

Description: Generates a HTML table with all defined mountpoints.
Parameter:

- param: parameter string containing status informations about the harddisks

Returns:

- HTML table with the fstab.

8.16.49 FDISK_fstabAddDialog2

Description: Dialog for adding fstab entries. This version uses the param and fstab parameters from the session.

8.16.50 FDISK_fstabAddDialog

Description: Dialog for adding fstab entries.
Parameter:

- param: parameter string containing status informations about the harddisks
- fstab: Array that contains the fstab information. The changed fstab will be written to this parameter too.

8.16.51 FDISK_adjustFstabParam

Description: Adjust the parameter block of a fstab line to make it use an supported FS.
Parameter:

- param: The parameter block of a fstab line
- sourceName: The name of the package source list

Returns:

- Adjust the parameter block of a fstab line

8.16.52 FDISK_genManualFstab

Description: Generates commands to edit a given fstab, add new entries and remove old ones before.
Parameter:

- param: parameter string containing status informations about the harddisks
- mntPrefix: Prefix to set before the mountpoint (e.g. /mnt/m23root/)

8.16.53 FDISK_getBelongingRaidDev

Description: Searches for the RAID device, a physical partition belongs to, if it is part of a RAID.
Parameter:

- dev: The physical partition (e.g. /dev/hda4) that belongs to a RAID.
- param: parameter string containing status informations about the harddisks

Returns:

- The RAID device (e.g. /dev/md0) the physical partition belongs to or false, if no belonging RAID was found.

8.16.54 FDISK_delPart

Description: deletes a partition from the param string and generates the parted commands
Parameter:

- dev: the partition to delete (e.g. /dev/hda4)
- param: parameter string containing status informations about the harddisks
- partJobs: parted commands
- deleteBelongingRaid: If set to true, the RAID, the partition belongs to will be destroyed.

Returns:

- Changed param string.

8.16.55 FDISK_addPart

Description: adds a partition to the param string and generates the parted commands
Parameter:

- param: parameter string containing status informations about the harddisks
- partJobs: parted commands
- vDev: virtuell internal used device number.
- start: start position for the search
- end: end position for the search
- type: type of the partition (primary, extended, logical)
- freeSpaces: array of the free space information

Returns:

- : Changed param string.

8.16.56 FDISK_listPartitions

Description: lists the partitions (/dev/hda1, /dev/hda2, ...) of a device and generates a selection
Parameter:

- param: parameter string containing status informations about the harddisks
- vDev: Virtual (internally used) device number or -1, if all partitions on all devices should be listed.
- selName: name of the selection
- excludeType: type of partitions, not to show in the selection

Returns:

- HTML code for the selection.

8.16.57 FDISK_definePartitionSelection

Description: Defines a HTML selection with the partitions (/dev/hda1, /dev/hda2, ...) of a device
Parameter:

- param: parameter string containing status informations about the harddisks
- vDev: Virtual (internally used) device number or -1, if all partitions on all devices should be listed.
- selName: name of the selection
- excludeType: type of partitions, not to show in the selection

Returns:

- The selected partition.

8.16.58 FDISK_getPartitionsFromParam

Description: Returns an array with the partitions (/dev/hda1, /dev/hda2, ...) of a device
Parameter:

- param: parameter string containing status informations about the harddisks
- vDev: Virtual (internally used) device number or -1, if all partitions on all devices should be listed.
- selName: name of the selection
- excludeType: type of partitions, not to show in the selection

Returns:

- Selected partition.

8.16.59 FDISK_getAllDrives

Description: gets all drives of the client
Parameter:

- partitions: associative array containing status information about the harddisks

8.16.60 FDISK_colorFS

Description: get color for a selected filesystem
Parameter:

- fsName: name of the file system: ext3, ext2, linux-swap,...

8.16.61 FDISK_getPartitionPercent

Description: calculates the percent of a selected partition
Parameter:

- param: parameter string containing status informations about the harddisks
- vDev: Virtual (internally used) device number.
- vPart: Virtual (internally used) partition number. This is normally another number than the physical number (e.g. 1 on /dev/hda1)

8.16.62 FDISK_getAfterPartition

Description: calculates ??? of free size after a selected partition
Parameter:

- param: parameter string containing status informations about the harddisks
- vDev: Virtual (internally used) device number.
- vPart: Virtual (internally used) partition number. This is normally another number than the physical number (e.g. 1 on /dev/hda1)
- factor: the factor to multiplay percent amount of free space

8.16.63 FDISK_getBeforeFristPartition

Description: gets the free space before the first partition
Parameter:

- param: parameter string containing status informations about the harddisks
- vDev: virtual device number to access the drive
- factor: the factor to multiplay percent amount of free space

8.16.64 FDISK_getPartitions

Description: get the partition info for the client from db
Parameter:

- client: name of the client

8.16.65 FDISK_getPartInfoString

Description: Generates an info string, that shows information about the device name of the partition, its filesystem and bolonging to a RAID.
Parameter:

- vDev: Virtual (internally used) device number.
- vPart: Virtual (internally used) partition number. This is normally another number than the physical number (e.g. 1 on /dev/hda1)
- param: parameter string containing status informations about the harddisks
- fstabA: Associative array with fstab information.

Returns:

- Info string.

8.16.66 FDISK_getDriveInfoString

Description: Generates an info string, that shows information about the device name of the drive and bolonging to a RAID.
Parameter:

- vDev: Virtual (internally used) device number.
- param: parameter string containing status informations about the harddisks
- fstabA: Associative array with fstab information.

Returns:

- Info string.

8.16.67 FDISK_getDriveInfoIcon

Description: Generates HTML code for showing an icon with status information about a drive.
Parameter:

- vDev: Virtual (internally used) device number.
- param: parameter string containing status informations about the harddisks
- fstabA: Associative array with fstab information.

Returns:

- HTML code for showing an icon with status information about the drive.

8.16.68 FDISK_getPartInfoIcon

Description: Generates HTML code for showing an icon with status information about a drive or partition.
Parameter:

- vDev: Virtual (internally used) device number.
- vPart: Virtual (internally used) partition number. This is normally another number than the physical number (e.g. 1 on /dev/hda1) and if set to false, the icon and the status information will be generated for a drive and not for a partition.
- param: parameter string containing status informations about the harddisks
- fstabA: Associative array with fstab information.

Returns:

- HTML code for showing an icon with status information about the drive or partition.

8.16.69 FDISK_printBars

Description: prints the partitions as colored table
Parameter:

- param: parameter string containing status informations about the harddisks
- dev: selected device (e.g. /dev/hda)
- addJavaScript: Set to true to add JavaScript code that calls the JS function emptySpace(), if empty parts of the drive are clicked, selectPartition(), if a partition is clicked and showPartTable(), if the mouse is over the bar.
- fstabA: Associative array with fstab information.

8.16.70 FDISK_getSupportedFS

Description: Generates and returns an array with the list of supported file systems.
Returns:

- Array with the list of supported file systems.

8.16.71 FDISK_listSupportedFS

Description: lists all supported fileSystems for the menu.
Parameter:

- selName: name the selection list, used for the html form
- showFirst: name of file system shown first

Returns:

- The currently choosen file system.

8.16.72 FDISK_deletePartitionFromParam

Description: Deletes all partition parameters of a partition from param without correcting the other partitions.
Parameter:

- vDev: Virtual (internally used) device number.
- vPart: Virtual (internally used) partition number. This is normally another number than the physical number (e.g. 1 on /dev/hda1)
- param: parameter string containing status informations about the harddisks.

Returns:

- Changed param without the partition.

8.16.73 FDISK_virtualDeletePartition

Description: deletes partition from param assigned thru $vDev and $vPart.
Parameter:

- vDev: Virtual (internally used) device number.
- devNr: device number of the real device
- param: parameter string containing status informations about the harddisks

8.16.74 FDISK_virtualAddPartition

Description: adds a partition to the param param
Parameter:

- vDev: virtuell internal used device number.
- param: parameter string containing status informations about the harddisks
- start: start MB of the new partition
- end: end MB of the new partition
- type: type of the partition (primary, extended, logical)
- devNr: returnes the device number

8.16.75 FDISK_listPartTable

Description: lists the partition information as table
Parameter:

- vDev: Virtual (internally used) device number.
- param: parameter string containing status informations about the harddisks

8.16.76 FDISK_listInstPartSelector

Description: lists all partitions to select for swap and install partition
Parameter:

- param: parameter string containing status informations about the harddisks
- default: partition that should be shown as selected
- type: array with filesystems that are possible for installation or swap
- selName: name of the selection

8.16.77 FDISK_formatExisting

Parameter:

- instPart: partition to put the operation system on (e.g. /dev/hda1)
- swapPart: partition to put the swap file system on (e.g. /dev/hdb2)
- command: parted commands to do the installation
- param: parameter string containing status informations about the harddisks

8.16.78 FDISK_getvPart

Description: returns vPart with the real device number.
Parameter:

- param: parameter string containing status informations about the harddisks
- dev: selected device (e.g. hda)
- devNr: number of partition

8.16.79 FDISK_rmJob

Description: generates a partition remove job
Parameter:

- dev: selected device (e.g. hda)
- devNr: number of partition, minor number in parted
- partJobs: associative array with partition jobs

8.16.80 FDISK_addJob

Description: generates a partition add job
Parameter:

- path: selected device (e.g. /dev/hda)
- start: start point fo the partition
- end: end point fo the partition
- type: type of the partition (primary, logical)
- partJobs: associative array with partition jobs
- fullPath: full path for the partition
- devNr: number of the device (e.g. 1 with /dev/hda1)

8.16.81 FDISK_bootflagJob

Description: enables the booting flag on a partition
Parameter:

- path: device to activate booting on (e.g. /dev/hda1)
- devNr: number of partition, minor number in parted
- partJobs: associative array with partition jobs

8.16.82 FDISK_formatJob

Description: generates a partition format job
Parameter:

- path: device to format (e.g. /dev/hda1)
- fileSys: file system of the partition: ext3, ext2, linux-swap
- partJobs: associative array with partition jobs

8.16.83 FDISK_countPartitions

Description: count all partitions of a selected type
Parameter:

- param: parameter string containing status informations about the harddisks
- vDev: Virtual (internally used) device number.
- type: type of the partition (primary, extended, logical)

8.16.84 FDISK_getFreeSpaces

Description: get free spaces as array
Parameter:

- param: parameter string containing status informations about the harddisks
- dev: selected device (e.g. hda)

8.16.85 FDISK_autoPart

Description: generation of param string and command list for automatic partition
Parameter:

- clientName: name of the client
- command: parted commands are written to $command
- dev: selected device (e.g. /dev/hda)
- param: parameter string containing status informations about the harddisks
- instPart: the variable the installation device name is written to
- swapPart: the variable the swap device name is written to
- minSwap: Minimal size of the swap partition in MB.
- maxSwap: Maximal size of the swap partition in MB.

8.16.86 FDISK_printColorDefinitions

Description: prints the color definitions for the filesystems

8.16.87 FDISK_showDiskDefine

Description: shows a dialog for defining the type and size of the fake drive for the clientBuilder
Parameter:

- client: client name

8.16.88 FDISK_defineDrive

Description: defines drive information for the clientBuilder
Parameter:

- client: client name
- path: path to the drive (/dev/hda, /dev/hdb, ...)
- size: size of the drive in MB
- upperI: upper tolerance border for disks with identical type
- lowerI: lower tolerance border for disks with identical type
- upperO: upper tolerance border for disks with other type
- lowerO: lower tolerance border for disks with other type
- asSpeciefied: use the speciefied disk, if it exists (is set to "yes" or empty)
- sizeAdjustmentType: defines how the partitions should be adjusted, if there is more or less space on the client that the defined one. "system" increases or tries to keep the size of the system partition. "percentage" makes a percentage adjustment of all partitions.

8.16.89 FDISK_dev2LDevLPart

Description: searches a special device (e.g. /dev/hda2) and writes the virtual device and partition numbers to the variables. These values can be used to access the file system via $param["dev$vDev"."part$vPart"."_fs"]
Parameter:

- param: the associative array containing all values describing the drives of the client
- dev: the device (e.g. /dev/hda2)
- vDev: the virtual device number, that is used to build the variable name to access the associative array.
- vPart: the virtual partition number, that is used to build the variable name to access the associative array. This number has not to be qual to the partition number of the real drive (e.g. /dev/hda5 can be $vPart == 3). If it is set to "empty", only vDev is calculated.

Returns:

- true if the search worked otherwise false.

8.16.90 FDISK_rereadPartTable

Description: Let the OS re-read the partition table.
Parameter:

- path: The device that was changed/created (e.g. /dev/sda5).

8.16.91 FDISK_genPartedCommands

Description: returnes the partition and formation commands that are generated from partJobs.
Parameter:

- partJobs: string with information about all created partition jobs
- mkfsextOptions: Extra parameter for mkfs.extX .

8.16.92 FDISK_listPartJobs

Description: print all part jobs in the table
Parameter:

- partJobs: string with information about all created partition jobs

8.16.93 FDISK_getDiskType

Description: returnes the type of the drive (DISK_TYPE_IDE, DISK_TYPE_SCSI)
Parameter:

- path: the path to the device (e.g. /dev/hde)

8.16.94 FDISK_getDriveAndNr

Description: splits a path (e.g. /dev/hda1) in the device (/dev/hda) and the device number (1). The device is returned as element 0 and the number as element 1 in an array.
Parameter:

- path: the path to partition (e.g. /dev/hde1)

Returns:

- Array with two parts. $out[0]=drive (e.g. /dev/hda), $out[1]=the device number

8.16.95 FDISK_getNextFdiskFormatJobNr

Description: returnes the next free job number for the parameters of a m23fdiskFormat job. (e.g. there are used the following parameters: command0 = rm, command1= add. Then the next command number to use will be command2 => return value is 2)
Parameter:

- fdiskParams: the parameters of the m23fdiskFormat job

8.16.96 FDISK_AFPselectDrive

Description: selects a drive from the settings in "options" and from available drives.
Parameter:

- drives: all drives available on the client
- options: options array of the client

8.16.97 FDISK_AFPlinearScale

Description: scales all partitions sizes to match the full disk size.
Parameter:

- driveToUse: device to use (e.g. /dev/hda)
- driveToUseSize: Size of the real drive to use.
- options: options array of the client
- command: array that stores the modificated format parameters
- formatarr: array that contains the original format parameters

8.16.98 FDISK_AFPgetPartSizes

Description: writes the sizes of the installation, swap and other partitions to the variables.
Parameter:

- formatArr: array that contains the oformat parameters
- options: options array of the client
- instPartSize: stores the size of the installation partition
- instSwapSize: stores the size of the swap partition
- otherSize: stores the size of the other partition(s)

8.16.99 FDISK_adjustFdiskParams

Description: adjusts the installation and swap drive for a derived client, based on the defined client settings
Parameter:

- path: the path to the device (e.g. /dev/hde)

8.16.100 FDISK_virtualAddDrive

Description: Adds a new drive to the param array.
Parameter:

- param: parameter string containing status informations about the harddisks
- path: Device name of the new drive (e.g. /dev/md0)
- size: Size in MB of the new drive.

8.16.101 FDISK_getDrivePartitionSize

Description: Calculates the size of a drive or partition.
Parameter:

- vDev: Virtual (internally used) device number.
- vPart: Virtual (internally used) partition number. This is normally another number than the physical number (e.g. 0 on /dev/hda1)
- param: parameter string containing status information about the harddisks

Returns:

- Size of the drive or partition in MB.

8.16.102 FDISK_listRaidTable

Description: Get informations about the assigned real drives/partitions of a RAID.
Parameter:

- raidDev: Device name of the new drive (e.g. /dev/md0)
- param: parameter string containing status information about the harddisks

Returns:

- HTML table with informations about the assigned real drives/partitions.

8.16.103 FDISK_addDrivePartitionToRaid

Description: Adds a drive or partition to a RAID.
Parameter:

- raidDev: Device name of the new drive (e.g. /dev/md0)
- raidType: RAID level (this can be 0,1,4,5,6,10)
- partitionDrive: Partition or drive to add (e.g. /dev/hdc)
- param: parameter string containing status information about the harddisks
- raidMode: Raid mode (e.g. 1 for RAID-1, 5 for RAID-5)

Returns:

- true if the RAID has the minimum amount of assigned drives/partitions and otherwise false.

8.16.104 FDISK_buildRaidDialog

Description: Shows a dialog for creating RAIDs of different types and assign real drives or partitions.
Parameter:

- param: parameter string containing status information about the harddisks
- partJobs: associative array with partition jobs
- currentDrive: the current drive to work on (e.g. hda)
- helpPage: Name of the help page to show.
- partitionDrive: Partition or drive to add (e.g. /dev/hdc)
- partMethod: partition method (used for the partition/format dialog). The next step will depend on this value.

Returns:

- true if the RAID has the minimum amount of assigned drives/partitions and otherwise false.

8.16.105 FDISK_addRaidJobs

Description: Generates jobs to create all RAIDs
Parameter:

- partJobs: associative array with partition jobs
- param: parameter string containing status information about the harddisks

8.16.106 FDISK_addRaidBeforeFormat

Description: Generates and places a job to create a RAID on given drives/partitions before the formating of the RAID device.
Parameter:

- raidDev: RAID device (e.g. /dev/md0)
- devList: Space separated list of devices to create the RAID on top (e.g. /dev/sda1 /dev/hda /dev/sbd5).
- partJobs: associative array with partition jobs.
- raidMode: The type of the RAID (0,1,5, ...)

8.16.107 FDISK_raidJob

Description: Generates a job to create a RAID on given drives/partitions.
Parameter:

- raidDev: RAID device (e.g. /dev/md0)
- devList: Space separated list of devices to create the RAID on top (e.g. /dev/sda1 /dev/hda /dev/sbd5).
- partJobs: associative array with partition jobs

8.16.108 FDISK_virtualDeleteDrive

Description: Deletes a (RAID) drive from param assigned thru $vDev.
Parameter:

- vDev: Virtual (internally used) device number.
- param: parameter string containing status informations about the harddisks

8.16.109 FDISK_deleteDriveFromParam

Description: Deletes all drive and partition parameters of a drive from param without correcting any order.
Parameter:

- vDev: Virtual (internally used) device number of the drive to delete.
- param: parameter string containing status informations about the harddisks.

Returns:

- Changed param without the drive.

8.17 ./inc/groups.php

Author: Hauke Goos-Habermann (HHabermann@pc-kiel.de)
Description: group management funtions

8.17.1 GRP_exists

Description: checks, if a group exists
Parameter:

- groupName: name of the group to check

8.17.2 GRP_add

Description: adds a group
Parameter:

- groupName: name of the group to add

8.17.3 GRP_getIdByName

Description: gets the Id of a groupname
Parameter:

- groupName: name of the group

Returns:

- Group number or false, if no matching group is found.

8.17.4 GRP_del

Description: deletes all clients from the group and the group itself
Parameter:

- groupName: name of the group

8.17.5 GRP_isClientInGroup

Description: returnes true, if a client is in the selected group, otherwise false
Parameter:

- clientName: name of the client
- groupName: name of the group

8.17.6 GRP_addClientToGroup

Description: adds a client to a group
Parameter:

- clientName: name of the client
- groupName: name of the group

8.17.7 GRP_delClientFromGroup

Description: removes a client from a group
Parameter:

- clientName: name of the client
- groupName: name of the group

8.17.8 GRP_listGroupsAndCount

Description: returnes a associative array with all groupnames and the amount of clients in each group

8.17.9 GRP_showGroupsAndCount

Description: generates a table with all groupnames and the amount of clients in each group

8.17.10 GRP_ren

Description: renames a group
Parameter:

- groupName: name of the group
- newGroupName: name of the new group

8.17.11 GRP_HTMLBackToDetails

Description: generates HTML code to return to the group details page
Parameter:

- groupName: name of the group
- section: name of the section to jump to

8.17.12 GRP_countClients

Description: returns the amount of client of a certain group
Parameter:

- groupName: name of the group

8.17.13 GRP_showGeneralInfo

Description: shows a table with general information about the group
Parameter:

- groupName: name of the group

8.17.14 GRP_showRenDialog

Description: shows a dialog to rename a group

8.17.15 GRP_moveClientToGroup

Description: moves a client from one group to another
Parameter:

- clientName: client to move
- oldGroup: name of the old group
- newGroup: name of the new group

8.17.16 GRP_listGroups

Description: returnes all groups in an array

8.17.17 GRP_groupSelection

Description: generates a HTML selection with all groups as options
Parameter:

- selName: name of the selection
- first: the group that should be shown first

8.17.18 GRP_showDelDialog

Description: shows a dialog for deleting a group
Parameter:

- groupName: name of the group

8.17.19 GRP_doClientMoreGroups

Description: dialog and logic for adding and removing the client to and from multiple groups
Parameter:

- clientName: name of the client
- type: type of the action ("add" for adding,"del" for removing)

8.17.20 GRP_listClientGroups

Description: returnes an array containing all groups a client is in
Parameter:

- clientName: name of the client

8.17.21 GRP_showClientGroups

Description: Shows a list containing all groups a client is in
Parameter:

- clientName: name of the client
- link: if there should be links to the group page
- output: If set to true, the list will be shown, if set to false returned instead.

Returns:

- Nothing or the list containing all groups a client is in.

8.17.22 GRP_listAllClientsInGroup

Description: returns an array that consists of all client names that are in a group
Parameter:

- groupName: name of the group to check

8.17.23 GRP_getDistrsAndSourcesLists

Description: writes the differnt distributions and package sources of the clients in a group as array to the both variables
Parameter:

- distrs: variable that should store the distributions
- sourceslists: variable that should store the sourceslist names

8.17.24 GRP_showSelDistrSources

Description: shows a dialog for selection of distribution and package source name. The choices are taken form distr and packagesource values of the clients in the group. If there is only one entry for one or both of the values, the value is written back to the input variable otherwise a HTML selection is shown.
Parameter:

- groupNames: group names stores in an array
- distr: distribution to show first and variable to write the distribution name back
- sourceslist: sources list to show first and variable to write the sources list name back

8.17.25 GRP_listAllClientsInGroups

Description: returns an array with all client names contained in the groups
Parameter:

- groupNames: the names of the groups stored in an array

8.17.26 GRP_HTMLBackToOverview

Description: generates HTML code to return to the group overview page

8.17.27 GRP_getAllPackages

Description: shows a list of all packages on all clients in the selected groups. the packages can be selected by checkboxes
Parameter:

- groupNames: group names stores in an array
- key: keyword for searching for packages
- withActions: you can select to draw te action

8.17.28 GRP_desasterRecovery

Description: recovers all selected clients and shows a message afterwards
Parameter:

- clients: an array containing all clients that should be recovered

8.18 ./inc/helper.php

Author: Hauke Goos-Habermann (HHabermann@pc-kiel.de)
Description: Helper functions that did not fit into another include file.

8.18.1 HELPER_URIencode

Description: Encodes a string like the JavaScript function URIencode would do it.
Parameter:

- in: Input string to be encoded.

Returns:

- Encoded version of the string

8.18.2 HELPER_isUpper

Description: Checks, if a character is upper case
Parameter:

- char: Character to check.

Returns:

- true, when upper case otherwise false

8.18.3 HELPER_filesize

Description: Gets the correct file size of a file, even if it is bigger than 2 GB.
Parameter:

- fileName: Name of the file with full path.

Returns:

- The file size of the file in bytes.

8.18.4 HELPER_isExecutedInCLI

Description: Checks, if it is run in CLI.
Returns:

- True, when run in CLI otherwise false.

8.18.5 HELPER_isExecutedOnUCS

Description: Checks, if it is run on UCS.
Returns:

- True, when run on UCS otherwise false.

8.18.6 HELPER_getContentFromURL

Description: Downloads an URL via curl and gives back the site code.
Parameter:

- url: The URL to download.
- range: If set, a part of the file will be downloaded. (e.g. 0-500 will download the first 500 kb)

Returns:

- The downloaded site code or false in case of an error.

8.18.7 HELPER_trimValue

Description: Runs trim on the input parameter and writes the result back.
Parameter:

- value: Value to trim.

8.18.8 HELPER_xargsRecursive

Description: Executes a BASH command with a list of arguments. If the BASH command fails, the argument list is split in two parts and recursively executed again.
Parameter:

- cmd: BASH command
- argsA: Array of commands for the BASH command.
- tabAmount: Amount of identing to start with.

Returns:

- Recursive calls of the BASH command with error checking.

8.18.9 HELPER_getNewLogLines

Description: Gets the last (new) lines of a (growing) log file.
Parameter:

- log: Name of the log file.
- sessionPrefix: Prefix for storing the last read line number in the session.

Returns:

- UTF8-encoded new lines of the log file.

8.18.10 HELPER_rmRecursive

Description: Removes a directory with sub-directories and contained files.
Parameter:

- : dir: Full path to the directory.

8.18.11 HELPER_showBAfH

Description: Shows the German BAfH excuse of the day.

8.18.12 HELPER_ucrc32

Description: Returns the unsigned crc32 sum of an input value.
Parameter:

- : in: Input to crc.

Returns:

- Unsigned crc32 sum of an input value.

8.18.13 HELPER_md5x5

Description: Hashes an input value 5 times with MD5.
Parameter:

- : in: Input to hash.

Returns:

- Hashed value.

8.18.14 HELPER_netmaskAmountOfSetBits

Description: Calculates the amount of set bits in a network mask (as used in the short form of netmasks).
Parameter:

- : nm: The netmask in decimal notation.

Returns:

- Amount of set bits in the network mask.

8.18.15 HELPER_networkCalculator

Description: Calculates the network IP by a given IP and the netmask.
Parameter:

- : ip: The IP.
- : nm: The netmask.

Returns:

- Network IP.s

8.18.16 HELPER_netmaskCalculator

Description: Converts a short netmask (e.g. 24 for 255.255.255.0) into the decimal notation.
Parameter:

- : nm: The netmask to convert. If a netmask in decimal notation is given, no conversation is done.

Returns:

- Netmask in decimal notation.

8.18.17 HELPER_importAllIntoPOST

Description: Imports all values into the $_POST array in case that there are too much array keys for the normal processing.

8.18.18 HELPER_randomUsername

Description: Generates a random username with a given length.
Parameter:

- length: Length of the username to create.

Returns:

- The random username.

8.18.19 HELPER_createSelfSignedCAAndServerCertificate

Description: Creates a selfsigned CA and a server certificate.
Parameter:

- CADn: Information about the CA.
- serverDn: Information about the server.
- password: Password for the private server key.
- expirationDate: Expiration of the certificates in days.

Returns:

- Associative array with the certificate of the CA, the certificate and private key of the server.

8.18.20 HELPER_arrayReOrderKeynumbers

Description: Changes all keys of the input array to simple ascending numbers, if the key of the inpur array is a number (if not, the key will be left unchanged). The order of the keys is preserved.
Parameter:

- inArray: The input array.

Returns:

- New array with the ascending key numbers.

8.18.21 HELPER_arrayInsertBeforeKeynumber

Description: Inserts a value into an array (that has simple numbers as keys) before a given key.
Parameter:

- inArray: The input array.
- beforeKeynumber: The key number the value should be inserted before.
- val: The value that should be inserted.

Returns:

- New array with the new value inserted.

8.18.22 HELPER_arrayInsertAfterKeynumber

Description: Inserts a value into an array (that has simple numbers as keys) after a given key.
Parameter:

- inArray: The input array.
- afterKeynumber: The key number the value should be inserted after.
- val: The value that should be inserted.

Returns:

- New array with the new value inserted.

8.18.23 HELPER_m23Array2Array

Description: Converts an m23 array to a normal array.
Parameter:

- m23Array: The m23 array to convert. The m23 array is a 2D array, that consists of keys build of a parameter names combined with a parameter number. Parameter names with the same parameter number belong together. (e.g. [command0] => format, [path0] => /dev/md0, [fs0] => ext4, ...)

Returns:

- A normal array, that may be edited more easyly. (e.g [0] => Array([command] => format, [path] => /dev/md0, [fs] => ext4))

8.18.24 HELPER_array2m23Array

Description: Converts a normal array to an m23 array.
Parameter:

- array: A normal array, that may be edited more easyly. (e.g [0] => Array([command] => format, [path] => /dev/md0, [fs] => ext4))

Returns:

- The m23 array, that is a 2D array, that consists of keys build of a parameter names combined with a parameter number. Parameter names with the same parameter number belong together. (e.g. [command0] => format, [path0] => /dev/md0, [fs0] => ext4, ...)

8.18.25 print_r2

Description: Function like print_r, but sorts the entries, if the input is an array and converts newlines to HTML breaks.
Parameter:

- in: Value to print.

8.18.26 HELPER_debugBacktraceToFile

Description: Writes/Appends debug information about all calling functions and parameters into a file.
Parameter:

- file: File name with full path, where the debug information should be stored.

8.18.27 HELPER_getRemoteFileContents

Description: Downloads a file if it is not older than a given time and returns its contents.
Parameter:

- url: The URL where to download the file from.
- storeFile: The file name to store the download in.
- refreshTime: The time in minutes the file is downloaded again.
- forceOverwrite: Set to true if the file should be overwritten even if the new file is epmty.

Returns:

- The contents of the files from chache or from download or false if no file could be found.

8.18.28 HELPER_passGenerator

Description: Generates semi-random passwords via pwgen or DB_genPassword.
Parameter:

- length: The length of the passwords.
- amount: The amount of passwords to generate.

Returns:

- Array with the generated passwords if $amount > 1 or the password string directly if $amount = 1.

8.18.29 HELPER_array2AssociativeArray

Description: Copies the values of an array as keys AND values to a new assiciative array.
Parameter:

- in: Input array.

Returns:

- Associative array with equal keys and values.

8.18.30 HELPER_randomMAC

Description: Generates a random MAC address.
Returns:

- Random MAC address in the format XX:XX:XX:XX:XX:XX (e.g. 70:c4:d4:49:6e:27).

8.18.31 HELPER_generateSalt

Description: Generates a random salt string.
Parameter:

- length: Length of the salt.

Returns:

- Random salt of given length.

8.18.32 HELPER_grubMd5Crypt

Description: Encrypts a password to the MD5 hash as expected by grub.
Parameter:

- password: Password to encrypt.
- length: Length of the salt.

Returns:

- Encrypted password in grub style or false if MD5 hash function isn't available.

8.18.33 HELPER_listFilesInDir

Description: Lists all files in a directory and returns an array with all file names.
Parameter:

- dirname: Name of the directory.

Returns:

- Array with all file names.

8.18.34 HELPER_getBootLoaders

Description: Returns a list of available bootloaders.
Returns:

- Array with available bootloaders.

8.18.35 HELPER_getTimeZones

Description: Searches for all time zones.
Parameter:

- country: two letter country name that is used to select a timezone if none is set with $first.

Returns:

- Array with all time zones.

8.18.36 HELPER_calcMBSize

Description: calculates the size in MB from a given input that can be a GB value or measured in **Parameter:**

- number: the number to convert
- from: if number is a percent value, the output will be the percentage of the from value
- trunc: set to true, if the output value should be trunced

8.18.37 HELPER_grep

Description: returnes all lines from $string seperated by $cut that contain $search
Parameter:

- string: the text, that should be searched
- search: the string to be searched
- cut: seperator for the input and output lines

Returns:

- The found lines as string separated by $cut.

8.18.38 HELPER_grepNot

Description: Returnes all lines from $string seperated by $cut that do NOT contain $search.
Parameter:

- string: the text, that should be searched
- search: the string to be searched
- cut: seperator for the input and output lines

Returns:

- The found lines as string separated by $cut.

8.18.39 HELPER_grepCount

Description: Counts the lines from $string seperated by $cut that contain $search.
Parameter:

- string: the text, that should be searched
- search: the string to be searched
- cut: seperator for the input and output lines

Returns:

- Amount of lines that match the $search.

8.18.40 HELPER_getFdiskMountPoints

Description: Returnes an array with all mount points listed in /etc/fstab
Parameter:

- excludeExtra: set to true, if you want to exclude /proc and /sys from the array

Returns:

- Found mount points as array keys and values.

8.18.41 HELPER_getApacheUser

Description: returnes the name of the Apache user

8.18.42 HELPER_getApacheGroup

Description: returnes the group of the Apache user

8.18.43 HELPER_putFileContents

Description: Writes data to a file.
Parameter:

- fileName: name of the file to write
- contents: Text or data that should be written to the file.

Returns:

- Error code from fwrite.

8.18.44 HELPER_getFileContents

Description: returnes the contents of a file (the file is read to a maximum of 5 MB)
Parameter:

- fileName: name of the file to read

8.18.45 HELPER_showFileContents

Description: Shows the contents of a file (the file is read to a maximum of 5 MB)
Parameter:

- fileName: name of the file to read

8.18.46 HELPER_maxPhpUploadSize

Description: Returns the maximum file upload size allowed by php.ini.

8.18.47 HELPER_compareLengthAbc

Description: Compares the length of two strings and then by alphabet
Parameter:

- $a: string of a certain length
- $b: string of a certain length

8.18.48 HELPER_sortByLength

Description: Sorts an array by length of its values, shortest value first, keeping key-value pairs
Parameter:

- $array: The array you want to sort by length

8.19 ./inc/help.php

Author: Hauke Goos-Habermann (HHabermann@pc-kiel.de)
Description: function to show the help box in the correct language

8.19.1 HELP_showHelp

Description: showes the help block for the online help
Parameter:

- topic: the name of the help file
- language: two character language description (e.g. de, en, fr,...)

8.19.2 HELP_getHelp

Description: Returnes the help block for the online help
Parameter:

- topic: the name of the help file or name of a man page starting with "man://" e.g. man://tar
- language: two character language description (e.g. de, en, fr,...)
- fileName: full path to a help file in a directory with language short name

Returns:

- help block string

8.19.3 HELP_getHelpString

Description: returns the help block for the online help
Parameter:

- topic: the name of the help file
- language: two character language description (e.g. de, en, fr,...)

8.19.4 HELP_showHelpTex

Description: shows the help file converted to LaTeX code
Parameter:

- fileName: name of the help file

8.20 ./inc/html.php

Author: Hauke Goos-Habermann (HHabermann@pc-kiel.de)
Description: functions for generating often used HTML code

8.20.1 HTML_imgSwitch

Description: Defines an image button with two states and a text next to it.
Parameter:

- $htmlName: Name of the html image input element.
- $off_img: Name and path of image to be displayed if its state is "off"
- $on_img: Name and path of image to be displayed it its state is "on"
- $off_text: Text to be displayed if state is "off"
- $on_text: Text to be displayed if state is "on"
- $separator: Anything which shall be displayed between the picture (clickable) and the text (not clickable)
- $default: State of the image input element on first load of page ("on" or "off")
- $outState: Current state of element (true for "on" or false for "off").

Returns:

- true, if the button was clicked otherwise false.

8.20.2 HTML_uploadFile

Description: Shows a dialog for uploading image files.
Parameter:

- htmlName: Name of the HTML element
- label: The visual naming of the HTML element.
- maxFileSize: The maximum allowed filesize in bytes.

Returns:

- The full path to the uploaded file or false in case of an error.

8.20.3 HTML_urlButton

Description: Defines a link that appears like a button.
Parameter:

- htmlNames: Name of the constant.
- label: Label of the button.
- url: The URL where the link button should point to.

8.20.4 HTML_sourceViewer

Description: Creates a source code viewer area with syntax highlighting.
Parameter:

- htmlNames: Name of the source viewer.
- code: The source code to show.
- highlighting: The GeSHi language of the source code (e.g. bash).

8.20.5 HTML_getOutputBuffer

Description: Gets the complete (previously rendered) HTML output buffer.
Returns:

- The complete contents of the (previously rendered) HTML output buffer.

8.20.6 HTML_setOutputBuffer

Description: Sets (replaces) the complete (previously rendered) HTML output buffer, that will be sent to the webbrowser.
Parameter:

- HTMLOutputBuffer: New HTML codes that should replace the complete current output buffer.

8.20.7 HTML_AJAXAutoSubmit

Description: Defines AJAX code that clicks a submit button when the given URL returns 'submit'. The defined constant should be used as LAST part of the $extra parameter in the HTML_submit function.
Parameter:

- htmlNames: Name of the HTML submit element (button).
- url: The HTTP URL to poll.
- timeout: Time in milliseconds to poll the URL for new status.

Returns:

- Constant name to insert into $extra.

8.20.8 HTML_liveSpan

Description: Creates a span that updates itself via AJAX by polling from a given URL.
Parameter:

- htmlNames: Name of the span.
- url: The HTTP URL to poll.
- staticValue: Value that should be shown, when AJAX is not available (e.g. when JavaScript is disabled)
- timeout: Time in milliseconds to poll the URL for new status.

8.20.9 HTML_manipulateOutputBuffer

Description: Manipulates the output buffer with already generated HTML code and replaces all occurrences the search term with the replace term.
Parameter:

- search: The search term.
- replace: The replace text.

Returns:

- true, if the search term was found.

8.20.10 HTML_showTitle

Description: Shows a title.
Parameter:

- title: Text of the title.

8.20.11 HTML_showSmallTitle

Description: Shows a title of the second tier.
Parameter:

- title: Text of the small title.

8.20.12 HTML_hiddenVar

Description: Create a hidden HTML variable to store values in an HTML form.
Parameter:

- var: Name of the hidden variable.
- val: Its value.

8.20.13 HTML_liveLogArea

Description: Creates a log area that updates itself via AJAX by polling from a given URL.
Parameter:

- htmlNames: Name of the log area.
- width: The width in characters of the log area.
- height: The height in characters of the log area.
- url: The HTTP URL to poll.
- timeout: Time in milliseconds to poll the URL for new status.
- maxLines: Maximum amount of lines to show in the log area (older lines are removed, when there are too many).

8.20.14 HTML_checkboxChangerButtons

Description: Defines buttons for changing all check boxes.
Parameter:

- htmlNames: Name of the HTML elements.

8.20.15 HTML_jsCheckboxChanger

Description: Generates JavaScript code for changing all check boxes.
Parameter:

- jsBlockName: Name of the JS block constant.

8.20.16 HTML_logArea

Description: Shows a text area for log information (readonly).
Parameter:

- htmlName: Name of the HTML element.
- cols: Number of columns.
- rows: Number of rows to show.
- text: The log information to show.

8.20.17 HTML_getQuestionnaireURL

Description: Returns the complete URL to the m23 questionnaire in the language of the webinterface.
Returns:

- Complete URL to the m23 questionnaire in the language of the webinterface.

8.20.18 HTML_questionnaire

Description: Shows the questionnairem window.
Parameter:

- disable: Set to true to disable showing of the window again.

8.20.19 HTML_esel

Description: Shows an dog-ear that can be opened to show "goos-habermann.de/m23ad".

8.20.20 HTML_JSMenuCloseAllEntries

Description: Closes all menu entries for a menu. This should be called at the end of a page to get it executed after loading.

8.20.21 HTML_JSMenuOpener

Description: Opens a menu entry when moving the mouse over the title and closes all other entries of the same menu.
Parameter:

- menuName: Name of the complete menu. This name must be the same on all entries belonging to the same menu.
- entryName: Name of the entry. This name must be unique.
- title: Title for the menu entry.
- html: HTML code of the menu entry. Here can stand all that is expressable with HTML. This part is shown and hidden.

- titleCSS: CSS class for marking the title.

Returns:

- The HTML code for displaying the menu entry.

8.20.22 HTML_jQueryMenu

Description: Creates an entry for the jQuery accordion menu
Parameter:

- title: Title for the menu entry.
- html: HTML code of the menu entry. Here can stand all that is expressable with HTML. This part is shown and hidden.
- titleCSS: CSS class for marking the title.

Returns:

- The jQuery code for the menu entry.

8.20.23 HTML_jQueryMenuHeader

Description: Generate code for beginning a the jQuery accordion menu.
Parameter:

- menuName: Name of the menu.

Returns:

- The jQuery code for beginning the menu.

8.20.24 HTML_jQueryReStoreYWindowPosition

Description: Generates jQuery code for storing the Y scroll position of the window and to restore the position after a submit.
Parameter:

- variablePrefix: Prefix for the hidden variable with the Y position.
- hiddenPosCode: Variable where the hidden variable HTML code is written to.

Returns:

- The jQuery code for storing the Y scroll position of the window and to restore the position after a submit.

8.20.25 HTML_jQueryMenuEnd

Description: Generates code for ending a the jQuery accordion menu.
Parameter:

- menuName: Name of the menu.

Returns:

- The jQuery code for ending the menu.

8.20.26 HTML_incStatusBarPercentByName

Description: Increments the status bar percent by a given amount.
Parameter:

- statusBarName: The name of the status bar.
- client: The name of the client, the status bar belongs to (or other values for identifying the object the status bar belongs to)
- percent: Percent value of the current job.

8.20.27 HTML_setStatusBarPercentPointByName

Description: Calculates the value of a percent point according to the amount of waiting packages and stores the result in the DB.
Parameter:

- statusBarName: The name of the status bar.
- client: The name of the client, the status bar belongs to (or other values for identifying the object the status bar belongs to)
- recalculate: true, if the remaining percent value of the status bar should be used to calculate a new (better fitting) percentpoint.

Returns:

- false on errors, otherwise true.

8.20.28 HTML_setStatusBarStatusByName

Description: Sets new percent value and/or new status text by clientname AND status bar name.
Parameter:

- statusBarName: The name of the status bar.
- client: The name of the client, the status bar belongs to (or other values for identifying the object the status bar belongs to)
- percent: Percent value to write into the DB (may be false, if it should not be changed).
- statustext: A text message that should be shown under the status bar and written to the DB (may be false, if it should not be changed).

Returns:

- : false on parameter error.

8.20.29 HTML_setStatusBarStatusByID

Description: Sets new percent value and/or new status text by status bar ID.
Parameter:

- id: ID of the status bar
- percent: Percent value to write into the DB (may be false, if it should not be changed).
- statustext: A text message that should be shown under the status bar and written to the DB (may be false, if it should not be changed).

Returns:

- : false on parameter error.

8.20.30 HTML_setStatusBarStatus

Description: Sets new percent value and/or new status text by status bar ID or clientname AND status bar name.
Parameter:

- id: ID of the status bar (Optional parameter to set values of status bar with given ID).
- percent: Percent value to write into the DB (may be false, if it should not be changed).
- statustext: A text message that should be shown under the status bar and written to the DB (may be false, if it should not be changed).
- statusBarName: The name of the status bar.
- client: The name of the client, the status bar belongs to (or other values for identifying the object the status bar belongs to)

Returns:

- : false on parameter error.

8.20.31 HTML_getStatusBarID

Description: Returns the status bar ID of the searched status bar.
Parameter:

- name: The name of the status bar.
- client: The name of the client, the status bar belongs to (or other values for identifying the object the status bar belongs to)

Returns:

- : The status bar ID of the searched status bar or false if none could be found.

8.20.32 HTML_newStatusBar

Description: Shows the iframe for a status bar. This actually displays the status bar.
Parameter:

- name: The name of the status bar.
- client: The name of the client, the status bar belongs to (or other values for identifying the object the status bar belongs to)
- type: The method of calculating/getting the percentage to display in the status bar.
- cmd: BASH command, if type is STATUSBAR_TYPE_bash.
- refreshtime: Time (in seconds) between refreshes of the status bar.
- statustext: A text message that should be shown under the status bar.
- percent: Percent value to write into the DB.

Returns:

- : The status bar ID of the just created status bar or false, if it could not be created.

8.20.33 HTML_showStatusBar

Description: Shows the iframe for a status bar. This actually displays the status bar.
Parameter:

- id: ID of the status bar.
- width: The width of the status bar iframe.
- height: The height of the status bar iframe.

8.20.34 HTML_showStatusBarHTML

Description: Shows the status bar, that is drawn in the iframe (this function is only called by statusBar.php).
Parameter:

- id: ID of the status bar.

8.20.35 HTML_multiCheckBox

Description: Defines a list of checkboxes, that represent a value each. The values of checked checkboxes are stored in an array and returned.
Parameter:

- htmlName: Name of the HTML element.
- valuesLabels: Array with the values and labels for the checkboxes.
- defaultChecked: Array with values that are checked by default.
- forceReload: Set to true if the check box should be set to the state of $defaultCheck in any case.

Returns:

- : Array with the values of all checked checkboxes.

8.20.36 HTML_multiCheckBoxShow

Description: Shows a list of checkboxes, that represent a value each. The values of checked checkboxes are stored in an array and returned.
Parameter:

- valuesLabels: Array with the values and labels for the checkboxes.
- defaultChecked: Array with values that are checked by default.
- forceReload: Set to true if the check box should be set to the state of $defaultCheck in any case.

Returns:

- : Array with the values of all checked checkboxes.

8.20.37 HTML_setPage

Description: Sets the m23 page as hidden value.
Parameter:

- page: Name of the page.

8.20.38 HTML_storableInput

Description: HTML text or password edit line with loading and storing the values to and from the session.
Parameter:

- htmlName: Name of the HTML element.
- prefKey: Variable name of the preference the dialog element stands for.
- initValue: The initial value if the element is shown first.
- storePointer: Additional pointer to the variable where to store the entered value.
- size: Size (in characters) of the input line.
- maxlength: The maximum length of the entered text.
- type: Type of the edit line (INPUT_TYPE_text or INPUT_TYPE_password)

Returns:

- Returns the entered value, the default value or false.

8.20.39 HTML_storableSelection

Description: Shows a list of radio buttons or a selection with loading and storing the checking state to and from the session.
Parameter:

- htmlName: Name of the HTML element.
- prefKey: Variable name of the preference the dialog element stands for.
- array: An array that hold the returned values (array keys) the naming for the elements (array values).
- type: SELTYPE_selection for a selection or SELTYPE_radio for radio buttons.
- vertical: Set to true if the radio buttons should be aligned vertically or to false for horizontal aligning. This parameter is ignored by selections.
- defaultSelection: The value of the item to select by default.
- storePointer: Additional pointer to the variable where to store the entered value.
- js: Here can JavaScript or other parameters be added.

Returns:

- true if the check box is checked.

8.20.40 HTML_storableCheckBox

Description: Shows a check box with label with loading and storing the checking state to and from the session.
Parameter:

- htmlName: Name of the HTML element.
- label: Label of the element.
- prefKey: Variable name of the preference the dialog element stands for.
- defaultCheck: Set to true if the check box should be checked if no HTML value is given.
- storePointer: Additional pointer to the variable where to store the entered value.

- checkedValue: The value that should be stored into $storePointer if the check box is checked.
- unCheckedValue: The value that should be stored into $storePointer if the check box is NOT checked.

Returns:

- true if the check box is checked.

8.20.41 HTML_getElementValue

Description: Gets the value for a HTML element by the session data or POST value.
Parameter:

- htmlName: Name of the HTML element.
- prefKey: Variable name of the preference the dialog element stands for.
- initValue: The initial value if the element is shown first.

Returns:

- Returns the default value, the session value or false.

8.20.42 HTML_listSelection

Description: shows a selection with options stored in an array
Parameter:

- selName: name of the selection
- list: array with the entries. The array can be a simple numeric array or an associative array with discrete entries for the shown name and the value. e.g. : $l[name0]="public"; $l[val0]="internal"; $l[name1]="public1"; $l[val1]="internal1"; public and public1 will be shown the user in the web-browser, while internal and internal1 are the values that are transfered to the server.
- first: entry that should be shown first (this is the internal value and NOT the name shown to the user). the first value from the list will be written to $first. set first to "false" to disable writing the first entry.
- firstName: if you want to use the associative array variant and a first value, you need to set the name that should be shown to the user. This name is stored in firstName

8.20.43 HTML_showTableHeader

Description: prints the header of a shadowed table
Parameter:

- centerTable: set to true if the table should be centered vertically
- tableStyle: CSS class of the inner table.
- width: Width of the table.

8.20.44 HTML_showTableEnd

Description: prints the end of a shadowed table
Parameter:

- centerTable: set to true if the table should be centered vertically

8.20.45 HTML_showFormHeader

Description: Shows the header of a formular
Parameter:

- addAction: set it, if additional parameters to index.php should be used
- method: default is POST, but you can set it to GET

8.20.46 HTML_showFormEnd

Description: Shows the end of a formular

8.20.47 HTML_submit

Description: Defines a submit button.
Parameter:

- htmlName: Name of the HTML element.
- label: Label of the element.
- extra: Extra options for the HTML input tag.

Returns:

- True if it was clicked otherwise false.

8.20.48 HTML_submitImg

Description: Defines a graphical submit button.
Parameter:

- htmlName: Name of the HTML element.
- img: Name of the image to show.
- alt: Alternative text to show when no images can be shown.
- extra: Extra options for the HTML input tag.

Returns:

- True if it was clicked otherwise false.

8.20.49 HTML_input

Description: HTML text or password edit line.
Parameter:

- htmlName: Name of the HTML element.
- htmlValue: The default text to show in the edit line if nothing was submitted.
- size: Size (in characters) of the input line.
- maxlength: The maximum length of the entered text.
- type: Type of the edit line (INPUT_TYPE_text or INPUT_TYPE_password)
- Returns the entered value, the default value or false.

8.20.50 array_makeFirst

Description: special sort function that makes a special element the first element and leaves the other elements in its previous order.
Parameter:

- arr: Array to sort
- first: Value of the element that should be put on top

8.20.51 HTML_getValidSelected

Description: Checks for a valid selected value from a list of possible values. In case the value could not be found, a default value is taken.
Parameter:

- selected: Array or single value to check, if it is on the list aof array keys.
- arrayKeys: An array that holds the possible returned values (array keys).
- defaultSelection: The value of the item to select by default.

Returns:

- A valid value from a list of possible values.

8.20.52 HTML_selection

Description: Shows a list of radio buttons or a selection.
Parameter:

- htmlName: Name of the HTML element.
- array: An array that hold the returned values (array keys) the naming for the elements (array values).
- type: SELTYPE_selection for a selection or SELTYPE_radio for radio buttons.
- vertical: Set to true if the radio buttons should be aligned vertically or to false for horizontal aligning. This parameter is ignored by selections.
- defaultSelection: The value of the item to select by default.
- prefKey: Variable name of the preference the dialog element stands for.
- js: Here can JavaScript or other parameters be added.
- multipleSize: If set to a number (and not to false) a multi selection is generated, where the user can select multiple entries. The number sets the amount of entries to show the user.

Returns:

- The value of the selected element or false if nothing was selected.

8.20.53 HTML_checkBox

Description: Shows a check box with label.
Parameter:

- htmlName: Name of the HTML element.
- label: Label of the element.
- defaultCheck: Set to true if the check box should be checked if no HTML value is given.

- prefKey: Variable name of the preference the dialog element stands for.
- htmlValue: Value of the checkbox if clicked.
- forceReload: Set to true if the check box should be set to the state of $defaultCheck in any case.

Returns:

- True if the check box is checked.

8.20.54 HTML_checkBoxCheckAll

Description: Generates an array with all checked checkboxes. It assumes that value of a checked checkbox is 1.
Parameter:

- filter: Filter to get only the POST elements which begin with the filter string.

Returns:

- Array with all checked checkboxes.

8.20.55 HTML_submitDefine

Description: Defines but does not show a button.
Parameter:

- htmlName: Name of the HTML element.
- label: Label of the element.
- extra: Extra options for the HTML input tag.

8.20.56 HTML_submitCheck

Description: Checks if a previously defined button was clicked.
Parameter:

- htmlName: Name of the HTML element.

Returns:

- True if the button was clicked.

8.20.57 HTML_showTableRow

Description: Shows a table row with a variable amount of entries. The parameters are shown side by side as rows in a table. If more than one HTML_showTableRow commands are executed in one table it is needed to always use the same amount of paramaters in each call.
Parameter:

- Arbitrary amount of cells to show in a table.

8.20.58 HTML_showTableHeading

Description: Shows a table heading row with a variable amount of entries. The parameters are shown side by side as rows in a table. If more than one HTML_showTableRow commands are executed in one table it is needed to always use the same amount of paramaters in each call.
Parameter:

- Arbitrary amount of cells to show in a table.

8.20.59 HTML_textArea

Description: Shows a text area to insert text.
Parameter:

- htmlName: Name of the HTML element.
- cols: Number of columns.
- rows: Number of rows to show.
- default: Text to show by default.

Returns:

- : The entered text.

8.20.60 HTML_showPagePrintButton

Description: Shows a print button that allows easy printing of the current m23 administration interface.

8.21 ./inc/hwinfo.php

Author: Daniel Kasten (DKasten@pc-kiel.de) ,Hauke Goos-Habermann (HHabermann@pc-kiel.de)
Description: functions to view the hardware information of the client.

8.21.1 HWINFO_getParam

Description: get hardware infos
Parameter:

- paramName: parameter to read from the hardware info (e.g. cpu, mem, ...)
- clientName: name of the client

8.21.2 HWINFO_getMemory

Description: returns the size of memory
Parameter:

- clientName: name of the client

8.21.3 HWINFO_getHDSize

Description: Returnes the sizes of all harddisks in a string, sperated by html breaks
Parameter:

- clientName: name of the client

8.21.4 HWINFO_printPartitions

Description: prints the partition information
Parameter:

- clientName: name of the client

8.21.5 DMI_getAllTextBox

Description: Get all DMI info in a text box.
Parameter:

- clientName: name of the client

Returns:

- All DMI info in a text box.

8.21.6 DMI_getParam

Description: get dmi info for a special parameter
Parameter:

- paramName: name of dmi setting
- clientName: name of the client

8.21.7 DMI_getBoard

Description: get the dmi board informations
Parameter:

- clientName: name of the client

8.21.8 DMI_getMemory

Description: get the dmi memory informations
Parameter:

- clientName: name of the client

8.21.9 DMI_getCPU

Description: get the dmi cpu informations
Parameter:

- clientName: name of the client

8.21.10 DMI_getSlot

Description: get the dmi information about slots
Parameter:

- clientName: name of the client

8.22 ./inc/i18n.php

Author: Hauke Goos-Habermann (HHabermann@pc-kiel.de)
Description: functions for multi language handling in m23.

8.22.1 I18N_number_format

Description: Converts numbers to the language specific number formating.
Parameter:

- in: Input number.

Returns:

- Language specific number formated number.

8.22.2 I18N_convertToHumanReadableName

Description: Converts a short language into a human readable name.
Parameter:

- lang: Two letter TLD (or longer code for countries that have more than a language (e.g. be-nl, be-fr))

Returns:

- Language in human readable notation.

8.22.3 I18N_m23instLanguage

Description: Checks if a m23inst.php exists for the given language and returns "en" if not.
Parameter:

- shortLanguage: Two letter TLD (or longer code for countries that have more than a language (e.g. be-nl, be-fr))

Returns:

- Language code with available m23inst.php file.

8.22.4 I18N_addLanguage

Description: Adds a new language to the i18n table.
Parameter:

- webinterface: Set to true, if it is a webinterface language. Set to false, for marking a client language.
- shortLanguage: Two letter TLD (or longer code for countries that have more than a language (e.g. be-nl, be-fr))
- longLanguage: Long human readable country/language name.
- country: Two letter TLD.
- lang: Locale setting (e.g. for locale and KDM)
- keymap: Available keymaps for the console etc.
- xkeyboard: X11 keybord setting.
- kdekeyboard: Language setting for the KDE keyboard.
- locale: List of locales (seperated by newlines) for the locale tool.

- kdekeyboards: List of additional KDE keyboards (seperated by commata).
- timezone: The timezone matching the language.
- packagelang: Language suffix that is added to the packages (e.g. language-pack-gnome-XX)

8.22.5 I18N_listClientLanguages

Description: Lists all languages, the m23 clients can be installed with.
Parameter:

- default: the language that should be shown first
- directOutputtedSelection: Set to true, if the selection should be shown instead of returned.

8.22.6 I18N_countCachedLanguages

Description: Counts the cached languages for a type.
Parameter:

- webinterface: Set to true, if it is a webinterface language. Set to false, for marking a client language.

8.22.7 I18N_cacheWebinterfaceLanguages

Description: Caches the language information from the language.info files to the DB.

8.22.8 I18N_getAllCachedLanguages

Description: Returns an associative array with the shortnames of the language as keys and the longnames as value.
Parameter:

- webinterface: Set to true, if it is a webinterface language. Set to false, for marking a client language.

8.22.9 I18N_listWebinterfaceLanguages

Description: Lists all languages, the m23 webinterface is available in, as option lines
Parameter:

- default: the language that should be shown first
- directOutputtedSelection: Set to true, if the selection should be shown instead of returned.

8.22.10 I18N_addClientLanguageToCache

Description: Adds a new client language to the i18n DB cache.
Parameter:

- shortLanguage: Two letter TLD (or longer code for countries that have more than a language (e.g. be-nl, be-fr))
- longLanguage: Long human readable country/language name.
- in: Associative array with the information for the language.

8.22.11 I18N_cacheClientLanguages

Description: Caches the client languages in the DB.

8.22.12 I18N_getLangVars

Description: Returns an associative array with language settings for the client.
Parameter:

- shortLanguage: Two letter TLD (or longer code for countries that have more than a language (e.g. be-nl, be-fr))

8.23 ./inc/ldap.php

Author: Hauke Goos-Habermann (HHabermann@pc-kiel.de)
Description: Functions for using a LDAP server

8.23.1 LDAP_getTypes

Description: Returns an array with the known LDAP types.
Returns:

- Aarray with the known LDAP types.

8.23.2 LDAP_connectServer

Description: Connects with read/write access to the LDAP server on the m23 server.
Parameter:

- name: name of the LDAP server stored in the configuration file

8.23.3 LDAP_makeConnection

Description: Connects to a LDAP server.
Parameter:

- host: hostname or IP of the LDAP server
- baseDN: baseDN for the LDAP server
- pwd: password for the administrator or empty for anonymous access

8.23.4 LDAP_listServers

Description: Returns an associative array with the LDAP server names as keys and values.
Returns:

- Associative array with the LDAP server names as keys and values.

8.23.5 LDAP_loadServer

Description: Loads the variables from a LDAP server.
Parameter:

- name: server name

8.23.6 LDAP_addPosix

Description: Adds a posix account to the LDAP server and encrypts the password with MD5.
Parameter:

- ldapServer: name of the LDAP server stored in the configuration file
- account: the login name
- forename: the forename of the user
- familyname: the familyname of the user
- pwd: the unencrypted password
- uid: Linux user ID
- gid: Linux group ID

8.23.7 LDAP_fqdn2dn

Description: Returns the DN converted from a FQDN
Parameter:

- domain: a full qualified domain name (e.g. test.m23.de)

8.23.8 LDAP_installServer

Description: Generates a script that installs and configures an openLDAP server
Parameter:

- host: the IP or hostname of the LDAP server
- org: name of the organisation
- domain: the DN (e.g. foo.m23.de)
- pwd: the unencrypted password for the admin

8.23.9 LDAP_addServerTophpLdapAdmin

Description: Adds an LDAP server to the phpLDAPadmin configuration file.
Parameter:

- name: how the LDAP server should be called
- host: the IP or hostname of the LDAP server
- base: the base DN (e.g. dc=m23, dc=de)
- pwd: the unencrypted password for the admin

8.23.10 LDAP_delServerFromphpLdapAdmin

Description: Deletes a LDAP server from the phpLDAPadmin configuration file.
Parameter:

- name: the name of the LDAP server that should be deleted

8.23.11 LDAP_checkphpLdapAdminConfiguration

Description: Checks if the phpLDAPadmin configuration file is existing and creates it if it's missing

8.23.12 LDAP_showServerManagementDialog

Description: Shows a dialog for adding, removing and changing LDAP servers.

8.23.13 LDAP_I18NLdapType

Description: Returns the human readable description of the LDAP usage type
Parameter:

- status: status string

8.23.14 LDAP_getNextID

Description: Returns the next higher than the highest ID
Parameter:

- type: "user" for user IDs

8.23.15 LDAP_getNextUserID

Description: Returns the next higher than the highest user ID

8.23.16 LDAP_addNewID

Description: Adds a new ID to the table of used IDs or returns "false" if the ID exists
Parameter:

- type: "user" for user IDs
- id: number of the new ID

8.23.17 LDAP_deleteID

Description: Delets an ID from the table of used IDs
Parameter:

- type: "user" for user IDs
- id: number of the new ID

8.23.18 LDAP_addNewUserID

Description: Adds a new user ID to the table
Parameter:

- id: number of the new ID

8.23.19 LDAP_addNewGroupID

Description: Adds a new user ID to the table
Parameter:

- id: number of the new ID

8.23.20 LDAP_getNextGroupID

Description: Returns the next higher than the highest group ID

8.23.21 LDAP_getFreeIDs

Description: Returns an array with free IDs of the selected type.
Parameter:

- type: "user" for user IDs
- start: start ID to check if it's free
- amount: the amount of IDs to return

8.23.22 LDAP_getFreeUserIDs

Description: Returns an array with free user IDs of the selected type.
Parameter:

- start: start ID to check if it's free
- amount: the amount of IDs to return

8.23.23 LDAP_getFreeGroupIDs

Description: Returns an array with free group IDs of the selected type.
Parameter:

- start: start ID to check if it's free
- amount: the amount of IDs to return

8.23.24 LDAP_matchLDAPserver

Description: Searches for the name of a LDAP server and returns the name of the found server or false
Parameter:

- host: the IP or hostname of the LDAP server
- base: the base DN (e.g. dc=m23, dc=de)

8.24 ./inc/m23shared/m23shared.php

Author: Hauke Goos-Habermann (HHabermann@goos-habermann.de)
Description: database functions, open, close the database, get ip of the calling client

8.24.1 m23SHARED_deleteBills

Description: Deletes all bills.

8.24.2 m23SHARED_deleteUserdataFromDB

Description: Deletes user data not needed for archiving purpose.

8.24.3 m23SHARED_isMarkedForDeletion

Description: Checks if the current user is marked for deletion.
Returns:

- True, if the user should be deleted on the next call or false if not.

8.24.4 m23SHARED_markForDeletion

Description: Marks the current user's data for deletion and disable his login.

8.24.5 m23SHARED_DBname

Description: Generates a database name for m23 shared by input and salt.
Parameter:

- adminName: Name of the m23shared admin

Returns:

- Database name.

8.24.6 m23SHARED_getDBnameByInterface

Description: Calculates the database name for m23 shared by setting in the user interface
Returns:

- Database name or false if there is no user logged in.

8.24.7 m23SHARED_getDBnameByClient

Description: Gets the database name for m23 shared as part of the client ID.
Returns:

- Database name or false if there is no DB name in the ID.

8.24.8 m23SHARED_init

Description: Checks if a m23shared environment is present, sets the variables in the session and chooses the according DB.
Returns:

- True if an m23 m23shared environment is present, false on error or no m23shared.

8.24.9 m23SHARED_new

Description: Adds a new m23shared user and creates a new DB.
Parameter:

- user: The name of the new user.
- password: The according password.

Returns:

- True if the creation was sucessfully, otherwise false.

8.24.10 m23SHARED_setLicense

Description: Sets the maximum allowed amount of clients and the evaluation time in days.
Parameter:

- payType: Payment type (M23SHARED_PAYTYPE_EVAL, M23SHARED_PAYTYPE_PROFESSIONAL)
- clientAmount: Amount of clients that can be manages with the license.
- evaluationDays: Days for evaluation time.

8.24.11 m23SHARED_evaluationDaysLeft

Description: Calculates how many days are left from evaluation period.
Returns:

- : Amount of days left (can be negative too if the evaluation time is over)

8.24.12 m23SHARED_evaluationEndDate

Description: Generates a string with the end date and time in current selected language of the evaluation period.
Returns:

- : String with the end date and time in current selected language of the evaluation period.

8.24.13 m23SHARED_paidClients

Description: Calculates and returns the amount of clients the customer has paid for.
Returns:

- : Amount of clients the customer has paid for.

8.24.14 m23SHARED_getCompleteClientName

Description: Returns the complete name of a m23 shared client ($_SESSION variables need to be set).
Parameter:

- : clientName: Name of the client (if the value from the session should not/cannot be taken)

Returns:

- : Client name of a m23 shared client (e.g. m23SrGH1ikdPZ8.test)

8.24.15 m23SHARED_getServerIP

Description: Returns the FQDN of the m23shared server.
Returns:

- : FQDN of the m23shared server.

8.24.16 m23SHARED_getPayTypeArray

Description: Returns an array with the nummeric pay types as index and the human readable translations as values.
Parameter:

- withoutEval: If set to true the evaluation entry will not be added.

Returns:

- : Array with assignment of nummeric pay types and human readable translations.

8.24.17 m23SHARED_getSalutationsArray

Description: Returns an array with salutation forms.
Returns:

- : Array with gender character as key and localised salutation.

8.24.18 m23SHARED_getSalutationHeadline

Description: Returns a saluation headline that can be used as beginning of an email.
Returns:

- : Salutation headline in current language.

8.24.19 m23SHARED_getPayTypeHumanReadable

Description: Returns the license and payment type of the m23@web account as human readable string.
Returns:

- : Human readable string with license and payment type of the m23@web account.

8.24.20 m23SHARED_getMonthlyFee

Description: Calculates the monthly fee with the given amount af paid clients.
Returns:

- : Monthly price.

8.24.21 m23SHARED_getCustomerEmail

Description: Returns the eMail address of customer.
Returns:

- : eMail address of customer.

8.24.22 m23SHARED_setCustomerEmail

Description: Sets the eMail address of customer.

8.24.23 m23SHARED_generateActivationKey

Description: Calculates a random activation key and stores it into the DB.
Returns:

- : Random activation key.

8.24.24 m23SHARED_getActivationKey

Description: Returns the activation key for this customer.
Returns:

- : Activation key.

8.24.25 m23SHARED_generateCustomerNumber

Description: Calculates the customer number by current time and random value and stores it into the DB.
Returns:

- : Customer number.

8.24.26 m23SHARED_getCustomerNr

Description: Returns the customer number for this customer.
Returns:

- : Customer number.

8.24.27 m23SHARED_activate

Description: Activates a customer account identified by user and activation code, makes some checks if the data is correct and sends a welcome email.
Parameter:

- user: The user name of the customer.
- code: The activation code for the account.

8.24.28 m23SHARED_setCustomerLanguage

Description: Sets the language for this customer.
Parameter:

- lang: Language for the customer.

8.24.29 m23SHARED_getCustomerLanguage

Description: Returns the language setting for this customer.
Returns:

- : Language for this customer.

8.24.30 m23SHARED_sendActivationMail

Description: Prepares and sends the activation mail.
Parameter:

- eMail: eMail address of the new customer.
- username: Username of the new customer.
- lang: Language that should be used in the eMail.

8.24.31 m23SHARED_sendActivationMail

Description: Prepares and sends the welcome mail.
Parameter:

- eMail: eMail address of the new customer.
- username: Username of the new customer.
- lang: Language that should be used in the eMail.

8.24.32 m23SHARED_setRealName

Description: Sets the real name of the customer.
Parameter:

- name: Real name of the customer.

8.24.33 m23SHARED_getRealName

Description: Returns the real name of the customer.
Returns:

- : Real name of the customer.

8.24.34 m23SHARED_changeClientAmount

Description: Changes the amount of paid clients for the current customer.
Parameter:

- newClientAmount: The amount of clients, the customer wants to pay for.

8.24.35 m23SHARED_showBill

Description: Shows the bill for a selected month.
Parameter:

- month: The month the bill should be generated.
- year: The year of the bill.

8.24.36 m23SHARED_billDateSelection

Description: Creates an HTML selection with valid billing months.
Parameter:

- htmlName: Name of the HTML selection.

Returns:

- The selected month and year.

8.24.37 m23SHARED_priceFormater

Description: Formats a price with two digit decimal place.
Parameter:

- price: The price to format.

Returns:

- The formatted price.

8.24.38 m23SHARED_calculateBill

Description: Calculates the bill for a choosen month in a selected year.
Parameter:

- month: Month to get the bill for.
- year: Year to get the bill for.

Returns:

- Associative array with the bill data.

8.24.39 m23SHARED_getLicenseType

Description: Returns the m23shared license of the current m23shared user.
Returns:

- License type as constant.

8.24.40 m23SHARED_sendAdminMail

Description: Sends an GPG encrypted eMail to the admin.
Parameter:

- subject: The subject of the eMail.
- text: The eMail message.

8.24.41 m23SHARED_getCurrentUser

Description: Returns the current m23shared user.
Returns:

- Current m23shared user.

8.24.42 m23SHARED_changePasswordDialog

Description: Tries to change the password for the current m23shared user and shows an error or sucess message. Both of the entered passwords must be identically.
Parameter:

- pwd1: The password.
- pwd2: The retyped password.

8.24.43 m23SHARED_changePasswordDialog

Description: Tries to change the eMail for the current m23shared user and shows an error or sucess message.
Parameter:

- email: New eMail address.

8.24.44 m23SHARED_getBillDates

Description: Returns an array filled with all month and years where bills are present.
Returns:

- Associative array with all month and years where bills are present as array name and value.

8.24.45 m23SHARED_downloadBillPDFLink

Description: Generates a HTML link that points to the script that generates the bill for a given monath and year.
Parameter:

- month: Month to get the bill for.
- year: Year to get the bill for.

Returns:

- : HTML link to the bill PDF.

8.24.46 m23SHARED_getCustomerAddress

Description: Returns HTML formatted address information for the current customer.
Returns:

- Address information for the current customer.

8.24.47 m23SHARED_getCustomerBankHTML

Description: Returns HTML formatted bank account information for the current customer.
Returns:

- Bank account information for the current customer.

8.24.48 m23SHARED_pdfBill

Description: Generates a bill in PDF format for a choosen month in a selected year.
Parameter:

- month: Month to get the bill for.
- year: Year to get the bill for.

8.24.49 m23SHARED_setBankAccount

Description: Sets bank account information for the current customer.
Parameter:

- bankAccountHolder: Name of the account holder.
- bank: Name of the bank.
- accountNumber: The number of the bank account.
- bankCode: The bank code number.

8.24.50 m23SHARED_setAddress

Description: Sets address information for the current customer.
Parameter:

- resident: The name of the resident or his company.
- postCode: Postcode.
- city: City.
- street: Street.
- houseNumber: House number.

8.24.51 m23SHARED_unusedPaidClientsAvailable

Description: Checks if there are unused paid clients and shows an error message if not.
Returns:

- : true if there are unused clients, false otherwise.

8.24.52 m23SHARED_showLicenseDialog

Description: Shows a dialog for viewing and changing the license and paid client amount.

8.24.53 m23SHARED_getAllm23sharedUsers

Description: Gets all m23shared users.
Parameter:

- onlyWithBillTable: If set to true, only m23shared users with existing bill table will be returned.

Returns:

- : Array with all m23 shared users.

8.24.54 m23SHARED_switchUser

Description: Changes the current m23shared user.

8.24.55 m23SHARED_prepareBillMailSending

Description: Prepares the bill mail sending queue.
Parameter:

- month: Month to mail the bill for.
- year: Year to mail the bill for.

Returns:

- Array with all users that have bill mails to send.

8.24.56 m23SHARED_markBillMailAsSent

Description: Marks a bill mail as sent.
Parameter:

- month: Month to mail the bill for.
- year: Year to mail the bill for.
- user: DB name of the customer.

8.24.57 m23SHARED_sendAllBillMails

Description: Sends the bills of all users from the previous month as PDF attachement. This should be run at the beginning of a month.

8.24.58 m23SHARED_addExtraBill

Description: Adds an extra entry to the bill.
Parameter:

- amount: Amount of good.
- description: Description of the good.
- unitprice: Price per unit.

8.24.59 m23SHARED_showDonationDialog

Description: Shows a dialog where the user can donate to the m23 project.

8.24.60 m23SHARED_showBootMediaDownloadDialog

Description: Shows a dialog with download icons for the different boot media.

8.24.61 m23SHARED_allUserDBQuery

Description: Executes an SQL query on all m23shared databases and returns an associated array with all results.
Parameter:

- sqlIN: The input query string that MUST include m23SHAREDDB as placeholder for the current m23shared DB name.

Returns:

- Associated array with the query result of all m23shared DBs.

8.24.62 m23SHARED_getUserByResident

Description: Gets the user name of m23shared customer by the resident name.
Parameter:

- resident: Resident name of the customer.

Returns:

- m23shared user name for the searched customer or empty string if none was found.

8.24.63 m23SHARED_getUserByCustomerNr

Description: Gets the user name of m23shared customer by the customer number.
Parameter:

- customerNr: Customer number of the m23shared customer.

Returns:

- m23shared user name for the searched customer or empty string if none was found.

8.24.64 m23SHARED_getUserByInfo

Description: Gets the user name of m23shared customer by searching all m23shared DBs for var and value in the remotevar table.
Parameter:

- remotevarVar: Variable name to search in the remotevar table.
- remotevarValue: Value to search in the remotevar table.

Returns:

- m23shared user name for the searched customer or empty string if none was found.

8.24.65 m23SHARED_showAdminDialog

Description: Shows a dialog for the admin to search users for and change values.

8.24.66 m23SHARED_blockAccount

Description: Blocks or unblocks an account.
Parameter:

- block: Set to true blocks, false unblocks.

8.24.67 m23SHARED_isAccountBlocked

Description: Checks if an account is blocked.
Returns:

- : True if the account is blocked, false if not.

8.24.68 m23SHARED_showAddExtraBillDialog

Description: Shows a dialog for adding extra entries to the bill.

8.24.69 m23SHARED_showBillDialog

Description: Shows a dialog for viewing and choosing the bill.

8.24.70 m23SHARED_showPriceListTable

Description: Shows a table with the price list.

8.24.71 m23SHARED_getInformationForBootingYourClientLink

Description: Generates a link to the help page information for booting the client.
Parameter:

- : client: Name of the client (if the value from the session should not/cannot be taken)

Returns:

- Link to the help page information for booting the client.

8.25 ./inc/massTools.php

Author: Hauke Goos-Habermann (HHabermann@pc-kiel.de)
Description: routines for mass installations

8.25.1 MASS_EGKradioBoxes

Description: Generates HTML code for showing 3 elements, that can be each a "radio button", selection "disabled" or "always selected".
Parameter:

- RB_name: name of the radio button
- arr: array with 3 values for [Enter,Generate,Keep]. Setting a value to "e" means that the user can select, "n" selection is disabled, "y" is always select.
- checkNr: the number of radio button that is enabled by default.

8.25.2 MASS_FHradioBoxes

Description: Generates HTML code for showing 2 radio buttons for selecting file or handy source
Parameter:

- RB_name: name of the radio button
- checkNr: the number of radio button that is enabled by default.

8.25.3 MASS_showFileHandDialog

Description: shows a dialog for selecting "by file" or "by hand" for the "enter" properties.
Parameter:

- EGKparams: enter generate keep parameters, that hold information about handling of the properties

8.25.4 MASS_propertyKeys

Description: returns the keys for all properties

8.25.5 MASS_showFileFormatDialog

Description: shows a dialog that lets the user select a DB file and assign the columns to the fields of the file
Parameter:

- EGKparams: enter generate keep parameters, that hold information about handling of the properties

8.25.6 MASS_keyToI18N

Description: converts the property names to I18N names
Parameter:

- key: property name

8.25.7 MASS_I18NTokey

Description: converts the I18N names to property names
Parameter:

- key: property name

8.25.8 MASS_showTableDefinition

Description: shows a dialog that lets the user define which field in the DB file should be assigned to which property
Parameter:

- EGKparams: enter generate keep parameters, that hold information about handling of the properties
- DBfileName: file name of the DB file

8.25.9 MASS_checkAndSaveFields

Description: saved the assignments from field number to property and other information to EGKparams and performes a simple check, to verify that the values of the properties are valuable. An error message is returned or an empty string, if all is ok.
Parameter:

- EGKparams: enter generate keep parameters, that hold information about handling of the properties

8.25.10 MASS_openDBFile

Description: opens a DB file
Parameter:

- fileName: name of the DB file

8.25.11 MASS_readDBFile

Description: reads a line from the DB file and returnes an associated array with the properties as key and the fields of the file as values.
Parameter:

- fileName: name of the DB file

Returns:

- Associative array with the values of the DB line or false, if the line was empty.

8.25.12 MASS_readDBFileRaw

Description: reads a line from the DB file and returnes the fields splitted to a normal array.
Parameter:

- file: file pointer
- glue: the seperator used to seperate the fields

8.25.13 MASS_closeDBFile

Description: closes the DB file.
Parameter:

- file: file pointer

8.25.14 MASS_getXProperties

Description: returnes the amount and keys of a secial kind (enter, generate, keep, hand, file)
Parameter:

- EGKparams: enter generate keep parameters, that hold information about handling of the properties
- x: the 1-letter code of enter, generate, keep, hand or file
- pre: set if there is a prefix before the key name

8.25.15 MASS_showGeneratorOptions

Description: shows the dialog for configuring the generator options
Parameter:

- EGKparams: enter generate keep parameters, that hold information about handling of the properties

8.25.16 MASS_passGenerator

Description: generates the selected amount of passwords with a random algorithm or the pwgen tool.
Parameter:

- length: length of the passwords to generate
- method: random or pwgen generated passwords that can be memorized by humans easily
- amount: the amount of passwords to generate

Returns:

- Array with the generated passwords as keys.

8.25.17 MASS_loginGenerator

Description: generates the selected amount of logins
Parameter:

- base: the base name of the login
- start: start number for incremental logins
- forenames: array with all forenames
- familynames: array with all familynames
- type: "incremental" if you want to add a incrementing number after the base name, "ForeFamilyName" if the logins should be created from fore- and familynames
- amount: the amount of logins to generate

8.25.18 MASS_ipGenerator

Description: generates the selected amount of IPs in the selected ranges. Only IPs are generated that aren't in use by m23 or (if activated) pingable.
Parameter:

- amount: the amount of IPs to generate
- rangeStr: string with IP range information
- ping: set to true, if each IP should be pinged before it becomes valid

8.25.19 MASS_minMaxIP

Description: calculates the possible minimum and maximum IP of a given netmask. The IPs are returned as an array: index 0 = minimum; index 1 = maximum.
Parameter:

- netmask: netmask to use
- ip: is used if the can only be set the current part of the ip (max and min ip part == 255)

8.25.20 MASS_generateNetmask

Description: generate netmasks from ip addresses via network class definitions.
Parameter:

- ip: the ip that shoulb be used to calculate the netmask

8.25.21 MASS_generateClientNames

Description: generates client names through appending of numbers.
Parameter:

- base: the client base name
- start: the start number
- amount: the amount of client names to generate

8.25.22 MASS_saveGeneratorOptions

Description: saves all geneator options to EGKparams
Parameter:

- EGKparams: enter generate keep parameters, that hold information about handling of the properties

8.25.23 MASS_showOverview

Description: shows a table with all generated client settings, that can be edited
Parameter:

- EGKparams: enter generate keep parameters, that hold information about handling of the properties

8.25.24 MASS_getAllFromFile

Description: returnes all values from one key of the DB file as an array.
Parameter:

- key: the key of the property
- generateAmount: the amount of values to be read from the DB file
- EGKparams: enter generate keep parameters, that hold information about handling of the properties
- fromDBFile: 2D array filled with the values for the keys

8.25.25 MASS_getLongestLength

Description: returnes the length of the longest entry in the array or max if bigger than max
Parameter:

- arr: the array
- amount: the amount of entries in the array
- max: maximal value to be returned

8.25.26 MASS_startInstall

Description: starts the installation of all client with all paramaters defined in the table
Parameter:

- EGKparams: enter generate keep parameters, that hold information about handling of the properties

8.26 ./inc/menu.php

Author: Hauke Goos-Habermann (HHabermann@pc-kiel.de)
Description: functions for menus

8.26.1 MENU_showEntry

Description: generates a menu entry, highlights it (if selected) and removes menu entry formatting tags from the menu entry label
Parameter:

- text: label of the menu entry
- link: link to the page
- icon: directory and name of the icon to show in front of the entry

8.26.2 MENU_startGroup

Description: shows the start of a menu group
Parameter:

- name: name of the menu group

8.26.3 MENU_endGroup

Description: shows the end of a menu group

8.27 ./inc/message.php

Author: Hauke Goos-Habermann (HHabermann@pc-kiel.de)
Description: function to show an info box

8.27.1 MSG_showMessageBoxPlaceholder

Description: Shows a placeholder for message boxes at the position of execution.

8.27.2 MSG_placeOrReturnMessageBox

Description: Replaces a (maybe) existing message box placeholder with the given text.
Parameter:

- text: Message to embed in the placeholder.

Returns:

- Empty string, if the text could be inserted into the existing placeholder or the message, if no placeholder was found.

8.27.3 MSG_getm23UpdateFeed

Description: Shows the m23 server update feed.
Parameter:

- width: Width of the box
- refreshTime: The time in minutes the file is downloaded again.

8.27.4 MSG_getm23DevelopmentBlog

Description: Shows the m23 development blog.
Parameter:

- width: Width of the box
- refreshTime: The time in minutes the file is downloaded again.

8.27.5 MSG_getRSSFeed

Description: Shows a RSS feed.
Parameter:

- url: The URL pointing to the RSS XML file.
- tableType: Name of the CSS table type
- width: Width of the box
- storeFile: The file name to store the download in.
- refreshTime: The time in minutes the file is downloaded again.

8.27.6 MSG_showInfo

Description: Shows the help block for the online help.
Parameter:

- message: the text for the info message
- language: two character language description (e.g. de, en, fr,...)
- width: width of the box

8.27.7 MSG_showError

Description: Shows the error block for the error messages.
Parameter:

- message: the text for the info message
- language: two character language description (e.g. de, en, fr,...)
- width: width of the box

8.27.8 MSG_showWarning

Description: Shows the warning block for the warning messages.
Parameter:

- message: the text for the info message
- language: two character language description (e.g. de, en, fr,...)
- width: width of the box

8.27.9 MSG_show

Description: Shows the message block for the messages.
Parameter:

- message: the text for the info message
- language: two character language description (e.g. de, en, fr,...)
- width: width of the box
- urgency: type of message (e.g. errortable, warningtable, infotable)

8.27.10 MSG_showMessageBoxHeader

Description: shows the header of the message block for the online help
Parameter:

- tableType: name of the CSS table type
- type: the heading of the box
- width: width of the box
- dontShowButReturn: Set to true if the HTML output should be returned rather than show.

8.27.11 MSG_showMessageBoxFooter

Description: shows the footer of the message block for the online help
Parameter:

- dontShowButReturn: Set to true if the HTML output should be returned rather than show.

8.27.12 MSG_showMessageBox

Description: shows the message block for the online help
Parameter:

- message: the text for the info message
- tableType: name of the CSS table type
- type: the heading of the box
- width: width of the box
- dontShowButReturn: Set to true if the HTML output should be returned rather than show.

8.27.13 MSG_showUpdateInfo

Description: shows a info message about the stored update jobs
Parameter:

- unr: the amount of update jobs and clients
- language: two character language description (e.g. de, en, fr,...)

8.27.14 MSG_showUpdateInfo

Description: shows a info message about stored jobs on N clients
Parameter:

- jobNr: the amount of jobs
- clNr: the amount of clients
- language: two character language description (e.g. de, en, fr,...)

8.27.15 MSG_showNewFeature

Description: shows the new feature block
Parameter:

- url: the url to the forum for the new feature
- language: two character language description (e.g. de, en, fr,...)
- width: width of the box

8.27.16 MSG_DeActivateBlogDialog

Description: Creates a dialog to en/disable a blog. The displaying state is written to the DB.
Parameter:

- blogVarName: Variable name of the blog to store in the DB.
- css: Name of the CSS class to color the dialog.
- blogName: Name of the blog als human readle heading.
- width: Width of the box containing the switch dialog.
- dialogCode: The HTML code of the dialog element is written to this variable.

Returns:

- : True if the blog should be shown otherwise false.

8.28 ./inc/packages.php

Author: Hauke Goos-Habermann (HHabermann@pc-kiel.de)
Description: basic package operations (search, add,...)

8.28.1 PKG_addHSUser

Description: Adds a job for creating an user on a halfSister client.
Parameter:

- client: Name of the client.
- login: Login name of the new user.
- firstpw: Password for the new user.
- uid: Optional user ID of the new user.
- gid: Optional group ID of the new user.

8.28.2 PKG_addUbuntuUser

Description: Adds a job for creating an user on a Ubuntu client.
Parameter:

- client: Name of the client.
- login: Login name of the new user.
- firstpw: Password for the new user.
- uid: Optional user ID of the new user.
- gid: Optional group ID of the new user.

8.28.3 PKG_addDebianUser

Description: Adds a job for creating an user on a Debian client.
Parameter:

- client: Name of the client.
- login: Login name of the new user.
- firstpw: Password for the new user.
- uid: Optional user ID of the new user.
- gid: Optional group ID of the new user.

8.28.4 PKG_addUser

Description: Adds a job for creating an user on the client.
Parameter:

- client: Name of the client.
- login: Login name of the new user.
- firstpw: Password for the new user.
- groups: Array of groups the user should be added.
- uid: Optional user ID of the new user.
- gid: Optional group ID of the new user.

8.28.5 PKG_cleanPackageLine

Description: Removes unwanted characters from a line containing package names and makes sure that there is only one line without line breaks.
Parameter:

- packageLine: Space seperated line containing the package names. The changed line will be written to the parameter too.

8.28.6 PKG_combinem23normal

Description: Combines the package names of multiple entries for m23normal and m23normalRemove jobs in a package selection.
Parameter:

- packageSelectionName: Name of the package selection to optimise.

8.28.7 PKG_importSelectedPackagesFromFile

Description: Imports space-seperated packages from a file and adds them to the wait4acc/selected packages of a client.
Parameter:

- client: Name of the client or empty.
- file: Name of the file with full path containing space-seperated packages.

8.28.8 PKG_exportSelectedPackages

Description: Exports the wait4acc/selected packages of a client.
Parameter:

- client: Name of the client or empty.

8.28.9 PKG_getDebootstrapCacheFilename

Description: Returns the file name of the debootstrap cache file (without path).
Parameter:

- release: Select the Debian/Ubuntu suite (squeeze, sarge, sid, precise).
- arch: the computer architecture of the client

Returns:

- The file name of the debootstrap cache file (without path).

8.28.10 PKG_getDebootstrapCacheSfURL

Description: Returns the URL to the debootstrap cache file on the SourceForge server.
Parameter:

- release: Select the Debian/Ubuntu suite (squeeze, sarge, sid, precise).
- arch: the computer architecture of the client

Returns:

- The URL to the debootstrap cache file on the SourceForge server.

8.28.11 PKG_baseSysDownloadedCompletelyTom23Server

Description: Checks, if the debootstrap cache file was downloaded completely to the m23 server.
Parameter:

- release: Select the Debian/Ubuntu suite (squeeze, sarge, sid, precise).
- arch: the computer architecture of the client

Returns:

- true, on complete download otherwise false.

8.28.12 PKG_downloadBaseSysTom23Server

Description: Downloads the debootstrap cache file to the m23 server and checks its validity (by signature).
Parameter:

- release: Select the Debian/Ubuntu suite (squeeze, sarge, sid, precise).
- arch: the computer architecture of the client

Returns:

- true, if the download is completed, otherwise false.

8.28.13 PKG_getDebootstrapCacheServerURL

Description: Returns the URL to the debootstrap cache file on the m23 server.
Parameter:

- release: Select the Debian/Ubuntu suite (squeeze, sarge, sid, precise).
- arch: the computer architecture of the client

Returns:

- The URL to the debootstrap cache file on the m23 server.

8.28.14 PKG_getDebootstrapCacheServerFile

Description: Returns the full path to the debootstrap cache file on the m23 server.
Parameter:

- release: Select the Debian/Ubuntu suite (squeeze, sarge, sid, precise).
- arch: the computer architecture of the client

Returns:

- The full path to the debootstrap cache file on the m23 server.

8.28.15 PKG_isReconfiguredWithExtraDistr

Description: Checks, if the distribution is used for configuring a system that was installed via image.
Parameter:

- pkgID: The ID of the base installation package.

Returns:

- True, if the distribution is used for configuring, otherwise false.

8.28.16 PKG_translateClientjobsStatus

Description: Translates the clientjobs status from the DB into a human readable form.
Parameter:

- status: Status code from the DB.

Returns:

- Human readable translation of the clientjobs status.

8.28.17 PKG_isSpecialPackageAvailableForClient

Description: Checks if a special package is available for the client's distribution.
Parameter:

- client: Name of the client.
- package: Name of the special package.

Returns:

- True if the special package is available otherwise false.

8.28.18 PKG_OptionPageHeader2

Description: Starts the option page for debconf settings with all necessary options.
Parameter:

- title: the window title of the OptionPage

8.28.19 PKG_OptionPageTail2

Description: Generates the bottom of the OptionPage for debconf settings.
Parameter:

- elem: The elements (variable name, type (string, boolean, (multi)select), default value, description) of the debconf templates.

8.28.20 PKG_decodeDebconfDescription

Description: Decodes and HTML-formats the description of a debconf template and extracts its title.
Parameter:

- descr: Text of the debconf description.
- title: Variable to write the title to.

Returns:

- : The decoded and HTML-formated description.

8.28.21 PKG_OptionPageRender2

Description: Renderes the layout of an OptionPage for debconf and stored the debconf settings into the DB.
Parameter:

- layout: The elements (variable name, type (string, boolean, (multi)select), default value, description) of the debconf templates.
- client: The name of the client, the debconf settings should be stored for.
- package: Name of the package, the debconf settings should be stored for.

8.28.22 PKG_countSpecialPackages

Description: counts the special packages of a clients matching the package name and status
Parameter:

- clientName: name of the client
- packageName: name of the special package
- status: status of the package

8.28.23 PKG_countWaitingJobs

Description: returns the amount of a special waiting package
Parameter:

- client: name of the client
- package: name of the package you want to know the amount

8.28.24 PKG_closeSearch

Description: closes a search request
Parameter:

- file: file handle

8.28.25 PKG_getNextPackage

Description: fetches the next package
Parameter:

- file: file handle

8.28.26 PKG_listRecommendPackages

Description: lists recommended packages matching the key
Parameter:

- key: search key
- install: set to true, if the selection for installing all packages should be first

8.28.27 PKG_listRecommendSubPackages

Description: returnes subpackages of a package
Parameter:

- cut: cuts the packages by $cut
- params: variable to write package names to

8.28.28 PKG_addRecommendPackages

Description: adds recommeded packages to db
Parameter:

- amount: amount of selected packages
- client: name of client to install packages on
- normalPackageType2: m23normal, m23normalRemove or orig: select if the packages should be (de)installed or use the saved action
- distr: Name of the distribution.

8.28.29 PKG_addPackageSelection

Description: Adds a package selections to the list of packages to install.
Parameter:

- client: name of client to install packages on
- packageSelectionName: Name of the package selection to install.
- normalPackageType2: m23normal, m23normalRemove or orig: select if the packages should be (de)installed or use the saved action
- distr: Name of the distribution.

8.28.30 PKG_addNormalPackagesToWait4Aac

Description: adds a package to waiting 4 accept status
Parameter:

- client: name of client to install packages on
- priority: priority of the package
- params: parameter for installing the package

8.28.31 PKG_addSpecialPackagesToWait4Aac

Description: adds a special package to waiting 4 accepts status
Parameter:

- client: name of client to install packages on
- priority: priority of the package
- params: parameter for installing the package
- distr: Name of the distribution.

8.28.32 PKG_countJobsWithStatus

Description: Counts named jobs on a client that have a special status.
Parameter:

- client: name of the client
- package: name of the package
- status: The status to search for

Returns:

- The amount of packages on the client with the given status.

8.28.33 PKG_getClientjobsStatus

Description: return the status of a job
Parameter:

- client: name of the client
- package: name of the package
- distr: the name of the distribution
- params: parameter for installing the package
- normalPackage: the name of a normal package

8.28.34 PKG_addNormalPackages

Description: adds normal packages to db
Parameter:

- amount: amount of selected packages
- client: name of client to install packages on

8.28.35 PKG_changePrioritySelectedPackages

Description: Changes the priority of selected wait4acc packages.
Parameter:

- amount: amount of selected packages
- client: name of client to install packages on
- newPriority: The new priority to set.

8.28.36 PKG_rmSelectedPackages

Description: removes normal packages from db
Parameter:

- amount: amount of selected packages
- client: name of client to install packages on

8.28.37 PKG_listSelectedpackages

Description: lists the packages with wait4acc status
Parameter:

- client: name of client to install packages on
- distr: the name of the distribution
- release: release of the distribution

8.28.38 PKG_countSelectedpackages

Description: counts the preselected packages
Parameter:

- client: name of client to install packages on

8.28.39 PKG_countJobs

Description: counts all packages of a client with a given status
Parameter:

- client: name of client
- status: status of the packages to be count or empty to count all jobs

8.28.40 PKG_hasOptions

Description: generates a link to the package option page (if it exists)
Parameter:

- package: name of package
- packageID: id for the selected package
- distr: the name of the distribution
- client: Name of the current client.
- release: The release of the client's distribution.

8.28.41 PKG_savePackageselection

Description: saves all selected packages a package selection
Parameter:

- client: name of client to install packages on
- selectionName: name for the package selection
- showMsg: set to true, if a message should be shown
- status: Status of the clientjobs that should be added.

8.28.42 PKG_addPackageToPackageselection

Description: Add packages to selection
Parameter:

- client: name of client to install packages on
- selectionName: name for the package selection
- packageName: name of the normal package
- params: parameter for the package
- normalPackage: the name of a normal package
- installedSize: the size of the package if it is installed
- priority: The priority of the package.

8.28.43 PKG_listSpecialpackages

Description: lists special packages matching a key
Parameter:

- key: search key

8.28.44 PKG_addSpecialPackages

Description: adds normal packages to db
Parameter:

- amount: amount of selected packages
- client: name of client to install packages on

8.28.45 PKG_getSpecialPackagePriority

Description: gets the priority of a special package
Parameter:

- package: name of package

8.28.46 PKG_getSpecialPackageDescription

Description: gets the description of a special package
Parameter:

- package: name of package

8.28.47 PKG_getSpecialPackageInfo

Description: gets informations from special packages
Parameter:

- package: name of package
- infoType: the type of information you want to get
- dist: shortname of the distribution

Returns:

- The information or false, if no information could be got.

8.28.48 PKG_getPackageID

Description: get the id for a wait4acc job
Parameter:

- client: name of the client
- package: name of the package

8.28.49 PKG_rmNormalJob

Description: adds a normal package removal job to the clientjobs table
Parameter:

- client: name of the client
- packageName: name of the package
- priority: The priority of the job.

8.28.50 PKG_addJob

Description: adds a job to the clientjobs table
Parameter:

- client: name of the client
- packageName: name of the package
- priority: priority of the package
- params: parameter for installing the package

8.28.51 PKG_discardNormalJob

Description: discards all normal packages from the clientjobs table, that match the param
Parameter:

- client: name of the client
- packageName: name of the package

8.28.52 PKG_addWait4AccJob

Description: adds a wait 4 accept job to the clientjobs table
Parameter:

- client: name of the client
- packageName: name of the package
- priority: priority of the package
- params: parameter for installing the package

8.28.53 PKG_addStatusJob

Description: adds a job to the clientjobs table
Parameter:

- client: name of the client
- packageName: name of the package
- priority: priority of the package
- params: parameter for installing the package
- status: the status of the package

8.28.54 PKG_acceptJobs

Description: accepts all preselected jobs
Parameter:

- client: name of the client
- showMsg: set to true, if a message about assigned jobs should be shown

8.28.55 PKG_discardJobs

Description: discards all preselected jobs
Parameter:

- client: name of the client

8.28.56 PKG_discardJob

Description: discards a selected job
Parameter:

- client: name of the client
- package: name of package you want to discard

8.28.57 PKG_changeClientPackageAction

Description: changes the status of a allready installed package
Parameter:

- client: name of the client
- package: name of package you want to discard
- action: the action you want the package set to

8.28.58 PKG_setClientPackageWait4Rm

Description: changes the status of a allready installed package to wait 4 delete
Parameter:

- client: name of the client
- package: name of package

8.28.59 PKG_setClientPackageInstalledOK

Description: changes the status of a package to "installed ok"
Parameter:

- client: name of the client
- package: name of package

8.28.60 PKG_addShutdownPackage

Description: adds a shutdown package, but only if the client is NOT running. returns true, if a schutdown package is added
Parameter:

- client: name of the client

8.28.61 PKG_addShutdownOrRebootPackage

Description: Adds a shutdown or a reboot package. No new job is addedm if there is already a waiting shutdown or reboot job. A shutdown package is added if the client can't be pinged and a reboot package if it is reachable via the network.
Parameter:

- client: name of the client

8.28.62 PKG_getAllParams

Description: gets all parameters of the parameters column of a clientjob
Parameter:

- packageID: the ID of the package

8.28.63 PKG_setAllParams

Description: sets all parameters in the parameters column of a clientjob
Parameter:

- packageID: the ID of the package
- params: the parameters as assiciative array

8.28.64 PKG_OptionPageHeader

Description: starts the option page with all necessary options
Parameter:

- title: the window title of the OptionPage

8.28.65 PKG_OptionPageRender

Description: renderes the layout of a OptionPage
Parameter:

- layout: for the definition of the layout array see the development guide

8.28.66 PKG_OptionPageTail

Description: generates the bottom of the OptionPage
Parameter:

- layout: for the definition of the layout array see the development guide

8.28.67 PKG_OptionPageSaveAlsParameters

Description: saves the entered values in the packagejobs params
Parameter:

- layout: for the definition of the layout array see the development guide

8.28.68 PKG_OptionPageGetValue

Description: gets tha value from a variable. The function tries to get the value from the $_GET array, if it doesn't work it falls back to the params array
Parameter:

- variable: the name of the variable you want to get the value from
- params: the parameters as assiciative array

8.28.69 PKG_listParams

Description: lists the parameters from a package in a nice format
Parameter:

- paramStr: the parameters as string (simply read from the packagejobs table)

8.28.70 PKG_getRecommendPackageAllInstalledSize

Description: calculates the whole sum of the installedSizes of all recommend packages of one package selection
Parameter:

- packageSelection: the name of the package selection

8.28.71 PKG_previewInstallationDeinstallation

Description: showes what happens if a client deinstalls/ installs waiting packages and generates a table with title
Parameter:

- clientName: name of the client
- install: set to true, if packages should be installed. if false the packages should be deinstalled

8.28.72 PKG_showPreviewInstallationDeinstallation

Description: showes what happens if a client installs / deinstalls waiting packages and generates a table with title
Parameter:

- clientName: name of the client
- install: set to true, if packages should be installed. if false the packages should be deinstalled

8.28.73 PKG_updateSourcesListAtAllClients

Description: updates the sources.list at all clients using it
Parameter:

- sourcename: name of the sources.list that should be updated

8.28.74 PKG_executeOnClientJobs

Description: Executes a sql statement on all package IDs.
Parameter:

- sql: initial SQL statement e.g. "DELETE FROM ‘clientjobs‘ WHERE "
- packageIDList: the list of IDs of jobs to be deleted

8.28.75 PKG_removeFromJobList

Description: removes all jobs identified by the IDs in packageIDList
Parameter:

- packageIDList: the list of IDs of jobs to be deleted

8.28.76 PKG_changeClientJobsStatus

Description: Sets a new status on all jobs identified by the IDs in packageIDList
Parameter:

- packageIDList: the list of IDs of jobs to be deleted
- status: New status to set

8.28.77 PKG_removeSpecialFromJobList

Description: Removes a special job from the joblist identified by package name and priority.
Parameter:

- clientName: Name of the client
- package: Name of the package.
- priority: Priority of the job.

8.28.78 PKG_previewUpdateSystem

Description: returns the information of an system update request
Parameter:

- clientName: name of the client
- completeUpdate: set it to "true", if it should be a full update (installation and removal of packages) or to "false" for an update of existing packages

8.28.79 PKG_showPreviewUpdateSystem

Description: generates HTML code woth inormation about the update preview
Parameter:

- clientName: name of the client
- completeUpdate: set it to "true", if it should be a full update (installation and removal of packages) or to "false" for an update of existing packages

8.28.80 PKG_rmAllSpecialPackagesByName

Description: deletes all special packages from a client matching the package name
Parameter:

- clientName: name of the client
- packageName: name of the special package

8.28.81 PKG_getClientsWithPackage

Description: Gets all clients that have the specific package installed (or with another status).
Parameter:

- packageName: Name of the package.
- status: The status the package should have.

Returns:

- Array with all clients that have the specific package installed (or with another status).

8.28.82 PKG_getClientsWithWaitingJobs

Description: Gets all clients that have waiting jobs.
Returns:

- Array with all clients that have waiting jobs.

8.28.83 PKG_getClientsByPackages

Description: Gets all clients that have the specific packages (not) installed (or with another given status).
Parameter:

- packageNames: Array with the packages to check.
- status: Debian status code or true for "installed".
- and: Set to true, if all packages must (not) match the status or, if false, at least one package must match the status.
- not: If set to true, only clients, that have no packes with the given status will be added to the output array.

Returns:

- Array with all clients that have the specific package (not) installed (or with another given status).

8.28.84 PKG_countPackages

Description: counts all packages on a client
Parameter:

- clientName: name of the client

8.28.85 PKG_copyWait4accPackagesToClient

Description: copies the waiting jobs from one client to another
Parameter:

- from: the source client
- to: the destination client

8.28.86 PKG_copyPackagesToClient

Description: copies all with a selected status jobs from one client to another
Parameter:

- from: the source client
- to: the destination client
- status: can be set to a package status or be empty to copy all jobs

8.28.87 PKG_remNormalPackages

Description: adds normal deinstallation jobs to db
Parameter:

- amount: amount of selected packages
- client: name of client to deinstall packages on

8.28.88 PKG_addRemovePackagesToWait4Aac

Description: adds a remove job to waiting 4 accept status
Parameter:

- client: name of client to frinstall packages from
- priority: priority of the package
- params: parameter for deinstalling the package

8.28.89 PKG_discardRemoveJob

Description: discards all remove jobs from the clientjobs table, that match the param
Parameter:

- client: name of the client
- packageName: name of the package

8.28.90 PKG_deletePackageselection

Description: delete all packages from package selection
Parameter:

- selectionName: name for the package selection

8.28.91 PKG_getAllPackageSelections

Description: returns all package selection names
Parameter:

- addEmpty: set to true to add an empty entry at the beginning.

8.28.92 PKG_multiPackageSelectionsSelection

Description: Generates a multi selection with all package selections.
Parameter:

- selName: name of the selection
- first: entry that should be shown first (this is the internal value and NOT the name shown to the user). the first value from the list will be written to $first. set first to "false" to disable writing the first entry.
- addEmpty: set to true to add an empty entry at the beginning.

8.28.93 PKG_showAllPackageSelections

Description: returns all package selection as HTML selection
Parameter:

- selName: name of the selection
- first: entry that should be shown first (this is the internal value and NOT the name shown to the user). the first value from the list will be written to $first. set first to "false" to disable writing the first entry.
- addEmpty: set to true to add an empty entry at the beginning.

8.28.94 PKG_getPackageParams

Description: gets the parameters for a selected package
Parameter:

- id: package ID

8.28.95 PKG_getClientbyPackageID

Description: gets the clientname that owns a selected package ID
Parameter:

- id: package ID

8.28.96 PKG_getInfoFromPackageID

Description: gets a row from "clientjobs" for a given package ID
Parameter:

- id: package ID
- variable: the name of the row (e.g. client)

8.28.97 PKG_getClientIDbyPackageID

Description: returns the ID of a client that owns a selected package ID
Parameter:

- id: package ID

8.28.98 PKG_getPackageParamsVar

Description: fetch the device for installation
Parameter:

- id: package ID
- var: name of variable you want to get the value of

8.28.99 PKG_getPackageIDsByName

Description: returnes all IDs as an array for jobs matching the client and job name and are a normal or special package.
Parameter:

- client: the name of the client, the jobs are for
- packageName: name of the package, can be the name of a normal or special package
- specialPackage: set to true, if you want to search for a special package

8.28.100 PKG_getClientPackages

Description: returns an array or a space separated list of all packages installed on a client
Parameter:

- client: the name of the client
- key: if it is not empty only packages that contain the key are returned
- arr: set to true if the result should be an array otherwise it's a string
- status: If set only returns packages of the given status

8.28.101 PKG_getPackagesListMarker

Description: returns the string to mark client names to store packages

8.28.102 PKG_savePackagesList

Description: stores the package names in the DB
Parameter:

- listName: name of the list to store the packages
- packages: array or blank seperated list of packages

8.28.103 PKG_loadPackagesList

Description: returns an array or a blank seperated list of all packages in the list
Parameter:

- listName: name of the list to store the packages
- arr: set to true if the result should be an arry otherwise it's a string

8.28.104 PKG_deletePackagesList

Description: deletes a packages list
Parameter:

- listName: name of the list to delete

8.28.105 PKG_addNormalJob

Description: Adds a normal package to the installation queue.
Parameter:

- client: the name of the client, the jobs are for
- packageName: name of the normal package
- priority: The priority of the job.

8.29 ./inc/pdf.php

Author: Hauke Goos-Habermann (HHabermann@pc-kiel.de)
Description: Functions for generating PDF files

8.29.1 PDF_showTableRow

Description: Shows a table row with a variable amount of entries. The parameters are shown side by side as rows in a table. If more than one PDF_showTableRow commands are executed in one table it is needed to always use the same amount of paramaters in each call.
Parameter:

- Arbitrary amount of cells to show in a table.

8.29.2 PDF_showTableHeader

Description: Inits some values for starting a new PDF table.

8.29.3 PDF_showTableEnd

Description: Prints the PDF table.

8.29.4 PDF_init

Description: Inits some basic variables for PDF creation.
Parameter:

- orientation: Orientation of the PDF document (P or Portrait, L or Landscape)
- unit: pt (point), mm (millimeter), cm (centimeter) or in (inch)
- format: PDF page format A3, A4, A5, Letter or Legal

8.29.5 PDF_output

Description: Shows the created PDF.

8.30 ./inc/plugin.php

Author: Hauke Goos-Habermann (HHabermann@pc-kiel.de)
Description: functions dealing with (de)installation of plugins and status information.

8.30.1 PLG_listMenuPlugins

Description: generate the menu entries for the plugins
Parameter:

- path: the path you want to scan for plugins

8.30.2 PLG_isPluginSelected

Description: checkes if the plugin was clicked
Parameter:

- path: path where to find the plugin files
- value: value the menu item should have to load the plugin page

8.30.3 PLG_isPluginSelected

Description: gets values from the plugins like plugin name, version,...
Parameter:

- fileName: file name with whole path to the plugin file
- var: variable you want to get the value from

8.30.4 PLG_getPLGName

Description: gets the name of the plugin
Parameter:

- fileName: file name with whole path to the plugin file

8.30.5 PLG_getPLGPage

Description: gets the page of the plugin
Parameter:

- fileName: file name with whole path to the plugin file

8.30.6 PLG_getPLGAuthor

Description: gets the author of the plugin
Parameter:

- fileName: file name with whole path to the plugin file

8.30.7 PLG_getPLGUpdateURL

Description: gets the update address of the plugin
Parameter:

- fileName: file name with whole path to the plugin file

8.30.8 PLG_getPLGClientRequires

Description: gets the "client requires" packages of the plugin
Parameter:

- fileName: file name with whole path to the plugin file

8.30.9 PLG_getPLGVersion

Description: gets the version of the plugin
Parameter:

- fileName: file name with whole path to the plugin file

8.30.10 PLG_showDownloadStatus

Description: shows the status of the plugin download
Parameter:

- fileName: file name with whole path to the plugin file

8.30.11 PLG_showDownloadStatus

Description: downloads or copies the plugin to the temp dir
Parameter:

- url: the place where to get the plugin file from. following transport protocols are allowed: http, ftp and local files. e.g. "http://myserver.de/test.m23plg" is avalid url.
- tempDir: where to store the plugin file temporary
- fileName: file name with whole path to the plugin file

8.30.12 PLG_checkOverwriting

Description: checks if the plugin files would overwrite existing files. the plugin file is extracted to a temporary directory. all file names are logged to a file that contains only the file names. these file names are checked against currently installed files. this routine checks if current files would be overwritten by the files of the plugin package. a list of files that would be overwritten is generated and aligned by a table.
Parameter:

- logfile: filename with whole path of the logfile containing the file names of the plugin file
- tempDir: where to store the plugin file temporary

8.30.13 PLG_DBInstall

Description: stores information about the plugin in the data base.
Parameter:

- tempDir: where to find the extracted files of the plugin
- files: all file names of the plugin file name

8.30.14 PLG_realInstall

Description: does the real installation
Parameter:

- tempDir: where to find the extracted files of the plugin

8.30.15 PLG_getTempDir

Description: generates the name for the plugin temp dir
Parameter:

- url: the place where to get the plugin file from. following transport protocols are allowed: http, ftp and local files. e.g. "http://myserver.de/test.m23plg" is avalid url.

8.30.16 PLG_getFilename

Description: gets the filename for the plugin file
Parameter:

- url: the place where to get the plugin file from. following transport protocols are allowed: http, ftp and local files. e.g. "http://myserver.de/test.m23plg" is avalid url.

8.30.17 PLG_install

Description: installs a plugin. extracts the files in the plugin file to a temporary directory. checks if currently existing files would be overwritten by the plugin files. if so, ask the user, if he wants to install or stop installation.
Parameter:

- url: the place where to get the plugin file from. following transport protocols are allowed: http, ftp and local files. e.g. "http://myserver.de/test.m23plg" is avalid url.

8.30.18 PLG_showPluginOverview

Description: shows a overview of all plugins

8.30.19 PLG_install

Description: deletes a plugin
Parameter:

- name: name of the plugin

8.30.20 PLG_getUpdateFile

Description: gets the update info file
Parameter:

- name: name of the plugin

8.30.21 PLG_update

Description: initalizes the update, shows information about the plugin update
Parameter:

- name: name of the plugin

8.30.22 PLG_listInfofile

Description: lists information of a plugin update file
Parameter:

- name: name of the plugin

8.30.23 PLG_realUpdate

Description: does the real installation/update
Parameter:

- name: name of the plugin
- url: the place where to get the plugin file from. following transport protocols are allowed: http, ftp and local files. e.g. "http://myserver.de/test.m23plg" is avalid url.

8.31 ./inc/pool.php

Author: Hauke Goos-Habermann (HHabermann@pc-kiel.de)
Description: functions for administrating package pools

8.31.1 POOL_selectPoolType

Description: shows buttons for selecting the type of pool and returns the pressed button

8.31.2 POOL_getPools

Description: returns an array with all pool names

8.31.3 POOL_showLoadDeleteCreate

Description: shows a dialog for loading, deleting and creating a pool
Parameter:

- poolName: name of the pool

8.31.4 POOL_create

Description: creates a new pool directory and type property file
Parameter:

- poolName: name of the pool
- poolType: type of the pool (cd or download)
- poolArch: CPU architecture for the packages

8.31.5 POOL_setProperty

Description: sets the contents of a property file
Parameter:

- poolName: name of the pool
- property: name of the pool property
- value: value to write in the pool property file

8.31.6 POOL_getProperty

Description: returns the contents of a property file
Parameter:

- poolName: name of the pool
- property: name of the pool property

8.31.7 POOL_delete

Description: deletes a pool
Parameter:

- poolName: name of the pool

8.31.8 POOL_showReadCD

Description: shows a dialog for copying the CD contents to the pool
Parameter:

- poolName: name of the pool

8.31.9 POOL_readCD

Description: copys the CD contents to the pool
Parameter:

- poolName: name of the pool
- mountPoint: the mount point of the CD drive

8.31.10 POOL_createExtendedPackageIndex

Description: creates the Packages* index files for the pool
Parameter:

- poolName: name of the pool

8.31.11 POOL_showCreatePackageIndex

Description: shows information (status of the Packages* generation, sources.list) about the currently generated pool

8.31.12 POOL_getSize

Description: returns the size of a pool in MB
Parameter:

- poolName: name of the pool

8.31.13 POOL_getDir

Description: returns the directory of the pool
Parameter:

- poolName: name of the pool

8.31.14 POOL_download

Description: shows error messages if the checks for distribution, sourceslist or packages are failing. Otherwise starts the distribution specific download routine.
Parameter:

- poolName: name of the pool
- distr: name of the distribution
- sourceslist: list of the package sources
- release: release branch of the choosen distribution to download the packages from
- downloadBasePackages: set to true if a bunch of basic packages should be included into the pool
- arch: download the packages for a specific CPU architecture

8.31.15 POOL_showDownloadStatus

Description: shows the package download status of a pool
Parameter:

- poolName: name of the pool

8.31.16 POOL_prepare

Description: Generates the needed configuration file for reprepro.
Parameter:

- poolName: name of the pool
- release: release of the distribution (e.g. sarge)
- distr: name of the distribution (e.g. Debian)
- arch: CPU architecture for the packages

8.31.17 POOL_makeRepository

Description: Generates a package source from packages stored in one directory.
Parameter:

- poolName: name of the pool
- archivPath: start search for packages in this subdirectory
- addCmds: additional commands that should be executed before starting the screen
-

8.31.18 POOL_getCDDistributionRelease

Description: Reads the distribution and the release name from a mounted CD and writes these information to the variables.
Parameter:

- mountPoint: the directory where the CD is mounted
- distr: the variable the name of the distribution (e.g. Debian) should be written to
- release: the variable release of the distribution (e.g. sarge) should be written to

8.31.19 POOL_showSourcesList

Description: Shows the sources list of a selected package source.
Parameter:

- poolName: name of the pool

8.32 ./inc/preferences.php

Author: Hauke Goos-Habermann (HHabermann@pc-kiel.de)
Description: functions to save and load preferences for client setup.

8.32.1 PREF_preferenceLoadManagerHandler

Description: Executes loading and deletion of preferences after pressing the according buttons and defines the buttons for PREF_showPreferenceManager();

8.32.2 PREF_preferenceSaveManagerHandler

Description: Executes the saving of preferences.

8.32.3 PREF_showPreferenceManager

Description: Shows a dialog to load and delete existing preferences and to create new preferences.

8.32.4 PREF_saveAllPreferenceValues

Description: Saves all values of a session into the preference.

8.32.5 PREF_loadAllPreferenceValues

Description: Loads all values of a preference into the session.

8.32.6 PREF_getClientPreferences

Description: list all preferences
Parameter:

- default: The name of the preference to list first
- directOutput: If enabled the preference names will be given out as a HTML option list. If disabled an array with the preference names as key and value will be returned.

Returns:

- Array with the preference names or nothing on enabled directOutput.

8.32.7 PREF_getValue

Description: gets a value from a selected preference. with preferences you can store variables and values for reuse.
Parameter:

- name: the name of the preference
- var: variable of the preference

8.32.8 PREF_putValue

Description: stores a value to a selected preference. with preferences you can store variables and values for reuse.
Parameter:

- name: the name of the preference
- var: variable of the preference
- value: value you want to set

8.32.9 PREF_delete

Description: deletes a preference
Parameter:

- name: the name of the preference

8.32.10 PREF_exists

Description: checks if a preference with the selected name exists
Parameter:

- name: the name of the preference

8.32.11 PREF_putAllOptions

Description: stores all settings in the options array to the preferences
Parameter:

- prefName: name of the preference the options should be stored under
- options: the array with the options

8.32.12 PREF_getAllValues

Description: gets all preferences and adds them to the options array
Parameter:

- prefName: name of the preference the options should be stored under
- options: the array with the options

8.33 ./inc/remotevar.php

Author: Hauke Goos-Habermann (HHabermann@pc-kiel.de)
Description: functions for saving and loading serverside variables.

8.33.1 RMV_exists4IP

Description: checks if a variable exists for a certain IP
Parameter:

- var: name of the variable to store
- ip: ip address to set the variable for

8.33.2 RMV_set4IP

Description: creates or updates a variable for a special ip
Parameter:

- var: name of the variable to store
- value: value to set
- ip: ip address to set the variable for

8.33.3 RMV_get4IP

Description: gets the value of a variable for the given ip
Parameter:

- var: name of the variable to get the value from
- ip: ip you want to get the value for

8.33.4 RMV_set

Description: creates or updates a variable for the ip of the calling client
Parameter:

- var: name of the variable to store
- value: value to set

8.33.5 RMV_get

Description: gets the value of a variable for the ip of the calling client
Parameter:

- var: name of the variable to get the value from

8.33.6 RMV_rm4IP

Description: removes a variable for a slelected ip
Parameter:

- var: name of the variable to get the value from
- ip: ip you want to delete the value from

8.33.7 RMV_rm

Description: removes a variable for the ip of the calling client
Parameter:

- var: name of the variable to get the value from

8.33.8 RMV_rm_old

Description: removes all vars older than $time seconds
Parameter:

- time: time in seconds

8.34 ./inc/scredit.php

Author: Hauke Goos-Habermann (HHabermann@pc-kiel.de)
Description: Functions for the embedded script editor.

8.34.1 SCREDIT_correctScriptFilename

Description: Corrects the given filename to have it a valid prefix and suffix.
Parameter:

- filename: The filename to check and correct.

8.34.2 SCREDIT_showEditor

Description: Shows a script editor with syntax highlighting if JavaScript is enabled or a normal textarea input dialog.

8.34.3 SCREDIT_newScriptTemplate

Description: Returns a template for a basic script.
Returns:

- Text of the script template.

8.35 ./inc/server.php

Author: Hauke Goos-Habermann (HHabermann@pc-kiel.de)
Description: functions for getting information from the server

8.35.1 SERVER_importGPGPackageSignKey

Description: Imports the m23 GPG package sign key.

8.35.2 SERVER_logLocalScreenSessionToFile

Description: Logs the output of a local screen session to a file.
Parameter:

- session: name screen session to log.
- user: user the script runs under

Returns:

- Full path to the log file.

8.35.3 SERVER_setSSLCertCheckDisabled

Description: Sets, if the SSL certificate check is disabled globally for all clients.
Parameter:

- : disableSSLCertCheck: true, if the check is disabled otherwise false.

8.35.4 SERVER_isSSLCertCheckDisabled

Description: Determines, if the SSL certificate check is disabled globally for all clients.
Returns:

- : true, if the check is disabled otherwise false.

8.35.5 SERVER_setServerSetting

Description: Sets the value of a server setting.
Parameter:

- : var: Name of the setting.
- : val: Value of the setting.

8.35.6 SERVER_getServerSetting

Description: Gets the value of a server setting.
Parameter:

- : var: Name of the setting.

Returns:

- : Value the setting.

8.35.7 SERVER_existsServerSetting

Description: Checks, if a named server setting exists in the DB.
Parameter:

- : var: Name of the setting.

Returns:

- : True if the setting exists.

8.35.8 SERVER_killPID

Description: Kills a process running under a given PID or sends a signal.
Parameter:

- : pid: The PID of the process to kill.
- : signal: The signal to send to the process.

Returns:

- : True if the process was killed or got the signal.

8.35.9 SERVER_killBackgroundJob

Description: Kills a job (that runs in screen) with a given name.
Parameter:

- job: Name of the job that should be killed.
- user: User the job runs under.

8.35.10 SERVER_insertLineNumber

Description: Inserts a text AT or AFTER a line number or creates a new file with the given name, if it doesn't exists.
Parameter:

- file: the name of the file
- lineNumber: reference line number for inserting
- insertText: text to insert
- mode: The access mode the newly created file should have.
- insertMode: "0" insert AT, "1" insert AFTER line number
- addIfNotExists: set to true, if the the line should be added only if the line doesn't exist. false, if the line should be added on every execution.

8.35.11 SERVER_addAdmin

Description: Adds an administrator with all access rights.
Parameter:

- newadmin: Name of the new admin to create.
- password: Password for the admin account.

Returns:

- : true, if the deletion was sucessfully otherwise false.

8.35.12 SERVER_delAdmin

Description: Deletes an administrator with all access rights.
Parameter:

- name: Name of the admin to delete.

Returns:

- : true, if the deletion was sucessfully otherwise false.

8.35.13 SERVER_fileExists

Description: Checks if a file exits, that the Apache user has never access to.
Parameter:

- : file: Name (with full path) of the file to check.

Returns:

- : True, if the file exists other wise false.

8.35.14 SERVER_getPublicSSHKeyOfm23Server

Description: Returns the public SSH key of the m23 server.
Returns:

- : Public SSH key of the m23 server.

8.35.15 SERVER_changeHtpasswd

Description: Changes the password of a user in a htpasswd file.
Parameter:

- : htpasswdFile: htpasswd file that contains user names and crypted passwords.
- : username: Name of the user to change
- : password: The according new password

Returns:

- : True if the password was changed sucessfully.

8.35.16 SERVER_delFromHtpasswd

Description: Removes a user with password to a htpasswd file.
Parameter:

- : htpasswdFile: htpasswd file that contains user names and crypted passwords.
- : username: Name of the user to remove

Returns:

- : True if the new user was added sucessfully.

8.35.17 SERVER_addToHtpasswd

Description: Adds a new user with password to a htpasswd file.
Parameter:

- : htpasswdFile: htpasswd file that contains user names and crypted passwords.
- : username: Name of the new user to add
- : password: The according password for the new user

Returns:

- : True if the new user was added sucessfully.

8.35.18 SERVER_dhcpServerInNetWarn

Description: Shows an error message if there is found another DHCP server on the net.
Returns:

- : false, if the IP address is static.

8.35.19 SERVER_sendScriptToSF

Description: Uploads a script to m23.sf.net for public use.
Parameter:

- name: Name of the script.
- author: Name of the script author.
- description: Short descriptive text for the purpose of the script.
- script: Source code of the script.

8.35.20 SERVER_dynamicIPWarn

Description: Shows an error message if the m23 server has a dynmic IP address.
Returns:

- : false, if the IP address is static.

8.35.21 SERVER_tmpNotWritable

Description: Shows an error message if /tmp is not writable.
Returns:

- : false, if /tmp is writable.

8.35.22 SERVER_rootFreeSpace

Description: Shows an error message if the free space of the root partition is low.
Returns:

- : false, if there is enough space.

8.35.23 SERVER_isProgramRunning

Description: checks if a certain program is running and returns true, if yes "no" otherwise
Parameter:

- progname: the name of the programm (e.g. "apache" for the Apache web server)

8.35.24 SERVER_checkPackageInstalled

Description: checks if a certain package is installed
Parameter:

- pkgName: the name of the package

8.35.25 SERVER_daemonStartStop

Description: starts, stops and restarts daemons
Parameter:

- daemonScript: the file name of the script, that handles the real starting, stopping and restarting and understands the $action
- action: start, stop or restart

Returns:

- : true on successfully execution otherwise false.

8.35.26 SERVER_getAptGetInstallCommand

Description: Returns the apt-get commands to install a tool on the server.
Parameter:

- pkgName: name of the software package

Returns:

- : apt-get commands to install a tool on the server.

8.35.27 SERVER_installTool

Description: installs a tool on the server
Parameter:

- pkgName: name of the software package

Returns:

- : true on successfully execution otherwise false.

8.35.28 SERVER_installToolInBackground

Description: Installs a tool on the server in background.
Parameter:

- pkgName: name of the software package

8.35.29 SERVER_programmStatus

Description: shows a row with information about the status of a certain program, with the possibillity to start, stop or restart the program.
Parameter:

- progname: the name of the programm (e.g. "apache" for the Apache web server)
- daemonScript: set it to the script that should be used for starting, stopping and restarting. If the script name isn't set, this is a normal tool and NOT a daemon.
- canBeInstalled: set to "true" if the programm can be installed by the package name

8.35.30 SERVER_apacheInfo

Description: returnes an information string for the Apache server

8.35.31 SERVER_mysqlInfo

Description: returnes an information string for the MySQL server

8.35.32 SERVER_dhcpInfo

Description: returnes an information string for the DHCP server

8.35.33 SERVER_LDAPInfo

Description: Returnes an information string for the LDAP server.

8.35.34 SERVER_programmStatusTableHeader

Description: shows the header of the table needed for the programm status lines

8.35.35 SERVER_runInBackground

Description: Runs a script with "screen" in the background under a given user
Parameter:

- jobName: name of the job screen should show
- cmds: the commands of the script
- user: user the script should be run under
- runInScreen: Set to true if the execution should be done in "screen". False executes it under the normal BASH.

8.35.36 SERVER_runningInBackground

Description: Returns "true" if a lock file for a given job name is existing.
Parameter:

- jobName: name of the job

8.35.37 SERVER_runningInScreen

Description: Returns "true" if a screen session with a given name exists for a given user.
Parameter:

- jobName: name of the job.
- user: User the screen session is run under.

8.35.38 SERVER_addLineToFile

Description: Adds (if the search pattern can't be found) a line to a file on the server
Parameter:

- file: name of the file to edit
- search: the search pattern
- add: the line to add

Returns:

- : true on successfully execution otherwise false.

8.35.39 SERVER_deleteFile

Description: Deletes a file from the server.
Parameter:

- fileName: Name of the file to delete.

Returns:

- : true on successfully execution otherwise false.

8.35.40 SERVER_getFileContents

Description: Get the contents of any file (even if only readable by root).
Parameter:

- fileName: Name of the file to read.

Returns:

- : Contents of the file

8.35.41 SERVER_putFileContents

Description: Stores a text to a file and changes it's mode, user and group.
Parameter:

- fileName: Name of the file to put the text to.
- text: The contents the file should have.
- mode: The access mode the file should have.
- user: The owner of the file.
- group: The owning group of the file.

Returns:

- true on success and false otherwise.

8.35.42 SERVER_delLineFromFile

Description: Deletes lines from the file that match the search pattern
Parameter:

- file: name of the file to edit
- search: the search pattern

Returns:

- : true on successfully execution otherwise false.

8.35.43 SERVER_addEtcHosts

Description: Adds a host to /etc/hosts and /etc/backuppc/hosts (if it doesn't exists allready)
Parameter:

- hostname: name of the host to add
- ip: its IP

8.35.44 SERVER_delEtcHosts

Description: Deletes a host entry from /etc/hosts and /etc/backuppc/hosts
Parameter:

- hostname: name of the host to delete

8.35.45 SERVER_getInstallationMedium

Description: Tries to figure out how the m23 server was installed
Returns:

- : CD, Internet or Unknown source.

8.35.46 SERVER_getOS

Description: Returns the version string of the distribution.
Returns:

- : version string.

8.35.47 SERVER_checkDownload

Description: Downloads a special file from m23.sf.net and checks if the size and md5 sum are matching.
Parameter:

- : useProxy: Set to true if the local proxy should be used.

Returns:

- : Status information if file size and md5 sum are matching.

8.35.48 SERVER_checkDiskFree

Description: Reports the free space of all mounted media.
Returns:

- : Output of "df".

8.35.49 SERVER_checkRunInVM

Description: Checks if the m23 server is executed in a virtual machine or on native hardware.
Returns:

- : VMWare, VirtualBox or native.

8.35.50 SERVER_checkKernel

Description: Returns the kernel information string of "uname -a".
Returns:

- : Kernel information string.

8.35.51 SERVER_multiMkDir

Description: Creates a directory and all needed directories on the way to the destination path.
Parameter:

- path: The complete path to create.
- mode: The access mode of the path to create (should start with "0" e.g. 0777)

8.35.52 SERVER_commandAvailable

Description: Checks, if a given command is available for the given user.
Parameter:

- user: user the command should be run under.

Returns:

- : true when the command is available otherwise false.

8.36 ./inc/sourceslist.php

Author: Hauke Goos-Habermann (HHabermann@pc-kiel.de)
Description: function to generate the sources.list for the client

8.36.1 SRCLST_getAddToFile

Description: Returns addToFile paramters from the given sources list as an associative array, where file name and file contents are seperated.
Parameter:

- sourceName: The name of the package source list

Returns:

- : Associative array with file name and file contents (e.g. [0] => Array ([file] => file1.txt, [text] => text1), [1] => Array ([file] => file2.txt, [text] => text2), ...)

8.36.2 SRCLST_getRelease

Description: Gets a release from the sourceslist table.
Parameter:

- name: the name of the package source list

Returns:

- Release name of choosen sources list.

8.36.3 SRCLST_genList

Description: generates the sources.list file for the client
Parameter:

- clientName: the name of the client

8.36.4 SRCLST_saveArchitectures

Description: Saves the architectures for package source list.
Parameter:

- sourceName: the name of the package source list
- archs: Associative array with the supported CPU architectures.

8.36.5 SRCLST_saveList

Description: saves the package source list
Parameter:

- name: the name of the package source list
- list: the list of sources as simple text
- description: a descriptive text for the list
- distr: the name of the distribution the list is for
- release: the name of the release the list is for

8.36.6 SRCLST_querySourceslists

Description: returns the result of the DB query after sourceslists for a special distribution
Parameter:

- distr: the distribution the sources list is for or "*" for all distributions

8.36.7 SRCLST_genSelection

Description: generates a HTML selection with the names of alls package sources
Parameter:

- selName: the name of the selection
- first: the package source that should be shown first
- distr: the distribution the sources list is for or "*" for all distributions

8.36.8 SRCLST_getValue

Description: gets a value from the sourceslist table
Parameter:

- name: the name of the package source list
- var: the name of the table row

8.36.9 SRCLST_loadSourceListFromDB

Description: loads and returnes the the package source list from the DB.
Parameter:

- name: the name of the package source list

8.36.10 SRCLST_loadSourceList

Description: Loads and returnes the package source list and tries to find a valid mirror for m23debs.
Parameter:

- name: the name of the package source list

Returns:

- package source list

8.36.11 SRCLST_getDescription

Description: returnes the the package source description
Parameter:

- name: the name of the package source list

8.36.12 SRCLST_delete

Description: deletes package source
Parameter:

- name: the name of the package source list

8.36.13 SRCLST_checkList

Description: checks a package info and returns the output of the OS package update function
Parameter:

- sourceName: the name of the package source list

8.36.14 SRCLST_packageInformationChangeInformationHumanReadable

Description: Returns the time point when the package information was changed last.
Parameter:

- distr: the short name of the distribution
- sourceName: the name of the package source list

Returns:

- : Time when the package information was changed last.

8.36.15 SRCLST_packageInformationChangeTime

Description: Returns the time point when the package information was changed last.
Parameter:

- distr: the short name of the distribution
- sourceName: the name of the package source list
- changedBefore: Amount of seconds before the package information was changed.

Returns:

- : Time when the package information was changed last.

8.36.16 SRCLST_packageInformationOlderThan

Description: Checks if a package info is older than a selected amount of minutes or if the package info directory is too smal.
Parameter:

- minutes: the amount of minutes the package information can be older to return true
- distr: the short name of the distribution
- sourceName: the name of the package source list

Returns:

- : true when package info is older than a selected amount of minutes or if the package info directory is too smal, otherwise false.

8.36.17 SRCLST_getStorageFS

Description: Returns a file systems that can be used to install the OS and to store data. A wished file system is given and an alternative FS is returned, if this FS is not supported.
Parameter:

- fs: File system to probe.
- sourceName: The name of the package source list

Returns:

- : File systems that can be used to install the OS and to store data

8.36.18 SRCLST_supportedFS

Description: Returns an array with file systems that supported by the OS.
Parameter:

- sourceName: The name of the package source list

Returns:

- : Array with file systems supported by the OS.

8.36.19 SRCLST_alternativeFS

Description: Returns the alternative file system that is supported by the OS.
Parameter:

- sourceName: The name of the package source list

Returns:

- : File system.

8.36.20 SRCLST_getParameter

Description: Returns special parameter(s) from the given sources list.
Parameter:

- sourceName: The name of the package source list
- parameter: The name of the parameter.

Returns:

- : Values for the given parameter in an array.

8.36.21 SRCLST_getMirror

Description: returns the mirror from the sources list
Parameter:

- sourceName: the name of the package source list

Returns:

- URL to the mirror

8.36.22 SRCLST_getDesktopList

Description: returnes an array with all supported desktops
Parameter:

- sourceName: the name of the package source list

8.36.23 SRCLST_showDesktopsSel

Description: returnes a selections with all desktops supported by the sources list
Parameter:

- sourceName: the name of the package source list
- selName: the name of the selection
- first: the desktop that should be shown first

8.36.24 SRCLST_doesDistrSupportEFI

Description: Checks, if a sources list contains a distribution that supports EFI.
Parameter:

- sourceName: the name of the package source list

Returns:

- true, if the distribution supports EFI, otherwise false.

8.36.25 SRCLST_getListnamesWithEfiSupport

Description: Gets a list with all sources lists that support EFI.
Returns:

- Array with all sources lists that support EFI.

8.36.26 SRCLST_clientUsesEfiButSourcesListDoesntSupportEfi

Description: Checks, if the client uses EFI and the choosen sources list doesn't.
Parameter:

- client: Name of the client.
- sourceName: The name of the package source list.

Returns:

- : true, if the client uses EFI and the choosen sources list doesn't, otherwise false.

8.36.27 SRCLST_showErrorIfClientUsesEfiButSourcesListDoesntSupportEfi

Description: Shows an error message, if the client uses EFI and the choosen sources list doesn't.
Parameter:

- client: Name of the client.
- sourceName: The name of the package source list.

Returns:

- : false, if the client uses EFI and the choosen sources list doesn't, otherwise true.

8.36.28 SRCLST_showAlternativeArchitectureSelection

Description: Shows a list with available CPU architectures of the sources list, in case that the wanted architecture is not available in the sources list. The alternative architecture will be written to the arch option of the client.
Parameter:

- sourceName: The name of the package source list.
- wantedArch: The CPU architecture of the m23 client.
- client: Name of the client.

Returns:

- : A CPU architecture supported by the package source list.

8.36.29 SRCLST_isArchAvailable

Description: Checks if a given architecture is supported by the sources list.
Parameter:

- sourceName: the name of the package source list
- arch: Architecture to check for.

Returns:

- true, if the architecture is supported, false otherwise.

8.36.30 SRCLST_getArchitectures

Description: Returnes a list of all CPU architectures supported by the sources list.
Parameter:

- sourceName: the name of the package source list

Returns:

- Associative array with the supported CPU architectures as variable AND key.

8.36.31 SRCLST_showEditor

Description: shows an editor for sources lists
Parameter:

- poolName: if it is set, the editor shows a package download dialog for the selected pool

8.36.32 SRCLST_getListnames

Description: Returns an array that contains all sourceslist names
Parameter:

- distr: the distribution the sources list is for or "*" for all distributions

8.36.33 SRCLST_cleanList

Description: Returns an array with all lines of the sources list that contain Debian sources
Parameter:

- list: the contents of the sources list

8.36.34 SRCLST_matchList

Description: Returns the name of the sources list that matches the searched sources list contents for the distribution or false
Parameter:

- distr: the distribution to search the name of the sources list under
- search: the contents of the sources list to search

8.36.35 SRCLST_possiblem23debsMirrors

Description: Returns an array with mirrors for m23 debs.
Returns:

- Array with mirrors for m23 debs.

8.36.36 SRCLST_checkm23debsMirror

Description: Checks, if the url contains a valid mirror for m23debs.
Parameter:

- url: URL of the (possible) m23debs mirror.

Returns:

- true, if the url contains a valid mirror for m23debs, otherwise false.

8.36.37 SRCLST_getWorkingm23debsMirror

Description: Get the url of a working m23debs mirror.
Returns:

- Url to a working m23debs mirror or false, if none could be found.

8.37 ./inc/update.php

Author: Hauke Goos-Habermann (HHabermann@pc-kiel.de)
Description: functions for updating the server

8.37.1 UPDATE_doUpdate

Description: downloads and executes the update script.
Parameter:

- URL: url to fetch the update file from

8.37.2 UPDATE_running

Description: checks, if an update is running (returns true otherwise false)

8.37.3 UPDATE_getUrl

Description: returnes a correct URL to the update source
Parameter:

- base: URL to the update script
- command: "info" or "cmd"
- version: m23 version
- patchLevel: patch version number

8.37.4 UPDATE_getInfo

Description: returns the information text from the URL
Parameter:

- URL: URL to the information text
- refreshTime: The time in minutes the file is downloaded again.

8.38 ./inc/vm.php

Author: Hauke Goos-Habermann (HHabermann@pc-kiel.de)
Description: Functions for managing virtual clients.

8.38.1 VM_captureVMScreenAsMovie

Description: Enables/disables capturing the screen of a VM to a movie file.
Parameter:

- type: VM_SW_VBOX for VirtualBox OSE
- vmname: Name of the VM.
- enable: true for enabling the capturing, false for disabling.
- movieFile: File to store the capturing in.
- width: Width of the movie.
- height: Height of the movie.
- rate: Bitrate of the movie.
- fps: Frames per second

Returns:

- BASH code for enabling/disabling the capturing of the VM's screens to a movie file.

8.38.2 VM_CloudStackDeleteClientVM

Description: Deletes a virtual machine for use with m23 in CloudStack, only a cloudstack admin can recover it
Parameter:

- virtualMachineId: CloudStack ID of the virtual machine
- VMDeletionOK: True if VM was successfully deleted, false otherwise

Returns:

- ErrorMessages or Success messages, sets parameter VMDeletionOK (true if all went well, false if an error ocurred)

8.38.3 VM_isCloudStackClient

Description: Checks, if the client is run in CloudStack
Returns:

- true, when the client is run in CloudStack otherwise false.

8.38.4 VM_CloudStackCheckConstants

Description: Checks, if the given constant values are valid.
Parameter:

- CLOUDSTACK_API_ENDPOINT: The API entpoint.
- CLOUDSTACK_API_KEY: The API key.
- CLOUDSTACK_SECRET_KEY: The secret API key.

Returns:

- true, when the constant values are valid otherwise false.

8.38.5 VM_CloudStackConfigGUI

Description: Shows a dialog for editing the CloudStack config file and uploading the m23 client ISO.

8.38.6 VM_CloudStackWriteConfFile

Description: Writes the CloudStack config file or writes a basic config file, if it does not exist.
Parameter:

- overwrite: Set to true, if the config file should be overwritten in any case.
- CLOUDSTACK_API_ENDPOINT: The API endpoint.
- CLOUDSTACK_API_KEY: The API key.
- CLOUDSTACK_SECRET_KEY: The secret API key.
- CLOUDSTACK_SERVICE_OFFERING_ID: The virtual CPU and RAM combination to use for a new VM.
- CLOUDSTACK_TEMPLATE_ID: The ID of the m23 client installation ISO.
- CLOUDSTACK_NETWORKIDS: The ID of the network to use.
- CLOUDSTACK_DISK_OFFERING_ID: The virtual hard disk type.

8.38.7 VM_CloudStackUploadIso

Description: Uploads and registers a new bootable ISO file into cloudstack from a given website
Parameter:

- isoName: the name you choose for the ISO file
- isoUrl: the url from where you want cloudstack to download the ISO file
- zoneID: The ID of the CloudStack zone.
- isoUploadSuccess: is set to True if action succeeded, false otherwise
- isoID: is set to Cloudstack-Iso-ID if action succeeded, otherwise not set

Returns:

- textmessage about result (errormessage or success message) and sets isoUploadSuccess to True if action succeeded, false otherwise, sets isoID to Iso-ID

8.38.8 VM_CloudStackEnablePortForwarding

Description: creates a port forwarding rule for a virtual machine, with private port and public port being the same
Parameter:

- virtualMachineId: the cloudstack ID of the virtual machine to which the rule shall apply
- pFSuccess: is set to true, if the rule was created

Returns:

- textmessage about result (errormessage or success message) and sets pFSuccess to True if action succeeded, false otherwise

8.38.9 VM_CloudStackDisablePortForwarding

Description: deletes a port forwarding rule for a virtual machine, with private port and public port being the same (CLOUDSTACK_X2GO_PORTNUMBER)
Parameter:

- virtualMachineId: the cloudstack ID of the virtual machine from which the port forwarding rule shall be deleted
- pFDSuccess: is set to true, if the rule was deleted successfully

Returns:

- textmessage about result (errormessage or success message) and sets pFDSuccess to True if action succeeded, false otherwise

8.38.10 VM_CloudStackSendSetVisualURL

Description: Sends the visual URL (current client ip:22) to the m23 server, if run under CloudStack.

8.38.11 VM_CloudStackStartVM

Description: starts a virtual machine in CloudStack
Parameter:

- clientname: CloudStack name of the instance / name of the m23 client.
- startVMOK: true if started successfully or already running, false otherwise

Returns:

- textmessage with result of start or error message

8.38.12 VM_CloudStackStopVM

Description: stops a virtual machine in CloudStack
Parameter:

- clientname: CloudStack name of the instance / name of the m23 client.
- stopVMOK: true if stopped successfully or already stopped, false otherwise

Returns:

- textmessage with result of stop or error message

8.38.13 VM_CloudStackGetVMStatus

Description: gets the status of a virtual machine
Parameter:

- clientname: CloudStack name of the instance / name of the m23 client.

Returns:

- textmessage with machine status (like 'Running' or 'Stopped') or FALSE if no status could be retrieved (e.g. if machine doesn't exist)

8.38.14 VM_CloudStackClientName2ClientID

Description: returns the Cloudstack-ID of a client with the given client host name
Parameter:

- clientname: Host name of the virtual machine

Returns:

- Cloudstack-Client-ID if the clientname can be retrieved, False otherwise

8.38.15 VM_CloudStackNetBootActivate

Description: attaches/exchanges or removes (if any) a network boot ISO to or from the client
Parameter:

- clientname: CloudStack name of the instance / name of the m23 client.
- activate: TRUE for attaching ISO, FALSE for removing
- nBASuccess: is set to True if action succeeded, false otherwise

Returns:

- textmessage about result (errormessage or success message) and sets nBASuccess to True if action succeeded, false otherwise

8.38.16 VM_CloudStackCreateVM

Description: Creates a virtual machine for use with m23 in CloudStack
Parameter:

- name: Name of the virtual machine, can contain ASCII letters 'a' through 'z', the digits '0' through '9', and the hyphen ('-'), must be between 1 and 63 characters long, and can't start or end with "-" and can't start with digit
- zoneID: zoneID for CloudStack

8.38.17 VM_CloudStack_getServerIP

Description: Gets the external m23 server IP if the m23 server is run as CloudStack VM.
Returns:

- External m23 server IP.

8.38.18 VM_GUIstepCreateCloudStackVM

Description: Shows a dialog to create a new VM in CloudStack.

8.38.19 VM_CloudStack_available

Description: Checks, if the CloudStack configuration file is included and contains the needed constants.
Returns:

- true, if the CloudStack are present.

8.38.20 VM_CloudStack_getObject

Description: Gets a new CloudStackClient object.
Parameter:

- CLOUDSTACK_API_ENDPOINT: The API entpoint.
- CLOUDSTACK_API_KEY: The API key.
- CLOUDSTACK_SECRET_KEY: The secret API key.

Returns:

- New CloudStackClient object.

8.38.21 VM_CloudStack_getVersion

Description: Gets the version of CloudStack.
Returns:

- CloudStack version.

8.38.22 VM_shutdownAndDisableNetbootAfterInstall

Description: Reboots an VM and disables network booting.
Parameter:

- vmname: Name of the VM.

8.38.23 VM_rebootAndActivateNetboot

Description: Reboots an VM and activates network booting.
Parameter:

- vmname: Name of the VM.

8.38.24 VM_shutdownAndDisableNetbootAfterInstall

Description: Shuts down an VM and disables network booting.
Parameter:

- vmName: Name of the VM.

8.38.25 VM_shutdownAndDisableNetboot

Description: Generates a BASH command to shut down an VM and to disable network booting.
Parameter:

- type: VM_SW_VBOX for VirtualBox OSE
- vmname: Name of the VM.

Returns:

- BASH code to shut down an VM and to disable network booting.

8.38.26 VM_rebootChangeBootDevice

Description: Generates a BASH command to reboot an VM and to disable network booting.
Parameter:

- type: VM_SW_VBOX for VirtualBox OSE
- vmname: Name of the VM.
- visual: If set to true, the VM should be run in visual mode otherwise in headless mode.

Returns:

- BASH code to reboot an VM and to disable network booting.

8.38.27 VM_getVBoxVersion

Description: Get the currently installed VirtualBox version of the host.
Parameter:

- clientNameOrIP: The name of the client or localhost or an IP.

Returns:

- The version number of VirtualBox.

8.38.28 VM_setVBoxAddonAsDefault

Description: Sets a choosen VirtualBox addition package version as default.
Parameter:

- version: Version number of the VirtualBox addition to set as default.

8.38.29 VM_downloadedVBoxAddons

Description: Lists all VirtualBox addition package versions that can be downloaded from the m23 server.
Returns:

- Associative array with ther version numbers of all VirtualBox addition packages that can be downloaded from the m23 server.

8.38.30 VM_getVBoxAddonDefaultVersion

Description: Gets the version number of the VirtualBox addition package.
Returns:

- The default version of the VirtualBox addition package.

8.38.31 VM_generateVBOXaddonDownloadCMD

Description: Generates the download commands to download a VirtualBox addition ISO and to extract the addition installers for Linux.
Parameter:

- version: Version number of the VirtualBox addition to download and extract.

8.38.32 VM_downloadVBOXaddons

Description: Downloads the VirtualBox addition ISOs and extracts the addition installers for Linux.
Parameter:

- checkedVersions: Array with all version numbers of the VirtualBox additions to download.

8.38.33 VM_VBOXaddonDownloadDialog

Description: Shows a dialog for downloading the VirtualBox additions to the m23 server.

8.38.34 VM_wasVBoxAddonDownloaded

Description: Checks, if the VirtualBox addition for a selected version was downloaded to the m23 server.
Parameter:

- version: Version number of the VirtualBox addition to look for.

Returns:

- True, if the addition is there.

8.38.35 VM_listDownloadableVBoxAddons

Description: Returns an array with the version numers of all VirtualBox addition ISOs that are 2.0.0 and above.
Returns:

- Array with the version numers of all VirtualBox addition ISOs that are 2.0.0 and above as key and value.

8.38.36 VM_stopVM

Description: Generates a BASH command to stop a virtual machine.
Parameter:

- type: VM_SW_VBOX for VirtualBox OSE
- vmname: Name of the VM.

Returns:

- BASH code to stop a virtual machine.

8.38.37 VM_pauseVM

Description: Generates a BASH command to pause a virtual machine.
Parameter:

- type: VM_SW_VBOX for VirtualBox OSE
- vmname: Name of the VM.

Returns:

- BASH code to pause a virtual machine.

8.38.38 VM_resumeVM

Description: Generates a BASH command to resume a virtual machine.
Parameter:

- type: VM_SW_VBOX for VirtualBox OSE
- vmname: Name of the VM.

Returns:

- BASH code to resume a virtual machine.

8.38.39 VM_webAction

Description: Executes an action for a VM controlled by the web UI.
Parameter:

- vmName: Name of the VM.
- action: Action for the VM given by the URL parameter.

Returns:

- True if the command can be executed otherwise false.

8.38.40 VM_delete

Description: Deletes a virtual machine from a VM host.
Parameter:

- vmname: Name of the VM.

Returns:

- true if it's an VM or false if not.

8.38.41 VM_vmSwNr2Name

Description: Converts the VM software constant (VM_SW_*) to the human readable name.
Parameter:

- vmType: Code number of the virtualisation software.

Returns:

- Human readable name of the VM software.

8.38.42 VM_getHTMLStatusBlock

Description: Generates and returns a status block in a HTML table with informations (VM host, VM software, VM power switch state, visual console URL and password, VM NICs) about the selected VM client.
Parameter:

- clientName: Name of the VM client.

Returns:

- HTML table with information about the VM.

8.38.43 VM_activateNetboot

Description: (De)Activates network booting of a VM.
Parameter:

- vmName: Name of the VM.
- activate: true for booting from network, false for booting from the HD.

Returns:

- The message of the VM management tool or false if it's not a VM.

8.38.44 VM_convertSwitchStatusInfo

Description: Returns the status of a VM guest in several ways.
Parameter:

- status: Status of the VM guest (one of VM_STATE_*)

Returns:

- Associative array with: $out['text']: The status as text in the current language. $out['icon']: The icon of the given status (as traffic lights). $out['imgTag']: The status as traffic light in an HTML img tag with the written status as title.

8.38.45 VM_getSWandHost

Description: Gets the VM software and VM host of a m23 client.
Parameter:

- clientName: Name of the virtualised client.

Returns:

- $out['vmSoftware']: The VM software used for the guest. $out['vmHost']: The name of the VM host.

8.38.46 VM_getStatus

Description: Returns the current status of a VM guest.
Parameter:

- clientName: Name of the virtualised client.

Returns:

- Array with the current state of the VM or false is the client is no VM guest.

8.38.47 VM_GUIstepCreateGuest

Description: Shows a dialog to create a new VM on the chosen host.

8.38.48 VM_GUIstepCheckHost

Description: Shows a dialog part with information about the chose VM host.

8.38.49 VM_GUIstepSelectHost

Description: Shows a dialog part for choosing the VM host.
Parameter:

- VM_software: Code number of the virtualisation software.

Returns:

- Gives back the VM host or false if there is no host for the choosen virtualisation solution.

8.38.50 VM_getAllVMHosts

Description: Returns a list of all VM hosts with a choosen virtualisation software.
Parameter:

- VM_software: Code number of the virtualisation software.

Returns:

- Associative array with the hostname as key and value.

8.38.51 VM_setVisualURL

Description: Sets the URL to connect to the visual management console.
Parameter:

- VMguest: Name of the guest that is run in the virtualisation software.
- url: The URL to connect to the visual management console (e.g. 192.168.1.23:23 with VNC).

Returns:

- MySQL resource or false on error.

8.38.52 VM_setHostInDB

Description: Sets the password for the login to the visual management console on the host for all guests, the host flag and the type of used virtualisation software.
Parameter:

- VMhost: Name of the host with the virtualisation software.
- password: Password to set.
- vmSoftware: Type of the virtualisation software.

Returns:

- MySQL resource or false on error.

8.38.53 VM_setGuestInDB

Description: Makes the client a VM guest in the DB.
Parameter:

- clientName: Name of the m23 client (VM guest)
- VMSoftware: Type of the virtualisation software.
- VMHostName: Name of the m23 client (VM host)

Returns:

- MySQL resource or false on error.

8.38.54 VM_statusIcons

Description: Returns HTML codes that include the VM status icons of the client.
Parameter:

- param: Client's parameter array.

Returns:

- HTML codes with included status icons.

8.38.55 VM_createDiskImage

Description: Creates a new empty virtual harddisk image file.
Parameter:

- type: VM_SW_KVM for KVM or VM_SW_VBOX for VirtualBox OSE
- vmname: Name of the VM.
- diskname: Name of the image file without extension.
- size: Size of the image file in MB.

Returns:

- BASH code to create a virtual disk image.

8.38.56 VM_delVMCMD

Description: Deletes a virtual machine.
Parameter:

- type: VM_SW_VBOX for VirtualBox OSE
- vmname: Name of the VM.

Returns:

- BASH code to delete a virtual machine.

8.38.57 VM_activateNetbootCMD

Description: Generates a BASH command line to (de)activate network booting of a VM.
Parameter:

- type: VM_SW_VBOX for VirtualBox OSE
- vmName: Name of the VM.
- activate: true for booting from network, false for booting from the HD.

Returns:

- BASH code to delete a virtual machine.

8.38.58 VM_createVM

Description: Creates a virtual machine.
Parameter:

- type: VM_SW_VBOX for VirtualBox OSE
- vmName: Name of the VM.
- ramSize: Size of the memory in MB.
- diskName: Name of the virtual harddisk file.
- mac: MAC address of the virtual network card. It can be in the format 12:23:34:45:56:78 or 122334455678.
- netDev: Device of the real network card that is used to let the VM communictae with the outer world.

Returns:

- BASH code to create a virtual machine.

8.38.59 VM_insertBootISO

Description: Inserts a bootable ISO into a VM.
Parameter:

- type: VM_SW_VBOX for VirtualBox OSE
- vmName: Name of the VM.
- iso: ISO file with full path.

Returns:

- BASH code to insert a bootable ISO into a VM.

8.38.60 VM_startVMInExistingXSession

Description: Starts a virtual machine in an existing X session.
Parameter:

- type: VM_SW_VBOX for VirtualBox OSE
- vmName: Name of the VM.

Returns:

- BASH code to start a virtual machine and finding the DISPLAY number of the user who runs this script.

8.38.61 VM_startVM

Description: Starts a virtual machine.
Parameter:

- type: VM_SW_VBOX for VirtualBox OSE
- vmName: Name of the VM.
- vnc: Set to true if the VM should be accessible since the booting via VNC.

Returns:

- BASH code to start a virtual machine.

8.38.62 VM_startVMCommandFile

Description: Writes a command file with the command(s) to start the VM.
Parameter:

- vmName: Name of the VM.
- cmd: Bash code to start the VM.

8.38.63 VM_stopVMCommandFile

Description: Removes automatical staring of a VM by removing the command file.
Parameter:

- vmName: Name of the VM.

8.38.64 VM_status

Description: Gets the current status of a virtual machine.
Parameter:

- type: VM_SW_VBOX for VirtualBox OSE
- vmName: Name of the VM.

Returns:

- BASH code to get the current status of a virtual machine or array containing the status of the VM.

8.38.65 VM_parseVBOXdisk

Description: Parses a harddisk/DVD/floppy status line of VirtualBox.
Parameter:

- param: Parameter line that may contain the complete path to the image file or "empty".

Returns:

- Name of the assigned image or false of the medium is empty.

8.38.66 VM_parseVBOXstate

Description: Parses the status (on, off, paused) line of VirtualBox.
Parameter:

- param: Parameter line that contains the status string of the VM.

Returns:

- VM_STATE_OFF, VM_STATE_PAUSE, VM_STATE_ON or false if the line could not be parsed.

8.38.67 VM_parseVBOXNic

Description: Parses the status line of a virtual network card.
Parameter:

- param: Parameter line that contains the status string of the VM.

Returns:

- Array with the current state of the network device.

8.38.68 VM_parseStatus

Description: Parses the complete status of a VM.
Returns:

- Array with the current state of the VM.