m23 software distribution

m23 Development Guide
for m23 rock 21.1

Hauke Goos-Habermann

August 13, 2021

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
Copyright © 1989, 1991 Free Software Foundation, Inc.

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

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

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

m23 helpers

3.1 mdoc

3.1.1 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
```

3.1.2 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.

3.1.3 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.
$*/
```

3.1.4 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
**/
```

3.1.5 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

3.1.6 example

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

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

Chapter 4

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.

4.1 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.

4.1.1 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"

4.1.2 End position of a patchable area

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

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

shadecolorGray

4.2 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.

4.2.1 Example (logo.php.m23customPatch)

shadecolorGray

4.3 /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".

4.3.1 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

4.4 Applying patches on m23 update

After an update of the m23 software, the patches need to be re-applied again. To automatise this step, you can place a BASH script with the needed calls to m23cutomPatch under /m23/bin/postinstHook.sh. This script will be called when the m23 package is configured. This happens during installation or during update.

Chapter 5

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.

5.1 ./data+scripts/packages/m23CommonInstallRoutines.php

Author: Hauke Goos-Habermann (HHabermann@pc-kiel.de)
Description: Installatio routines shared among distributions.

5.1.1 CIR_stopHaveged

Description: Kills haveged to make it not consume much CPU ressources.

5.1.2 CIR_rootInRamdiskOrExit

Description: Checks, if root is stored on the ramdisk and exits the script, if not.

5.1.3 CIR_setDateAndTimeTemorarily

Description: Sets the date and time by calling 'date' temporarily.

5.1.4 CIR_WorkaroundForMissingModulesDep

Description: Workaround for missing modules.dep to disable the repeated showing of the "FATAL" error.

5.1.5 CIR_transferClientIP

Description: Transfers the current IP of a m23shared client to the m23 server.

5.1.6 CIR_writeClientID

Description: writes the client ID to /m23clientID
Parameter:

- clientParams: Associated array with the parameters of the client.

5.1.7 CIR_detectSCSI

Description: detects SCSI controlers and loads the modules

5.1.8 CIR_waitForNextJob

Description: waits one minute and tries to fetch the next job from the server and executes it

5.1.9 CIR_enableDropbear

Description: sets up and starts dropbear SSH server

5.2 ./inc/assimilate.php

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

5.2.1 ASSI_showClientAddDialog

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

5.2.2 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

5.2.3 ASSI_addUbuntuRoot

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

5.2.4 ASSI_prepareClient

Description: Prepares a client for assimilisation.

5.3 ./inc/autoTest.php

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

5.3.1 AUTOTEST_createLDAPUserAndGroup

Description: Creates an LDAP user and group on the local LDAD server and shows an info message, if creation was sucessfully or a warning with the direct message from the LDAP server.

5.3.2 AUTOTEST_isDebug

Description: Checks, if autoTest is in debug mode (constant AT_debug is set).
Returns:

- true, if autoTest is in debug mode, otherwise false.

5.3.3 AUTOTEST_sshVMServer

Description: Executes commands on the virtualisation server.
Parameter:

- cmds: The commands to execute on the virtualisation server.

Returns:

- The output of the commands.

5.3.4 AUTOTEST_sshTunnelOverServer

Description: Executes commands on a client indirectly, that is reachable over an m23 server only. SSH connections require root user on the m23 server and the client.
Parameter:

- serverNameOrIP: The name of the m23 server or its IP.
- clientNameOrIP: The name of the client or its IP.
- cmds: The commands to execute on the client
- password: Parameter to set the SSH password or NULL to use SSH keys only.

Returns:

- The output of the client's screen.

5.3.5 AUTOTEST_getSeleniumSafeString

Description: Replaces characters in a string to make it safe for usage in Selenium and SimpleXMLElement.
Parameter:

- in: Input string. eg. "&page=fdisk" becomes "

Returns:

- Safe made string.

5.3.6 AUTOTEST_getSeleniumSafeString

Description: Urldecodes characters that were made it safe for usage in Selenium and SimpleXMLElement.
Parameter:

- in: Input string. eg. "

Returns:

- Unsafe made string.

5.3.7 AUTOTEST_VM_hostSanityCheck

Description: Checks the existence of tools on the VM host.
Returns:

- true or dies with an error message.

5.3.8 AUTOTEST_replaceConstantsInString

Description: Searches for variable declarations in a string and replaces them with the values of according constants.
Parameter:

- in: Input string. eg. "name=$VM_NAME" may become "name=debianVM"

Returns:

- The input string with replaced variable declarations.

5.3.9 AUTOTEST_SEL_getURLByMatch

Description: Searches for one URL in the given HTML code that must contain all search terms.
Parameter:

- htmlsource: The HTML code.
- searchA: Array with the search terms.

Returns:

- The one matching URL or NULL if there is none matching.

5.3.10 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.

5.3.11 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.

5.3.12 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)

5.3.13 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.

5.3.14 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.

5.3.15 AUTOTEST_VM_delete

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

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

5.3.16 AUTOTEST_VM_export_m23ServerISO_as_OVA

Description: Exports a VM, that was installed via the m23 server installation ISO, to OVA file.
Parameter:

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

5.3.17 AUTOTEST_VM_stop

Description: Stops a VM.
Parameter:

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

5.3.18 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.

5.3.19 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.

5.3.20 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.

5.3.21 AUTOTEST_VM_restoreSnapshot

Description: Stops a VM and restores a snapshot.
Parameter:

- vmName: Name of the VM.
- snapshotName: Name of the snapshot to restore.

5.3.22 AUTOTEST_executePHPFunction

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

- vmName: Name of the VM.
- params: Parameter string with function name as 1st part and its parameters concenated by "°".

5.3.23 AUTOTEST_VM_keyboardWrite

Description: Emulates the keystrokes into a VM.
Parameter:

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

5.3.24 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.

5.3.25 AUTOTEST_VM_screenPixelDiff

Description: Compares the VM's screen with a previously saved screen and gives back the amount of changed pixels.
Parameter:

- vmName: Name of the VM.

Returns:

- Amount of changed pixels (or 99999999999 if there is no previously saved screen).

5.3.26 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.

5.3.27 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.

5.3.28 AUTOTEST_SELENIUM_exec

Description: Runs a command on the HTTP2SeleniumBridge.
Parameter:

- bridgeURL: URL to the HTTP2SeleniumBridge server (eg. http://192.168.1.153:23080)
- cmd: The command to execute (eg. typeInto)
- ID or name: ID or name of the HTML element to
- additional parameters: Additional parameters with variable name first, value second.

Returns:

- HTTP output of the HTTP2SeleniumBridge call.

5.3.29 AUTOTEST_SELENIUM_open

Description: Opens a website on the HTTP2SeleniumBridge.
Parameter:

- url: The URL to open (with "http(s)://" at the beginning and possible "<user>:<password>@"

Returns:

- HTTP output of the HTTP2SeleniumBridge call.

5.3.30 AUTOTEST_SELENIUM_selectFromID

Description: Selects an option from a selection choosen by selection ID.
Parameter:

- ID: ID of the selection
- val: The value of the option to select

Returns:

- HTTP output of the HTTP2SeleniumBridge call.

5.3.31 AUTOTEST_SELENIUM_selectFromName

Description: Selects an option from a selection choosen by selection name.
Parameter:

- name: Name of the selection
- val: The value of the option to select

Returns:

- HTTP output of the HTTP2SeleniumBridge call.

5.3.32 AUTOTEST_SELENIUM_deselectFromID

Description: Deselects an option from a selection choosen by selection ID.
Parameter:

- ID: ID of the selection
- val: The value of the option to deselect

Returns:

- HTTP output of the HTTP2SeleniumBridge call.

5.3.33 AUTOTEST_SELENIUM_deselectFromName

Description: Deselects an option from a selection choosen by selection name.
Parameter:

- name: Name of the selection
- val: The value of the option to deselect

Returns:

- HTTP output of the HTTP2SeleniumBridge call.

5.3.34 AUTOTEST_SELENIUM_clickButtonID

Description: Clicks a button choosen by button ID.
Parameter:

- ID: ID of the button

Returns:

- HTTP output of the HTTP2SeleniumBridge call.

5.3.35 AUTOTEST_SELENIUM_clickButtonName

Description: Clicks a button choosen by button name.
Parameter:

- name: Name of the button

Returns:

- HTTP output of the HTTP2SeleniumBridge call.

5.3.36 AUTOTEST_SELENIUM_typeIntoID

Description: Types into an editline or textarea choosen by its ID.
Parameter:

- ID: ID of the editline or textarea

Returns:

- HTTP output of the HTTP2SeleniumBridge call.

5.3.37 AUTOTEST_SELENIUM_typeIntoName

Description: Types into an editline or textarea choosen by its name.
Parameter:

- name: Name of the editline or textarea

Returns:

- HTTP output of the HTTP2SeleniumBridge call.

5.3.38 AUTOTEST_SELENIUM_setCheckID

Description: (Un)Ticks a checkbox choosen by its ID.
Parameter:

- ID: ID of the checkbox
- checked: true for activating the checkbox, false for unticking it

Returns:

- HTTP output of the HTTP2SeleniumBridge call.

5.3.39 AUTOTEST_SELENIUM_setCheckName

Description: (Un)Ticks a checkbox choosen by its name.
Parameter:

- name: Name of the checkbox
- checked: true for activating the checkbox, false for unticking it

Returns:

- HTTP output of the HTTP2SeleniumBridge call.

5.3.40 AUTOTEST_SELENIUM_selectRadioID

Description: Selects a radio button choosen by its ID.
Parameter:

- ID: ID of the radio button
- val: The value of the radio button to select

Returns:

- HTTP output of the HTTP2SeleniumBridge call.

5.3.41 AUTOTEST_SELENIUM_selectRadioName

Description: Selects a radio button choosen by its name.
Parameter:

- name: Name of the radio button
- val: The value of the radio button to select

Returns:

- HTTP output of the HTTP2SeleniumBridge call.

5.3.42 AUTOTEST_SELENIUM_getsource

Description: Gets the current page source of the selenium browser into the output page.
Returns:

- Current page source

5.3.43 AUTOTEST_SELENIUM_close

Description: Closes the browser window
Returns:

- HTTP output of the HTTP2SeleniumBridge call.

5.3.44 AUTOTEST_SELENIUM_quit

Description: Closes the browser and webserver and quits HTTP2SeleniumBridge.
Returns:

- HTTP output of the HTTP2SeleniumBridge call.

5.4 ./inc/backup.php

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

5.4.1 BACKUP_showClientSettings

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

- client: name of the client

5.4.2 BACKUP_getBackupDirs

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

- client: name of the client

5.4.3 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

5.4.4 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.

5.4.5 BACKUP_addAdmin

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

- admin: Name of the admin.

5.4.6 BACKUP_delAdmin

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

- admin: Name of the admin.

5.5 ./inc/bittorrent.php

Author: Hauke Goos-Habermann (HHabermann@pc-kiel.de)
Description: Funtions for distributing files via BitTorrent.

5.5.1 BT_dlFile

Description: Starts a Bittorrent download.
Parameter:

- torrentFile: Name of the torrent file under BT_DIR.
- dest: Path on the client where to

Returns:

- : true, when the torrent file is present otherwise false.

5.5.2 BT_status

Description: Shows status information about the (maybe) running Bittorrent tracker and initial client.
Parameter:

- return: If set to true, the result will be returned otherwise shown.
- nl2br: If set to true, the ASCII line breaks will be converted to HTML line breaks.

Returns:

- : Status information, if $return == true.

5.5.3 BT_checkSoftware

Description: Checks, if a Bittorrent software (client + tracker) is installed.
Parameter:

- user: user the command should be run under.

Returns:

- : true when the command is available otherwise false.

5.5.4 BT_stopService

Description: Stops a Bittorrent service.
Parameter:

- job: Name of the job.

Returns:

- : true, when the service could be stoped otherwise false.

5.5.5 BT_startService

Description: Starts a Bittorrent service.
Parameter:

- job: Name of the job.
- cmds: BASH commands to execute.

Returns:

- : true, when the service could did not run before otherwise false.

5.5.6 BT_stopTracker

Description: Stops the Bittorrent tracker.

5.5.7 BT_startTracker

Description: Starts the tracker.

5.5.8 BT_restartTracker

Description: Restarts the tracker.

5.5.9 BT_autostart

Description: Starts Bittorrent tracker and initial client when there are .torrent files in the share directory or stops when there are none.

5.5.10 BT_updateWhitelist

Description: Updates the white list with all allowed torrent files on the tracker.

5.5.11 BT_createTorrent

Description: Creates a torrent file and adds it to the white list.
Parameter:

- fileToShare: The file (in the Bittorrent share directory) to create a torrent file for.

Returns:

- : true, when the torrent file was created successfully otherwise false.

5.5.12 BT_startClient

Description: Starts the Bittorrent client.

5.5.13 BT_stopClient

Description: Stops the Bittorrent client.

5.5.14 BT_restartClient

Description: Restarts the client.

5.6 ./inc/burn.php

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

5.6.1 BURN_listBurners

Description: returnes a selection of the available burners
Parameter:

- first: entry that should be shown first

5.6.2 BURN_getDevice

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

- burner: device name and burner name

5.6.3 BURN_blank

Description: blanks a CD-RW
Parameter:

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

5.6.4 BURN_burn

Description: burns an ISO
Parameter:

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

5.6.5 BURN_getStatus

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

5.6.6 BURN_showLog

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

5.6.7 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).

5.6.8 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.

5.7 ./inc/capture.php

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

5.7.1 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

5.7.2 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

5.7.3 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

5.7.4 CAPTURE_deActivate

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

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

5.7.5 CAPTURE_isActive

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

5.7.6 CAPTURE_captureImg

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

5.7.7 CAPTURE_toggle

Description: toggles the current capture state

5.7.8 CAPTURE_showMessageBox

Description: shows a message box, if capturing is enabled

5.7.9 CAPTURE_showEntries

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

5.7.10 CAPTURE_deleteById

Description: deletes a capture entry.
Parameter:

- id: the id of the capture entry to delete

5.7.11 CAPTURE_showMarker

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

5.7.12 CAPTURE_showTableWith

Description: Adds a width element if in captureLoad mode.

5.8 ./inc/CAutoTest.php

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

5.8.1 CAutoTest::__construct

Description: Constructor for new CAutoTest objects.
Parameter:

- xmlFile: Name of the XML file to open.
- argv: Arguments from the command line (without the script name as 0th element) as array

5.8.2 CAutoTest::__destruct

Description: Destructor for this CAutoTest object.

5.8.3 CAutoTest::internalVariablesToConstants

Description: Maps internal variables to constants (for backward compatibility).

5.8.4 CAutoTest::getSeleniumVMIP

Description: Gets the IP of the Selenium VM
Returns:

- IP of the Selenium VM

5.8.5 CAutoTest::isSeleniumVMOnline

Description: Is the Selenium VM online (pingable)
Returns:

- true, if the Selenium VM is online, false otherwise.

5.8.6 CAutoTest::startSeleniumVM

Description: Starts the Selenium VM, if it's not running.

5.8.7 CAutoTest::importEnvironmentVariables

Description: Imports environment variables starting with 'AT_' into the runtime variables array.

5.8.8 CAutoTest::loadXMLFile

Description: Loads an XML file, replaces problematic characters and includes other files.
Parameter:

- xmlFile: Name of the XML file to open.

Returns:

- SimpleXMLElement object with the loaded XML file.

5.8.9 CAutoTest::readSettings

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

5.8.10 CAutoTest::internalValuesFromXML

Description: Reads the variables section of an XML file and imports the variables into the array with runtime variables.
Parameter:

- xmlO: Prased XML structure pointing to the trigger.

5.8.11 CAutoTest::VMCreate

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

5.8.12 CAutoTest::VMStop

Description: Stops a virtual machine.

5.8.13 CAutoTest::VMStart

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

5.8.14 CAutoTest::VMRestoreSnapshot

Description: Stops a VM and restores a snapshot.
Parameter:

- snapshotName: Name of the snapshot to restore.

5.8.15 CAutoTest::VMexportm23ServerISOasOVA

Description: Exports a VM, that was installed via the m23 server installation ISO, to OVA file.

5.8.16 CAutoTest::getISO

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

- Filename and path to the ISO image for booting or NULL, if no ISO is used.

5.8.17 CAutoTest::isVM

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

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

5.8.18 CAutoTest::isXMLTEST

Description: Check, if the only the XML should be parsed.
Returns:

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

5.8.19 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.

5.8.20 CAutoTest::getLogFileName

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

- Log file name.

5.8.21 CAutoTest::addToLogFile

Description: Appends lines to the log file.
Parameter:

- lines: The lines to add.

5.8.22 CAutoTest::getTimestampString

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

- Date and time in human readable form.

5.8.23 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.

5.8.24 CAutoTest::isTriggered

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

- true, when the current sequence event was triggered.

5.8.25 CAutoTest::setTimeout

Description: Sets the timeout for the current sequence element.

5.8.26 CAutoTest::setVMScreenChangeIntervall

Description: Sets the time to wait (in seconds) between making two screenshots and checking for changes.

5.8.27 CAutoTest::checkVMScreenChange

Description: Makes another screenshot and compares it with the previous, if the waiting time is over. In case that too less pixels have been changed a warning is logged and shown.

5.8.28 CAutoTest::decTimeout

Description: Decrements the remaining time for the timeout.

5.8.29 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).
- execAttributes: Additional attributes (parameters) (eg. name/ID for Selenium calls)
- timeout: Time to wait (in seconds) until the element of sequence will become a failure.
- description: Description for the test.
- triggerAttributes: Trigger attribute(s). Can hold additional parameters. (Result is written to this pointer)
- runIf: Value of the runIf attribute.
- vmScreenChangeIntervall: Time to wait (in seconds) between making two screenshots and checking for changes.

5.8.30 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.

5.8.31 CAutoTest::getAnswersA

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

- Array with the answers of the current sequence element.

5.8.32 CAutoTest::getTriggerTypes

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

- Trigger type of the current sequence element.

5.8.33 CAutoTest::getTriggerAttributes

Description: Returns the attributes (additional parameters) for execution of the current sequence element.
Returns:

- Attributes (additional parameters) for execution of the current sequence element.

5.8.34 CAutoTest::getTriggerParams

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

- Trigger parameter of the current sequence element.

5.8.35 CAutoTest::getExecTypes

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

- Type of execution of the current sequence element.

5.8.36 CAutoTest::getExecParams

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

- Parameter for execution of the current sequence element.

5.8.37 CAutoTest::getExecAttributes

Description: Returns the attributes (additional parameters) for execution of the current sequence element.
Returns:

- Attributes (additional parameters) for execution of the current sequence element.

5.8.38 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.

5.8.39 CAutoTest::isAnswerFoundInTriggerResult

Description: Checks, if a trigger result contains an answer.
Parameter:

- triggerResult: A text (eg. extracted by OCR or from HTML source) that may contain answers.
- answer: A normal string or an $I18N_ variable. In case of a variable all translations will be searched in the trigger result. If it begins with "!", the answer should NOT be found in the result.

Returns:

- true, if the answer (or its translation) was (NOT) found in the trigger result.

5.8.40 CAutoTest::executePHPFunction

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

- params: Parameter string with function name as 1st part and its parameters concenated by "°".

5.8.41 CAutoTest::executeTriggerAction

Description: Executes the action of the current sequence element.

5.8.42 CAutoTest::waitForTrigger

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

5.8.43 CAutoTest::getCurElement

Description: Returns the current sequence element.
Returns:

- Trigger type of the current sequence element.

5.8.44 CAutoTest::nextCurElement

Description: Increments the current sequence element number.

5.8.45 CAutoTest::isConsecutiveIdenticalMessage

Description: Checks, if an identical message was assigned in the last call.
Parameter:

- msg: Message to check.

Returns:

- true, if an identical message was assigned in the last call, otherwise false.

5.8.46 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.
- ignoreConsecutiveIdenticalMessages: If set to true, consecutive identical messages are not given shown or logged, but

5.8.47 CAutoTest::debugPrint

Description: Shows a message (, array or object in human readable form) and logs it to the log file, if autoTest is in debug mode.
Parameter:

- msg: Message (, array or object in human readable form) to show in the console and the log file.
- prefix: A prefix show before the message to indicate the type of the message.

5.8.48 CAutoTest::isDebug

Description: Checks, if autoTest is in debug mode (variable AT_debug is set).
Returns:

- true, if autoTest is in debug mode (variable AT_debug is set), otherwise false.

5.8.49 CAutoTest::getStatusDescriptionInBraces

Description: Returns a status string with optional description in braces.
Parameter:

- status: The status.
- description: Optional description.

Returns:

- Status string with optional description in braces.

5.8.50 CAutoTest::evaluateRunIf

Description: Checks, if the condition of the runIf attribute is met.
Returns:

- true, if the condition of the runIf attribute is met, otherwise false.

5.8.51 CAutoTest::evaluateSetVar

Description: Parses the value of an "setVar" attribute from an aarray with information about the answer and sets it to the runtime variables array, if it seems to be valid.

5.8.52 CAutoTest::elemGood

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

- answer: Array with information about the answer.

5.8.53 CAutoTest::elemWarn

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

- answer: Array with information about the answer.

5.8.54 CAutoTest::elemGoodOrWarn

Description: The current sequence elements was finished sucessfully or with a warning. The result is shown and written 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.

5.8.55 CAutoTest::elemBad

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

- answer: Array with information about the answer.

5.8.56 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.

5.8.57 CAutoTest::triggerTypeToConstant

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

- type: Trigger type (string).

Returns:

- Trigger type constant.

5.8.58 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)
- testTriggerAttributes: Trigger attribute(s). Can hold additional parameters. (Result is written to this pointer)

5.8.59 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(s). (Result is written to this pointer)
- testActionAttributes: Action attribute(s). Can hold additional parameters. (Result is written to this pointer)

5.8.60 CAutoTest::seleniumReload

Description: Reloads the currently loaded page
Returns:

- Current page source

5.8.61 CAutoTest::seleniumGetSource

Description: Gets the current page source of the selenium browser into the output page.
Returns:

- Current page source

5.8.62 CAutoTest::seleniumExec

Description: Runs a command on the HTTP2SeleniumBridge.
Parameter:

- args: Array with the arguments.

Returns:

- HTTP output of the HTTP2SeleniumBridge call.

5.8.63 CAutoTest::seleniumExecExtra

Description: Runs an action on the HTTP2SeleniumBridge.
Parameter:

- action: Name of the action (eg. nextdriverid or free)
- args: Array with the arguments.

Returns:

- HTTP output of the HTTP2SeleniumBridge call.

5.8.64 CAutoTest::gotSeleniumDriverID

Description: Returnes true, if a Selenium webdriver ID was acquired.
Returns:

- true, if a Selenium webdriver ID was acquired, otherwise false.

5.8.65 CAutoTest::getSeleniumDriverID

Description: Gives the Selenium webdriver ID this instance should use or let the programm die, if all are used.
Returns:

- Selenium webdriver ID.

5.8.66 CAutoTest::freeSeleniumDriverID

Description: Frees the used Selenium webdriver ID.

5.8.67 CAutoTest::seleniumGetStatus

Description: Gets the status message(s) of the HTTP2SeleniumBridge.
Returns:

- Status message(s) of the HTTP2SeleniumBridge call.

5.8.68 CAutoTest::seleniumHostRunning

Description: Checks, if the HTTP2SeleniumBridge is running.
Returns:

- true, if the HTTP2SeleniumBridge is running, otherwise false.

5.8.69 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).

5.8.70 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.

5.9 ./inc/CAutoUpdate.php

Author: Hauke Goos-Habermann (HHabermann@pc-kiel.de)
Description: Class for handling automatic updates.

5.9.1 CAutoUpdate::getActiveSetID

Description: Gets the ID of the active set or NULL, if none is active or no sets are present.
Returns:

- : ID of the active set or NULL, if none is active or no sets are present.

5.9.2 CAutoUpdate::getActiveSet

Description: Gets the active set or false, if none is active or no sets are present.
Returns:

- : Active set or false, if none is active or no sets are present.

5.9.3 CAutoUpdate::addSet

Description: Adds an entry to the schedule.
Parameter:

- startsEnds: Array with the start and end times
- groups: Array with the groups.
- type: Type of update (CAutoUpdate::UPDATE_FULL or CAutoUpdate::UPDATE_NORMAL)
- parallelUpdates: Amount of clients that should be updated in parallel.

Returns:

- : true, on successfully insert, otherwise false.

5.9.4 CAutoUpdate::setParallelUpdates

Description: Sets the amount of parallel updates for a set.
Parameter:

- id: ID of the set.
- amount: The amount of parallel updates for a set

5.9.5 CAutoUpdate::getParallelUpdates

Description: Gets the amount of parallel updates in a set.
Parameter:

- id: ID of the set.

Returns:

- Amount of parallel updates in a set or 0, if no type is set or it is invalid.

5.9.6 CAutoUpdate::setType

Description: Sets the type for a set.
Parameter:

- id: ID of the set.
- type: Update type.

5.9.7 CAutoUpdate::getType

Description: Gets the type of a set as constant value or in translated form.
Parameter:

- id: ID of the set.
- translate: If set to true, the type will be translated to a human readable form.

Returns:

- Type of a set as constant value or in translated form or false, if no type is set or it is invalid.

5.9.8 CAutoUpdate::getUpdateTypeTranslation

Description: Returns the whole translation array with update type as key and the translation as value or the translation for a given constant.
Parameter:

- const: If set to a constant, the translation will be returned, otherwise the whole array.

Returns:

- : Translation array, translation for a given constant or false, if the constant is not valid.

5.9.9 CAutoUpdate::delSet

Description: Deletes one set.
Parameter:

- id: ID of the set to delete.

Returns:

- : true, of the set with the given ID could be deleted, otherwise dies.

5.9.10 CAutoUpdate::getAllSets

Description: Generates an array with all sets and the according information for each set.
Parameter:

- orderBy: Rule for ordering the entries in the ourput array (CAutoUpdate::ORDERBY_ID or CAutoUpdate::ORDERBY_LASTRUN)

Returns:

- : Array with all sets and the according information for each set.

5.9.11 CAutoUpdate::loadSets

Description: Loads all range sets into an array with.
Parameter:

- orderBy: Rule for ordering the entries in the ourput array (CAutoUpdate::ORDERBY_ID or CAutoUpdate::ORDERBY_LASTRUN)

Returns:

- : Array with all sets and the according information for each set.

5.9.12 CAutoUpdate::saveSets

Description: Saves all changed sets to the DB.
Returns:

- : true, if a set is active.

5.9.13 CAutoUpdate::getSetAmount

Description: Get the amount of update sets.
Returns:

- : Amount of update sets.

5.9.14 CAutoUpdate::configureCron

Description: Adds and removes the crontab entry dynamically when the first update set gets created or the last gets deleted.

5.9.15 CAutoUpdate::isASetActive

Description: Checks, if a set is active.
Returns:

- : true, if a set is active.

5.9.16 CAutoUpdate::activateSet

Description: Activates one set and deactivates all other sets.
Parameter:

- id: ID of the set to activate.

Returns:

- : true, of the set with the given ID could be activated, otherwise dies.

5.9.17 CAutoUpdate::activateNextSet

Description: Activates the next set with the oldest timestamp.
Returns:

- : ID of the active set or NULL, if no set could be activated.

5.9.18 CAutoUpdate::activateNextTimlySet

Description: Activates the next set with the oldest timestamp that should be executed now.
Returns:

- : ID of the active set or NULL, if no set could be activated.

5.9.19 CAutoUpdate::addStartEnd

Description: Adds a start end range for running the updates.
Parameter:

- id: ID of the set to add the time range.
- start: Start time of the range as combined numeric day and hour/minute string.

- End: End time of the range as combined numeric day and hour/minute string.

Returns:

- true, if the time range could be added, otherwise false.

5.9.20 CAutoUpdate::delStartEnd

Description: Deletes a start end range.
Parameter:

- id: ID of the set to remove the time range.
- start: Start time of the range as combined numeric day and hour/minute string.

5.9.21 CAutoUpdate::getAllStartsEndsHumanReadable

Description: Generate a list of all time ranges (with optional button for deleting an entry).
Parameter:

- id: ID of the set to get all time ranges.
- addButtons: If set to true, each range will get a deletion button.
- defaultSeparator: The (HTML) string to separate the entries.

Returns:

- List of all time ranges (with optional button for deleting an entry).

5.9.22 CAutoUpdate::addGroup

Description: Adds a client group for running the updates on them.
Parameter:

- id: ID of the set to add the group.
- group: Name of the client group.

Returns:

- true, if the group could be added, otherwise false.

5.9.23 CAutoUpdate::delGroup

Description: Deletes a group.
Parameter:

- id: ID of the set to remove the group from.
- group: Name of the client group.

5.9.24 CAutoUpdate::getAllGroupsHumanReadable

Description: Generate a list of all groups (with optional button for deleting an entry).
Parameter:

- id: ID of the set to get all groups.
- addButtons: If set to true, each group will get a deletion button.
- defaultSeparator: The (HTML) string to separate the entries.

Returns:

- List of all groups (with optional button for deleting an entry).

5.9.25 CAutoUpdate::getImgButtonParams

Description: Checks, if an image button was pressed and extracts the two parameters from it.
Parameter:

- reg: Regular expression to find the HTML names.
- a: Variable to write the value of the first parameter to.
- b: Variable to write the value of the second parameter to.

Returns:

- true, if a button mathing the regex was clicked, otherwise false.

5.9.26 CAutoUpdate::isTimeToRunNow

Description: Checks, if the given set should be run now.
Parameter:

- id: ID of the set

Returns:

- true, if the given set should be run now, otherwise false.

5.9.27 CAutoUpdate::getActiveGroups

Description: Gets the groups from the current active set.
Returns:

- The groups from the current active set or false, if no set is active.

5.9.28 CAutoUpdate::getAllClientnames

Description: Gets the clients in all groups of the active set.
Parameter:

- running: If set to true, only clients running auto update will be listed, otherwise all clients.

Returns:

- Array with the clients in all groups of the active set or empty array in case of an error.

5.9.29 CAutoUpdate::startNewClients

Description: Starts auto update on a given amount of clients that haven't run auto update for at least one day.
Parameter:

- clientAmount: Amount of clients to start.

Returns:

- Amount of new started clients.

5.9.30 CAutoUpdate::run

Description: Chooses the set that should be run now and starts new auto update on clients.

5.9.31 CAutoUpdate::showAutoUpdateManager

Description: Shows a dialog for viewing, creating, changing and deleting automatic update sets.

5.10 ./inc/CChecks.php

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

5.10.1 CChecks::checkSizeInMB

Description: Checks if a size in MB is valid.
Parameter:

- size: Size to check.

Returns:

- true, if the size is valid (numeric) otherwise false.

5.10.2 CChecks::checkm23AdminName

Description: Checks if an m23 admin name is valid.
Parameter:

- name: Login name of the m23 admin

Returns:

- true, if the name is valid otherwise false.

5.10.3 CChecks::checkDiskDefinedSize

Description: Checks if a size (in MB) for the defined disk is valid.
Parameter:

- size: Size to check.

Returns:

- true, if the size is valid (numeric) otherwise false.

5.10.4 CChecks::checkFdiskAdjustmentUpperToleranceIdentical

Description: Checks if a size (may contain g/G for GB, m/M for MB or **Parameter:**

- size: Size to check.

Returns:

- true, if the size is valid otherwise false.

5.10.5 CChecks::checkFdiskAdjustmentLowerToleranceIdentical

Description: Checks if a size (may contain g/G for GB, m/M for MB or **Parameter:**

- size: Size to check.

Returns:

- true, if the size is valid otherwise false.

5.10.6 CChecks::checkSwapPart

Description: Checks if the device name for the swap partition is valid.
Parameter:

- swapPart: Swap partition to check.

Returns:

- true, if the swap partition name is valid otherwise false.

5.10.7 CChecks::checkEFIPart

Description: Checks if the device name for the EFI partition is valid.
Parameter:

- EFIPart: EFI partition to check.

Returns:

- true, if the EFI partition name is valid otherwise false.

5.10.8 CChecks::checkInstPart

Description: Checks if the device name for the installation partition is valid.
Parameter:

- instPart: Installation partition to check.

Returns:

- true, if the installation partition name is valid otherwise false.

5.10.9 CChecks::checkMountDev

Description: Checks if the device name for mounting a disk or partition is valid.
Parameter:

- dev: Device name for disk or partition to check.

Returns:

- true, if the device name for mounting a disk or partition is valid otherwise false.

5.10.10 CChecks::checkMountPoint

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.

5.10.11 CChecks::checkFdiskAdjustmentSpecifiedDev

Description: Checks if the device name for the disk of the defined client is valid.
Parameter:

- dev: Device name to check.

Returns:

- true, if the installation partition name is valid otherwise false.

5.10.12 CChecks::checkRelease

Description: Checks if a release name is valid.
Parameter:

- release: Release name to check.

Returns:

- true, if the release name is valid otherwise false.

5.10.13 CChecks::checkDistribution

Description: Checks if a distribution name is valid.
Parameter:

- distr: Distribution name to check.

Returns:

- true, if the distribution name is valid otherwise false.

5.10.14 CChecks::checkNfshomeserver

Description: Checks if the NFS share is valid.
Parameter:

- nfshomeserver: NFS share name to check.

Returns:

- true, if the NFS share is valid otherwise false.

5.10.15 CChecks::checkGroupname

Description: Checks if the groupname is valid.
Parameter:

- group: Groupname to check.

Returns:

- true, if the groupname is valid otherwise false.

5.10.16 CChecks::checkUserGroupIDs

Description: Checks the user ID and group ID are valid.
Parameter:

- userID: The user ID to check.
- groupID: The group ID to check.

Returns:

- true, if user ID and group ID are valid otherwise false.

5.10.17 CChecks::checkLanguage

Description: Checks if the language is valid.
Parameter:

- language: Language value to check.

Returns:

- true, if the language is valid otherwise false.

5.10.18 CChecks::checkRootpassword

Description: Checks if the root password is valid.
Parameter:

- rootpassword: The password to check.

Returns:

- true, if the root password is valid otherwise false.

5.10.19 CChecks::firstpw

Description: Checks if the first name's password is valid.
Parameter:

- pass: The password to check.
- allowEmpty: Set to true, if empty passwords should be allowed (e.g. when read from an LDAP server)

Returns:

- true, if the password is valid otherwise false.

5.10.20 CChecks::checkLogin

Description: Checks if the given login is valid.
Parameter:

- login: The login name to check.
- allowEmpty: Set to true, if empty logins should be allowed (e.g. when read from an LDAP server)

Returns:

- true, if the login name is valid otherwise false.

5.10.21 CChecks::checkProxy

Description: Checks the IP and port of the package proxy.
Parameter:

- proxyIP: The IP of the package proxy.
- proxyPort: The port of the package proxy.

Returns:

- true, if the package IP and port are valid otherwise false.

5.10.22 CChecks::checkOffice

Description: Checks if the given office name is valid.
Parameter:

- office: The office name to check.

Returns:

- true, if the office name is valid otherwise false.

5.10.23 CChecks::checkEmail

Description: Checks if the given eMail is valid (or optionally empty).
Parameter:

- email: eMail address to check.
- allowEmpty: Set to true, if empty eMail addresses should accepted as valid.

Returns:

- true, if the eMail address is valid (or empty) otherwise false.

5.10.24 CChecks::checkFamilyname

Description: Checks if the user's familyname is valid.
Parameter:

- familyname: The familyname of the user.

Returns:

- true, if the familyname name is valid otherwise false.

5.10.25 CChecks::checkForename

Description: Checks if the user's forename is valid.
Parameter:

- forename: The forename of the user.

Returns:

- true, if the forename name is valid otherwise false.

5.10.26 CChecks::checkIPGeneric

Description: Checks if an IP is valid and adds an error message to the message manager in case of an error.
Parameter:

- ip: IP to check.
- msg: Error message to add in case of an error.

Returns:

- true, if the IP is correct otherwise false.

5.10.27 CChecks::checkIP

Description: Checks if an IP is valid and adds an error message to the message manager in case of an error.
Parameter:

- ip: IP to check.

Returns:

- true, if the IP is correct otherwise false.

5.10.28 CChecks::checkNonusedIP

Description: Checks if an IP is not in use and adds an error message to the message manager in case of an error.
Parameter:

- ip: IP to check.

Returns:

- true, if the IP is not used otherwise false.

5.10.29 CChecks::checkDNS1

Description: Checks if the 1st DNS server has an valid IP and adds an error message to the message manager in case of an error.
Parameter:

- dns1: DNS server IP to check.

Returns:

- true, if the IP is correct otherwise false.

5.10.30 CChecks::checkDNS2

Description: Checks if the 2nd DNS server has an valid IP and adds an error message to the message manager in case of an error.
Parameter:

- dns2: DNS server IP to check.

Returns:

- true, if the IP is correct otherwise false.

5.10.31 CChecks::checkGateway

Description: Checks if the gateway has an valid IP and adds an error message to the message manager in case of an error.
Parameter:

- gateway: Gateway IP to check.

Returns:

- true, if the IP is correct otherwise false.

5.10.32 CChecks::checkClientname

Description: Checks if a client name is valid (and optionally, if the client doesn't exist) and adds an error message to the message manager in case of an error.
Parameter:

- clientName: Clientname to check.
- checkNonused: Set to true, to check for non using.

Returns:

- true, if the clientname is correct otherwise false.

5.10.33 CChecks::checkNonusedClientname

Description: Checks if a client name is valid and if the client doesn't exist and adds an error message to the message manager in case of an error.
Parameter:

- clientName: Clientname to check.

Returns:

- true, if the MAC is correct and not in use otherwise false.

5.10.34 CChecks::checkNetmask

Description: Checks if the netmask is valid and adds an error message to the message manager in case of an error.
Parameter:

- netmask: Netmask to check.

Returns:

- true, if the netmask is correct otherwise false.

5.10.35 CChecks::checkMAC

Description: Checks if a MAC is valid (and optionally if it is not in use) and adds an error message to the message manager in case of an error.
Parameter:

- mac: MAC address to test.
- checkNonused: Set to true, to check for non using.

Returns:

- true, if the MAC is correct (and optionally not in use) otherwise false.

5.10.36 CChecks::checkNonusedMAC

Description: Checks if a MAC is valid and if it is not in use. Then adds an error message to the message manager in case of an error.
Parameter:

- mac: MAC address to test.

Returns:

- true, if the MAC is correct and not in use otherwise false.

5.10.37 CChecks::checkIPRange

Description: Checks if the input IPs are valid and if the first IP is "smaler" than the second. Then adds an error message to the message manager in case of an error.
Parameter:

- firstIp: The first IP marking the begin of the IP range.
- lastIp: The last IP marking the end of the IP range.

Returns:

- true, if the IPs are correct and the first IP is "smaler" otherwise false.

5.10.38 CChecks::checkStatus

Description: Checks if the client status number is valid. Then adds an error message to the message manager in case of an error.
Parameter:

- status: The status number to check.

Returns:

- true, if the status number is correct otherwise false.

5.10.39 CChecks::checkBootType

Description: Checks if if the client status number is valid. Then adds an error message to the message manager in case of an error.
Parameter:

- status: The status number to check.

Returns:

- true, if the status number is correct otherwise false.

5.10.40 CChecks::genericCHECK_FW

Description: Generic checking routine, that checks if an input value matches a rule.
Parameter:

- rule: CHECK_FW rule to check the input value with.
- val: Value to check.
- errorMsg: Error message to add, if the checking fails.

Returns:

- true, if the input value matches the rule otherwise false.

5.10.41 CChecks::checkPoolName

Description: Checks if the pool name is valid.
Parameter:

- poolName: The pool name to check.

Returns:

- true, if the pool name is valid otherwise false.

5.11 ./inc/CClientLister.php

Author: Hauke Goos-Habermann (HHabermann@pc-kiel.de)
Description: Class for generating lists of clients.

5.11.1 CClientLister::showClientTable

Description: Shows the table with matching clients and their given columns.

5.11.2 CClientLister::generateHTMLClientNameIdCheckbox

Description: Generates a checkbox for a client and its ID.
Parameter:

- clientInfo: Array containing information about the client (e.g. its name and ID).

Returns:

- HTML code for a checkbox containing client and ID information.

5.11.3 CClientLister::importCheckedClientNamesIds

Description: Imports the clientnames/IDs of the checked checkboxes.

5.11.4 CClientLister::getCheckedClientNames

Description: Returns name (as key) and ID (as value) of checked clients as array.
Returns:

- Array containing name (as key) and ID (as value) of checked clients.

5.11.5 CClientLister::isVisibleByPingableFilter

Description: Checks, if only clients that can (not) be pinged should be shown and if the given client matches the filter rule.
Parameter:

- CClientO: Object to the client to check.

Returns:

- true, if there is no pingable filter rule, or if the client matches the filter rule.

5.11.6 CClientLister::setVisibleByPingableFilter

Description: Sets the pingable filter rule (show only clients that can (not) be pinged or all clients).
Parameter:

- filterState: true (for showing only pingable clients), false (for not pingable clients) or null for all clients (filter deactivated).

5.11.7 CClientLister::getClientNames

Description: Gets an array with all clients matching the filter rules.
Returns:

- Array with all clients matching the filter rules.

5.11.8 CClientLister::showClientList

Description: Shows the table header of the client output list.

5.11.9 CClientLister::showClientListExtraLine

Description: Shows the extra line at the end of the client list.

5.11.10 CClientLister::setClientListExtraLine

Description: Sets the extra line that will be shown at the end of the client list.
Parameter:

- Arbitrary amount of cells to show at the end of the client list table.

5.11.11 CClientLister::getColumnNrToSort

Description: Figures out to column number which contains the values to sort.
Returns:

- Column number which contains the values to sort or false, if $this->orderBy is a SQL sorting statement.

5.11.12 CClientLister::getColumnNrWithContinousNumber

Description: Figures out to column number which contains the continous numbers.
Returns:

- Column number which contains the continous numbers or false, if no column with continous numbers is present.

5.11.13 CClientLister::cmpArrayElements

Description: Builds a sorting function that compares the colums of two arrays.
Parameter:

- columnNr: Column number to compare in the two arrays.

5.11.14 CClientLister::sortLines

Description: Does extra line sorting of special fields, that could not be sorted by SQL.
Parameter:

- lines: Array with the lines to sort.

5.11.15 CClientLister::getAscDescHeader

Description: Generates a sorting header with column title and ascending/descending button.
Parameter:

- orderBy: Ordering constant (ORDERBY_*).
- i18n: The name of the I18N variable.

5.11.16 CClientLister::getStatusHeader

Description: Generates a sorting header for the client's status.

5.11.17 CClientLister::getClientNameHeader

Description: Generates a sorting header for the client's name.

5.11.18 CClientLister::getInstallDateHeader

Description: Generates a sorting header for the client's installation date.

5.11.19 CClientLister::getLastModifyHeader

Description: Generates a sorting header for the client's last modifikation date.

5.11.20 CClientLister::getIPHeader

Description: Generates a sorting header for the client's IP address.

5.11.21 CClientLister::getMACHeader

Description: Generates a sorting header for the client's MAC address.

5.11.22 CClientLister::getJobsHeader

Description: Generates a sorting header for the client's waiting.

5.11.23 CClientLister::getPackagesHeader

Description: Generates a sorting header for the client's packages.

5.11.24 CClientLister::getGroupHeader

Description: Generates a sorting header for the client's groups.

5.11.25 CClientLister::showClientListHeader

Description: Shows the table header of the client output list.

5.11.26 CClientLister::setOutputColumns

Description: Sets the sequence of the columns to show in the output list.
Parameter:

- Arbitrary amount of CClientLister::COLUMN_* constants to mark the purpose of the columns.

5.11.27 CClientLister::setActionString

Description: Sets the action string, where CLIENT_NAME will be replaced by the actual name of the client and CLIENT_ID by its ID.
Parameter:

- actionString: Full action string (with placeholders).

5.11.28 CClientLister::getActionString

Description: Gets the action string.
Returns:

- The action string

5.11.29 CClientLister::setOrderBy

Description: Sets the ordering method for generating the output.
Parameter:

- order: ORDERBY_* mode to sort the output.

Returns:

- true on success.

5.11.30 CClientLister::addStatusFilter

Description: Adds an OR filter to get only clients that gave the given status (stati).
Parameter:

- operator: Operator (can be '=', '<', '>') selects if the client status should be equal, smaler or bigger that the given status.
- status: Status to compare with the state of the client

5.11.31 CClientLister::addKeyValueStoreFilter

Description: Adds a filter to get only clients that have a key value pair set the client's key value store.
Parameter:

- key: Name of the key.
- value: The value to store under the key.

5.11.32 CClientLister::addGroupFilter

Description: Adds the group filter to get only clients that are into the given group.
Parameter:

- groupName: Name of the group to filter.

5.11.33 CClientLister::addSearchFilter

Description: Adds a search filter to get only clients that match the search word in at least one table field.
Parameter:

- search: The search word.

5.11.34 CClientLister::setAscending

Description: Sets the output ordering ascending (default) or descending.
Parameter:

- ascending: Set to true, if the output should be generated in ascending order. For descending order, set to false.

5.11.35 CClientLister::isAscending

Description: Checks if the sorting of the column is ascending.
Returns:

- true, if sorting of the column is ascending otherwise false.

5.11.36 CClientLister::vmRunOnHostFilter

Description: Sets a filter to only give out virtual clients that are hosted on a given VM host.
Parameter:

- vmID: ID of the m23 client, that is VM host for other m23 clients.

5.11.37 CClientLister::resetGetting

Description: Sets back the MySQL connection to initalise a new search and getting of clients from the beginning.

5.11.38 CClientLister::getClient

Description: Gets a client matching all active filters. This can be called many times.
Returns:

- Client information in an associative array or false, if no (additional) clients clould be got.

5.11.39 CClientLister::getAllMatchingClients

Description: Gets all clients matching all active filters.
Returns:

- Array with information about all matching clients with client ID as key.

5.11.40 CClientLister::isMatchingClientPresent

Description: Checks if at least one clients with a given key-value-combination is found.
Parameter:

- key: The key to search for (e.g. office)
- val: The value to search for (e.g. home)

Returns:

- true, if at least one client is found.

5.11.41 CClientLister::getAllAutoUpdateClientNames

Description: Gets an array with the clients that currently run autoUpdate.
Returns:

- Associative array with the names of all client that currently run autoUpdate as key and value.

5.11.42 CClientLister::IPexists

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

- ip: IP to check

5.11.43 CClientLister::MACexists

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

- mac: MAC to check

5.11.44 CClientLister::ClientExists

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

- clientName: name of the client

5.12 ./inc/CClient.php

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

5.12.1 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.

5.12.2 CClient::__destruct

Description: Destructor for a CClient object. Before the object is removed from the RAM, all client settings are written to the DB.

5.12.3 CClient::setKeyValueStore

Description: Sets a value in the key value store of the client.
Parameter:

- key: Name of the key.
- value: The value to store under the key.
- check: An optional variable firewall check constant or rule or a function for check the validity of the value.
- errorMsg: An error message to give out, if the value fails the check.

5.12.4 CClient::getKeyValueStore

Description: Gets a value from the key value store of the client.
Returns:

- The value or NULL, if there is no value for the key.

5.12.5 CClient::copyImagingParameters

Description: Copies the imaging parameters (if present).
Parameter:

- options: Associative array with all options.

5.12.6 CClient::copyMassOptions

Description: Copies the mass installation options (if present).
Parameter:

- options: Associative array with all options.

5.12.7 CClient::setInstallationStatusBar

Description: Sets new percent value and/or new status text on the client's installation status bar.
Parameter:

- 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.

5.12.8 CClient::delSpecialJob

Description: Removes a special job from the joblist identified by package name and priority.
Parameter:

- package: Name of the package.
- priority: Priority of the job.

5.12.9 CClient::addJob

Description: Adds a job to the client's job table.
Parameter:

- packageName: name of the package
- priority: priority of the package
- params: parameter for installing the package

5.12.10 CClient::includeDistributionSpecificPackagesPHP

Description: Includes distribution specific packages.php.

5.12.11 CClient::addNormalJob

Description: Adds a normal package to the installation queue.
Parameter:

- packageName: name of the package
- priority: priority of the package

5.12.12 CClient::addSpecialJob

Description: Adds a special package to the installation queue.
Parameter:

- packageName: name of the package.
- params: Parameter for the special package.
- priority: priority of the package (if false, the priority from the special package will be used).

5.12.13 CClient::addUpdateSourcesListJob

Description: Adds a job to update the package source of the client to the installation queue.

5.12.14 CClient::addUpdatePackageInfosJob

Description: Adds a job to update the package information of the client to the installation queue.

5.12.15 CClient::addNormalUpdateJob

Description: Adds a job to perform a normal update of the client.
Parameter:

- type: normal or complete.

5.12.16 CClient::addCompleteUpdateJob

Description: Adds a job to perform a complete update of the client.
Parameter:

- type: normal or complete.

5.12.17 CClient::addUpdateJob

Description: Adds a job to update the client to the installation queue.
Parameter:

- type: normal or complete.

5.12.18 CClient::startAutoUpdate

Description: Starts auto update on a client.
Parameter:

- type: Normal (CAutoUpdate::UPDATE_NORMAL) or full (CAutoUpdate::UPDATE_FULL) update.

5.12.19 CClient::stopAutoUpdate

Description: Stops the auto update after successful or failed attempt.

5.12.20 CClient::getAutoUpdate_lastAttempt

Description: Gives back the timestamp of the last assignment of an auto update job.
Returns:

- Timestamp of the last assignment of an auto update job or false, in case of an error.

5.12.21 CClient::hasAutoUpdateJob

Description: Checks, if the client has an auto update job.
Returns:

- true, if the client has an auto update job, otherwise false.

5.12.22 CClient::checkAutoUpdate

Description: Checks, if an auto update is (physically) running or if it has failed or finished.

5.12.23 CClient::unsetInstPartDev

Description: Unsets the installation partition of the client (by removing the variable in the client info).

5.12.24 CClient::setInstPartDev

Description: Sets the installation partition of the client.
Parameter:

- instPart: Installation partition device name.

Returns:

- true, if the installation partition is valid otherwise false.

5.12.25 CClient::setUEFI

Description: Sets the UEFI usage of the client.
Parameter:

- enabled: true, if UEFI is used on the client, otherwise false.

5.12.26 CClient::isUEFIActive

Description: Returns, if the client uses UEFI.
Returns:

- true, if the client uses UEFI otherwise false.

5.12.27 CClient::isHalfSisterClient

Description: Returns, if the client uses a halfSister distribution.
Returns:

- true, if the client uses a halfSister distribution otherwise false.

5.12.28 CClient::getEFIBootPartDev

Description: Gets the EFI boot partition of the client.
Returns:

- The EFI partition of the client or false, if not set.

5.12.29 CClient::setEFIBootPartDev

Description: Sets the EFI partition of the client.
Parameter:

- EFIPart: EFI partition device name.

Returns:

- true, if the EFI boot partition is valid otherwise false.

5.12.30 CClient::unsetEFIBootPartDev

Description: Unsets the EFI boot partition of the client (by removing the variable in the client info).

5.12.31 CClient::getInstPartDev

Description: Gets the installation partition of the client.
Returns:

- The installation partition of the client or false, if not set.

5.12.32 CClient::unsetSwapPartDev

Description: Unsets the swap partition of the client (by removing the variable in the client info).

5.12.33 CClient::setSwapPartDev

Description: Sets the swap partition of the client.
Parameter:

- swapPart: Swap partition device name.

Returns:

- true, if the swap partition is valid otherwise false.

5.12.34 CClient::setMBRPartDev

Description: Sets the drive or partition where the bootloader should be installed.
Parameter:

- mbrPart: Drive or partition where the bootloader should be installed.

5.12.35 CClient::getSwapPartDev

Description: Gets the swap partition of the client.
Returns:

- The swap partition of the client or false, if not set.

5.12.36 CClient::isDerivedClient

Description: Checks, if the client is derived from a defined client.
Returns:

- true, if the client is derived, otherwise false.

5.12.37 CClient::setLanguage

Description: Sets the language of the client.
Parameter:

- language: Language of the client.

Returns:

- true, if the language is valid otherwise false.

5.12.38 CClient::getLanguage

Description: Gets the language of the client.
Returns:

- The language of the client.

5.12.39 CClient::setTimeZone

Description: Sets the timezone of the client.
Parameter:

- timeZone: Timezone of the client.

Returns:

- true, if the timezone is valid otherwise false.

5.12.40 CClient::getTimeZone

Description: Gets the timezone of the client.
Returns:

- The timezone of the client.

5.12.41 CClient::setBootloader

Description: Sets the bootloader of the client.
Parameter:

- bootloader: Bootloader of the client.

Returns:

- true, if the bootloader is valid otherwise dies.

5.12.42 CClient::getBootloader

Description: Gets the bootloader of the client.
Returns:

- The bootloader of the client.

5.12.43 CClient::getClientGroup

Description: Gets the m23 group of the client.
Returns:

- The m23 group of the client.

5.12.44 CClient::setNetRootPwd

Description: Generates and sets the netboot root password.

5.12.45 CClient::getNetRootPwd

Description: Gets the netboot root password.
Returns:

- Netboot root password.

5.12.46 CClient::setNfshomeserver

Description: Sets the NFS share of the client.
Parameter:

- nfshomeserver: NFS share with path.

Returns:

- true, if the NFS share is valid.

5.12.47 CClient::getNfshomeserver

Description: Gets the NFS share of the client.
Returns:

- NFS share of the client.

5.12.48 CClient::setBoolProperty

Description: Sets a boolean value in the client's settings.
Parameter:

- var: Name of the setting variable.
- setIt: Set to true, if the variable should be set, otherwise false.
- func: Name of the calling function (for error reporting)
- trueVal: String that should be set in the client's settings, if $setIt is true.
- falseVal: String that should be set in the client's settings, if $setIt is false.

Returns:

- true on sucessfully setting.

5.12.49 CClient::setInstallPrinter

Description: Sets, if the local printer should be detected/installed.
Parameter:

- setIt: Set to true, if the local printer should be detected/installed, otherwise false.

Returns:

- true on sucessfully setting.

5.12.50 CClient::getInstallPrinter

Description: Checks, if the local printer should be detected/installed.
Returns:

- true, if the local printer should be detected/installed, otherwise false.

5.12.51 CClient::setAddNewLocalLogin

Description: Sets, if the local login should be created.
Parameter:

- setIt: Set to true, if the local login should be created, otherwise false.

Returns:

- true on sucessfully setting.

5.12.52 CClient::getAddNewLocalLogin

Description: Checks, if the local login should be created.
Returns:

- true, if the local login should be created, otherwise false.

5.12.53 CClient::setGetSystemtimeByNTP

Description: Sets, if the system time should be set by NTP.
Parameter:

- setIt: Set to true, if the system time should be set by NTP otherwise false.

Returns:

- true on sucessfully setting.

5.12.54 CClient::getGetSystemtimeByNTP

Description: Checks, if the system time should be set by NTP.
Returns:

- true, if the system time should be set by NTP otherwise false.

5.12.55 CClient::setRootPassword

Description: Sets the root password for the client.
Parameter:

- rootPassword: The (encrypted) root password to set.
- cryptRootPw: set to true, if the password should be encrypted or false, if it's already encrypted.

Returns:

- true, if the root password is valid.

5.12.56 CClient::addToClientGroup

Description: Adds the client to an m23 client group.
Parameter:

- group: Name of the client group.

Returns:

- true, if the was added to the group.

5.12.57 CClient::setUserGroupIDs

Description: Sets the user ID and group ID (for LDAP).
Parameter:

- userID: The user ID.
- groupID: The group ID.

Returns:

- true, if user ID and group ID are valid otherwise false.

5.12.58 CClient::getGroupID

Description: Returns the (LDAP) group ID.
Returns:

- Group ID.

5.12.59 CClient::getUserID

Description: Returns the (LDAP) user ID.
Returns:

- User ID.

5.12.60 CClient::setUserDetails

Description: Generates HTML code for returning to the client control center page.
Parameter:

- forename: Forename of the user.
- familyname: Familyname of the user (or empty).
- eMail: eMail address of the user (or empty).
- office: Office of the user (or empty).
- login: The login name of the user.
- firstpw: The password of the user.

Returns:

- true, if all input parameters are valid.

5.12.61 CClient::getFirstpw

Description: Returns the user's first password.
Returns:

- The user's first password.

5.12.62 CClient::getFamilyname

Description: Returns the user's familyname.
Returns:

- The user's familyname.

5.12.63 CClient::getForename

Description: Returns the user's forename.
Returns:

- The user's forename.

5.12.64 CClient::getLogin

Description: Returns the user's login.
Returns:

- The user's login.

5.12.65 CClient::setKerberos

Description: Sets Kerberos information.
Parameter:

- KRB_server: IP or hostname of the Kerberos server.
- KRB_adminServer: IP or hostname of the Kerberos admin server.
- KRB_realm: Kerberos realm.

5.12.66 CClient::enableLDAPTLS

Description: Enables LDAP TLS.

5.12.67 CClient::setLDAPType

Description: Sets the LDAP type.
Parameter:

- type: LDAP usage type to set.

5.12.68 CClient::setLdapBasePrefix

Description: Sets the prefix, that should be put before the baseDN.
Parameter:

- prefix: The prefix, that should be put before the baseDN..

5.12.69 CClient::getLDAPType

Description: Returns the LDAP type of the client.
Returns:

- LDAP type of the client.

5.12.70 CClient::setLDAPServer

Description: Sets the LDAP server.
Parameter:

- LDAPServer: The name of the LDAP server.

Returns:

- true, if the LDAP server was set, otherwise false.

5.12.71 CClient::getLDAPServer

Description: Returns the LDAP server of the client.
Returns:

- LDAP server of the client.

5.12.72 CClient::addToCredentialsToLDAPServer

Description: Adds the credentials of the main (desktop) user to the given LDAP server.
Returns:

- If the credentials could be saved to the LDAP server, otherwise false.

5.12.73 CClient::getBackToDetailsLink

Description: Generates HTML code for returning to the client control center page.
Parameter:

- section: section to jump on the page

5.12.74 CClient::getClientWorkPHPURL

Description: Returns the URL to the work.php for this client.
Parameter:

- ip: If set, this IP is used instead of the server's IP.

Returns:

- URL to the work.php for this client.

5.12.75 CClient::getClientCurrentWorkPHP

Description: Returns the current contents of the work.php for this client.
Parameter:

- otherScript: If set, this job will be taken instead of the job with the lowest priority.

Returns:

- Current contents of the work.php for this client.

5.12.76 CClient::getClientPackages

Description: Returns an array or a space separated list of all packages installed on a client
Parameter:

- 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 (DEBPKGSTAT_installed, DEBPKGSTAT_removed, DEBPKGSTAT_purge).

5.12.77 CClient::save

Description: Saves the client parameters and options to the DB.

5.12.78 CClient::getProperty

Description: Gets a client property from $this->clientInfo and dies (or returns an error value), if this property is not set.
Parameter:

- key: Name of the property.
- dieMessage: Message to show before dying.
- return: This will returned in case of an error, if set to another value, than 'null'.

Returns:

- Value of the property (or error value).

5.12.79 CClient::getSourcesList

Description: Returns the sources list of the client.
Returns:

- Sources list of the client.

5.12.80 CClient::setArch

Description: Set the CPU architecture of the client.
Parameter:

- arch: The architecture to set.

Returns:

- true on successfully setting of the client's architecture, otherwise false.

5.12.81 CClient::getArch

Description: Returns the architecture of the client.
Returns:

- Architecture of the client.

5.12.82 CClient::setDistribution

Description: sets the distribution of the client.
Parameter:

- distr: Distribution of the client.

Returns:

- true on sucessfully setting, otherwise false.

5.12.83 CClient::getDistribution

Description: Returns the distribution of the client.
Returns:

- Distribution of the client.

5.12.84 CClient::setRelease

Description: Sets the distribution release of the client.
Parameter:

- release: Distribution release of the client.

Returns:

- true on sucessfully setting, otherwise false.

5.12.85 CClient::getRelease

Description: Returns the distribution release of the client.
Returns:

- Distribution release of the client.

5.12.86 CClient::updateModifyDate

Description: Updates the last modified date.

5.12.87 CClient::getModifyDate

Description: Returns the last modified date.
Returns:

- Last modified date of the client.

5.12.88 CClient::getModifyDateHumanReadable

Description: Returns the last modified date in human readable form.
Returns:

- Last modified date of the client in human readable form.

5.12.89 CClient::updateInstallDate

Description: Updates the installation date.

5.12.90 CClient::getInstallDate

Description: Returns the installation date.
Returns:

- Installation date of the client.

5.12.91 CClient::getInstallDateHumanReadable

Description: Returns the installation date in human readable form.
Returns:

- Last installation date of the client in human readable form.

5.12.92 CClient::export

Description: Exports all client settings as associativearray

5.12.93 CClient::setClientName

Description: Renames a client.
Parameter:

- clientName: New name of the client.

Returns:

- true on successfully setting of the client name, otherwise false.

5.12.94 CClient::getClientName

Description: Returns the name of the client.
Returns:

- Name of the client.

5.12.95 CClient::getID

Description: Returns the ID of the client.
Returns:

- ID of the client.

5.12.96 CClient::setDNS

Description: Sets the main and (optionally) the backup DNS server(s).
Parameter:

- dns1: The IP of the main DNS server.
- dns2: The IP of the backup DNS server.

Returns:

- true when the DNS(s) IP(s) is correct.

5.12.97 CClient::getDNS1

Description: Returns the IP of the main DNS server.
Returns:

- Main DNS server IP of the client.

5.12.98 CClient::getDNS2

Description: Returns the IP of the backup DNS server (if set).
Returns:

- Backup DNS server IP of the client.

5.12.99 CClient::setPackageProxy

Description: Sets the IP and port of the package proxy.
Parameter:

- proxyIP: The IP of the package proxy.
- proxyPort: The port of the package proxy.

Returns:

- true, if the package IP and port are valid and set otherwise false.

5.12.100 CClient::setIP

Description: Sets the IP of the client to an unsused IP.
Parameter:

- IP: unused IP for the client.

Returns:

- true on successfully setting the client's IP, otherwise false.

5.12.101 CClient::getIP

Description: Returns the client's IP.
Parameter:

- return: This will returned in case of an error, if set to another value, than 'null'.

Returns:

- IP of the client.

5.12.102 CClient::setNetmask

Description: Sets the netmask of the client.
Parameter:

- netmask: Netmask of the client.

Returns:

- true on successfully setting the client's netmask, otherwise false.

5.12.103 CClient::getNetmask

Description: Returns the client's netmask.
Returns:

- Netmask of the client.

5.12.104 CClient::setMAC

Description: Sets the MAC address of the client.
Parameter:

- mac: MAC of the client.

Returns:

- true on successfully setting the client's MAC, otherwise false.

5.12.105 CClient::getMAC

Description: Returns the client's MAC.
Parameter:

- return: This will returned in case of an error, if set to another value, than 'null'.

Returns:

- MAC of the client.

5.12.106 CClient::setSchool

Description: Sets the school the client belongs to.
Parameter:

- schoolName: Name of the school.

Returns:

- true on successfully setting the client's school, otherwise false.

5.12.107 CClient::getFreeIPABulkPassword

Description: Gets a the computer/host's FreeIPA bulk password (eventually after generating and storing).
Returns:

- Computer/host's bulk password.

5.12.108 CClient::getKerberosComputerPassword

Description: Gets a the Kerberos computer/host password (eventually after generating and storing).
Returns:

- Kerberos computer/host password.

5.12.109 CClient::getSchool

Description: Returns the school the client belongs to.
Parameter:

- return: This will returned in case of an error, if set to another value, than 'null'.

Returns:

- School the client belongs to.

5.12.110 CClient::setSerial

Description: Sets the serial (eg inventary number) of the client.
Parameter:

- serial: Serial/inventary number.

Returns:

- true on successfully setting the client's MAC, otherwise false.

5.12.111 CClient::getSerial

Description: Returns the client's serial number.
Parameter:

- return: This will returned in case of an error, if set to another value, than 'null'.

Returns:

- Serial number of the client.

5.12.112 CClient::setUCSNetwork

Description: Sets the UCSNetwork for the client.
Parameter:

- UCSNetwork: Name of the UCSNetwork.

Returns:

- true on successfully setting the client's UCSNetwork, otherwise false.

5.12.113 CClient::getUCSNetwork

Description: Returns the client's UCSNetwork.
Returns:

- UCSNetwork of the client.

5.12.114 CClient::setGateway

Description: Sets the gateway address for the client.
Parameter:

- gateway: IP address of the gateway.

Returns:

- true on successfully setting the client's gateway, otherwise false.

5.12.115 CClient::getGateway

Description: Returns the client's gateway.
Returns:

- Gateway of the client.

5.12.116 CClient::setGateway

Description: Sets the client's status.
Parameter:

- status: The status number to check.

Returns:

- true on successfully setting the client's status, otherwise false.

5.12.117 CClient::getStatus

Description: Returns the client's status.
Returns:

- Status of the client.

5.12.118 CClient::isDefinedClient

Description: Checks, if the client is defined client.
Returns:

- true, if the client is defined client, otherwise false.

5.12.119 CClient::generateHTMLStatusBar

Description: Generates HTML code containing the status of the client with links to the pages.

5.12.120 CClient::getBootType

Description: Returns the client's network boot type.
Returns:

- Network boot type of the client.

5.12.121 CClient::usesDynamicIP

Description: Checks if the client uses dynamic IPs.
Returns:

- true, if the client uses dynamic IPs otherwise false.

5.12.122 CClient::setBootType

Description: Sets the client's (network) boot type.
Parameter:

- bootType: CClient::BOOTTYPE_PXE, CClient::BOOTTYPE_NOBOOT, CClient::BOOTTYPE_ETHERBOOT, CClient::BOOTTYPE_GRUB2EFIX64

Returns:

- true on successfully setting the client's boot type, otherwise false.

5.12.123 CClient::getNetworkBootTypesArrayForSelection

Description: Generates an array with all avaialable network boot types for using it in a selection.
Returns:

- Array with all avaialable network boot types (depends on the usage of m23shared) for using it in a selection.

5.12.124 CClient::startInstall

Description: Starts the installation on the client

5.12.125 CClient::addShutdownPackage

Description: Adds a shutdown package, but only if the client is NOT running.
Returns:

- true, if a shutdown package is added.

5.12.126 CClient::activateNetboot

Description: Activates network booting for the client.
Returns:

- true on successfully restarting the DHCP server.

5.12.127 CClient::deactivateNetboot

Description: Deactivates network booting for the client.
Returns:

- true on successfully restarting the DHCP server.

5.12.128 CClient::wol

Description: Wakes a client over the network.

5.12.129 CClient::isNetbootActive

Description: Check, if network booting is active for the client.
Returns:

- true when network booting is active otherwise false.

5.12.130 CClient::isPingable

Description: Checks, if the client can be pinged over the network.

5.12.131 CClient::sshFetchJob

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

5.12.132 CClient::executeBySSH

Description: Runs a commands under a plain BASH with root rights on the client.
Parameter:

- cmds: the commands of the script

Returns:

- The output of the script.

5.12.133 CClient::generateHTMLClientNameBar

Description: Generates an URL with the client name linking to the client details page.
Returns:

- URL with the client name linking to the client details page.

5.12.134 CClient::generateHTMLPackagesBar

Description: Generates an URL with the amount of the client's packages linking to the client packages page.
Returns:

- URL with the amount of the client's packages linking to the client packages page.

5.12.135 CClient::generateHTMLWaitingAllJobsBar

Description: Generates an URL with the amount of the client's waiting jobs and all jobs linking to the change jobs page.
Returns:

- URL with the amount of the client's packages linking to the client packages page.

5.12.136 CClient::generateHTMLWaitingAllJobsBar

Description: Generates an URL with the amount of the client's waiting jobs and all jobs linking to the change jobs page.
Returns:

- URL with the amount of the client's packages linking to the client packages page.

5.12.137 CClient::getActionString

Description: Generates the action string (e.g. an URL for GET)
Parameter:

- actionString: Action string that may contain CLIENT_NAME (will be replaced by the name of the client) and CLIENT_ID (will be replaced by the ID of the client).

Returns:

- Changed (or unchanged) action string.

5.12.138 CClient::destroy

Description: Destroys a client finally.

5.12.139 CClient::getVMRole

Description: Returns the client's vmRole.
Returns:

- vmRole of the client.

5.12.140 CClient::getVmSoftware

Description: Returns the client's vmSoftware.
Returns:

- vmSoftware of the client.

5.13 ./inc/CFDiscAutomatic.php

Author: Hauke Goos-Habermann (HHabermann@pc-kiel.de)
Description: Class for visualisation of partitioning and formating.

5.13.1 CFDiscAutomatic::__construct

Description: Constructor for new CFDiscAutomatic objects. The object holds all information about the partitioning (of a client and loads the values from the DB).
Parameter:

- in: Name of an existing client (to load) or data of an empty disk.

5.13.2 CFDiscAutomatic::__destruct

Description: Destructor for a CFDiscAutomatic object. Before the object is removed from the RAM, all settings are written to the DB.

5.13.3 CFDiscAutomatic::getBiggestDrive

Description: Get the drive with the highest capacity.
Returns:

- Name of the drive with the highest capacity.

5.13.4 CFDiscAutomatic::createAutoPartitionFormatJob

Description: Generates a automatic partition and formating job for the drive with the highest capacity.

5.14 ./inc/CFDiskBasic.php

Author: Hauke Goos-Habermann (HHabermann@pc-kiel.de)
Description: Class for basic partitioning and formating functions.

5.14.1 CFDiskBasic::__construct

Description: Constructor for new CFDiskBasic objects. The object holds all information about the partitioning (of a client and loads the values from the DB).
Parameter:

- in: Name of an existing client (to load) or data of an empty disk.

5.14.2 CFDiskBasic::__destruct

Description: Destructor for a CFDiskBasic object. Before the object is removed from the RAM, all settings are written to the DB.

5.14.3 CFDiskBasic::deletePartitionJob

Description: Generates a partition removal job and adds it to the list of partition steps.
Parameter:

- dev: selected device (e.g. hda)
- pPart: Physical partition number, minor number in parted
- addAtTheBeginning: Adds the job to the beginning of the partitionSteps and partitionStepsForShift arrays.

5.14.4 CFDiskBasic::createPartitionJob

Description: Generates a partition add job and adds it to the list of partition steps.
Parameter:

- dev: selected device (e.g. /dev/hda)
- start: start point of the partition (in MB)
- end: end point of the partition (in MB)
- type: type of the partition (primary, logical)
- pPart: number of the device (e.g. 1 with /dev/hda1)

5.14.5 CFDiskBasic::bootflagJob

Description: Enables the booting flag on a partition and adds it to the list of partition steps.
Parameter:

- dev: Disk to activate booting on (e.g. /dev/hda)
- pPart: number of partition, minor number in parted

5.14.6 CFDiskBasic::EFItypeAndGUIDJob

Description: Sets EFI boot partition type and GUID for the EFI boot partition and adds it to the list of partition steps.
Parameter:

- dev: Disk with the EFI partition (e.g. /dev/hda)
- pPart: Number of EFI partition, minor number in parted

5.14.7 CFDiskBasic::formatJob

Description: Generates a partition format job and adds it to the list of partition steps.
Parameter:

- dev: device to format (e.g. /dev/hda1)
- fileSys: file system of the partition: ext3, ext2, linux-swap
- partJobs: associative array with partition jobs

5.14.8 CFDiskBasic::createRaidJob

Description: Generates a RAID creation job and adds it to the list of partition steps.
Parameter:

- dev: RAID device to create (e.g. /dev/md0)
- devsBuildingDiskArray: Array with the devices that build the RAID.
- level: RAID level.

5.14.9 CFDiskBasic::deleteRaidJob

Description: Generates a RAID deletion job and adds it to the list of partition steps.
Parameter:

- dev: RAID device to delete (e.g. /dev/md0)

5.14.10 CFDiskBasic::makeInstOrEFIPartBootable

Description: Enables the booting flag on the installation/EFI partition by adding a job.

5.14.11 CFDiskBasic::rereadPartTable

Description: Let the OS re-read the partition table.
Parameter:

- dev: The device that was changed/created (e.g. /dev/sda5).

5.14.12 CFDiskBasic::getMknodCommandsForDeviceArray

Description: Generates the mknod commands for given /dev/sdX(Y) devices (disks or partitions).
Parameter:

- devs: Array with the devices (e.g. /dev/sda5) to created the mknod commands for.

Returns:

- mknod commands with the parameter matching the given /dev/sdX(Y).

5.14.13 CFDiskBasic::efiCheckOnClient

Description: Checks the validity of an EFI partition diretly on the m23 client, if EFI is used and writes the result to /tmp/efiCheck.result.

5.14.14 CFDiskBasic::runIfEFIPartInvalid

Description: Takes BASH commands (for partitioning and formating) and executes them only, if EFI is not used, the commands are NOT ment for the drive the EFI partition is on or the EFI partition itself and if the EFI partition is invalid.
Parameter:

- cmd: BASH commands (for partitioning and formating)
- drive: The drive (eg. /dev/sda) the commands are ment for
- pPart: The physical partition number (eg. 1) the commands are ment for or false, if the whole drive is the parameter for the BASH code (eg. when creating a new partition)

Returns:

- Commands for creating and deletion of partitions, formating or building RAIDs.

5.14.15 CFDiskBasic::genPartedCommands

Description: Generates commands for creating and deletion of partitions, formating or building RAIDs.
Parameter:

- mkfsextOptions: Extra parameter for mkfs.extX .
- sourceslist: Name of the package sources list of the client (needed for finding a supported file system).
- addLogStatusCommands: Set to true, if after each partition / format command a check and reporting to the m23 server should be added.

Returns:

- Commands for creating and deletion of partitions, formating or building RAIDs.

5.14.16 CFDiskBasic::createPartition

Description: Creates a new partition on a disk (if possible).
Parameter:

- dev: The device (e.g. /dev/sda).
- start: Start position (in MB) of the partition to create.
- end: End position (in MB) of the partition to create.
- type: type of the partition (CFDiskIO::PT_PRIMARY, CFDiskIO::PT_EXTENDED, CFDiskIO::PT_LOGICAL)
- bootable: Set to true, if the partition should be made bootable.

Returns:

- The physical partition number of the newly created partition.

5.14.17 CFDiskBasic::createUEFIPartition

Description: Creates a new UEFI partition with a size of 512 MB at the start of the disk (if possible).
Parameter:

- dev: The device (e.g. /dev/sda).

Returns:

- The physical partition number of the newly created partition.

5.14.18 CFDiskBasic::deletePartition

Description: Deletes a partition from a disk.
Parameter:

- dev: The device (e.g. /dev/sda).
- deleteBelongingRaid: If set to true, the RAID, the partition belongs to will be destroyed.
- massInstallMode: Adds the job to the beginning of the partitionSteps and partitionStepsForShift arrays and doesn't unset installation and swap partition.

5.14.19 CFDiskBasic::formatPartition

Description: Formats a partition.
Parameter:

- dev: partition to format (e.g. /dev/hda1)
- fs: type of filesystem

5.14.20 CFDiskBasic::deleteAllPartitions

Description: Deletes all partitions on a disk.
Parameter:

- diskDev: The device (e.g. /dev/sda).

5.14.21 CFDiskBasic::createInstallPartition

Description: Creates and formats an installation partition and makes it bootable. The created partition is stored in the client parameters as installation partition.
Parameter:

- diskDev: The disk device (e.g. /dev/hda) where the installation partition should be created on.
- instStart: Start position of the installation partition (in MB).
- instEnd: End position of the installation partition (in MB).

5.14.22 CFDiskBasic::createSwapPartition

Description: Creates and formats a swap partition. The created partition is stored in the client parameters as swap partition.
Parameter:

- diskDev: The disk device (e.g. /dev/hda) where the swap partition should be created on.
- instStart: Start position of the swap partition (in MB).
- instEnd: End position of the swap partition (in MB).

5.14.23 CFDiskBasic::createFormatPartition

Description: Creates and formats a partition.
Parameter:

- diskDev: The disk device (e.g. /dev/hda) where the installation partition should be created on.
- instStart: Start position of the installation partition (in MB).
- instEnd: End position of the installation partition (in MB).
- fileSystem: Type of the file system (eg. ext4)
- partType: Type of the partition (eg. CFDiskIO::PT_PRIMARY)

5.14.24 CFDiskBasic::autoPartitionDisk

Description: Automatically partitions and formats a disk.
Parameter:

- diskDev: The disk device (e.g. /dev/hda) that should be partitionated and formated automatically.
- minSwap: Minimal size of the swap partition in MB.
- maxSwap: Maximal size of the swap partition in MB.

5.14.25 CFDiskBasic::PM_auto2Disk1SysSwap2Home

Description: Automatically partitions two disks (1st for system + swap, 2nd for /home).

5.14.26 CFDiskBasic::PM_auto500GBsysSwapData

Description: Automatically partitions and formats the first 500GB a disk. It creates a system, swap and data partition.
Parameter:

- diskDev: The disk device (e.g. /dev/hda) that should be partitionated and formated automatically.

5.14.27 CFDiskBasic::createAllRaidJobs

Description: Generates the jobs to create all RAIDs.
Returns:

- true, if all RAIDs are complete, otherwise false.

5.14.28 CFDiskBasic::createRaidJobForRaid

Description: Generates the jobs to create a given RAID.
Parameter:

- vrDisk: Virtual (internally used) disk number for the RAID.

Returns:

- true, if $vrDisk is a RAID, otherwise false.

5.14.29 CFDiskBasic::createRaid

Description: Creates a new RAID disk.
Parameter:

- dev: Device name for the RAID (eg. /dev/md0)
- level: RAID level.

Returns:

- Virtual RAID disk number, if the device string is suitable for a RAID, otherwise false.

5.14.30 CFDiskBasic::deleteRaid

Description: Deletes a RAID disk.
Parameter:

- dev: Device name for the RAID (eg. /dev/md0)

Returns:

- true, if deleting of the RAID works, otherwise false.

5.14.31 CFDiskBasic::assignDeviceToRaid

Description: Adds a partition or disk to a RAID disk.
Parameter:

- raidDev: Device name of the RAID (eg. /dev/md0)
- devToAdd: partition or disk device name to add.

Returns:

- true, if the device could be added, otherwise false.

5.14.32 CFDiskBasic::deleteDeviceFromRaid

Description: Deletes a partition or disk from a RAID disk.
Parameter:

- raidDev: Device name of the RAID (eg. /dev/md0)
- devToRemove: Partition or disk device name to remove.

Returns:

- true, if the device could be removed, otherwise false.

5.14.33 CFDiskBasic::deleteAllPartitionsOnDisk

Description: Deletes all partitions on a disk.
Parameter:

- dev: Device name of the disk.

5.14.34 CFDiskBasic::fdiskAdjustPartitioningLinearScale

Description: Scales all partitions sizes to match the full disk size.
Parameter:

- diskDevToAdjust: Device name of the disk to ajust.

5.14.35 CFDiskBasic::fdiskAdjustPartitioning

Description: Adjusts the disk for a derived client, based on the defined client's settings

5.15 ./inc/CFDiskGUI.php

Author: Hauke Goos-Habermann (HHabermann@pc-kiel.de)
Description: Class for visualisation of partitioning and formating.

5.15.1 CFDiskBasic::__construct

Description: Constructor for new CFDiskGUI objects. The object holds all information about the partitioning (of a client and loads the values from the DB).
Parameter:

- in: Name of an existing client (to load) or data of an empty disk.

5.15.2 CFDiskGUI::__destruct

Description: Destructor for a CFDiskGUI object. Before the object is removed from the RAM, all settings are written to the DB.

5.15.3 CFDiskGUI::getFSHint

Description: Extends selected file system names by hints.
Parameter:

- fs: Name of the file system.

Returns:

- Name of the file system potentially extended by a hint.

5.15.4 CFDiskGUI::fdiskSessionSetter

Description: Generic function to store values in the client partition and format session or loads them.
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.

5.15.5 CFDiskGUI::fdiskSessionPage

Description: Stores the page in the session or loads it.
Parameter:

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

Returns:

- The current page.

5.15.6 CFDiskGUI::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).

5.15.7 CFDiskGUI::fdiskSessionHelpPage

Description: Stores the help page in the session or loads it.
Parameter:

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

Returns:

- The current help page.

5.15.8 CFDiskGUI::getDiskLayoutEntryStart

Description: Gets the start position of a disk layout entry.
Parameter:

- vDisk: Virtual (internally used) device number.
- layoutNr: Number of the disk layout entry.

Returns:

- Start position of a disk layout entry or dies, if not set.

5.15.9 CFDiskGUI::getDiskLayoutEntryEnd

Description: Gets the end position of a disk layout entry.
Parameter:

- vDisk: Virtual (internally used) device number.
- layoutNr: Number of the disk layout entry.

Returns:

- End position of a disk layout entry or dies, if not set.

5.15.10 CFDiskGUI::getDiskLayoutEntrySize

Description: Gets the size of a disk layout entry.
Parameter:

- vDisk: Virtual (internally used) device number.
- layoutNr: Number of the disk layout entry.

Returns:

- Size of a disk layout entry or dies, if not set.

5.15.11 CFDiskGUI::getDiskLayoutEntryType

Description: Gets the type of a disk layout entry.
Parameter:

- vDisk: Virtual (internally used) device number.
- layoutNr: Number of the disk layout entry.

Returns:

- Type of a disk layout entry or dies, if not set.

5.15.12 CFDiskGUI::getDiskLayoutEntryFileSystem

Description: Gets the filesystem of a disk layout entry.
Parameter:

- vDisk: Virtual (internally used) device number.
- layoutNr: Number of the disk layout entry.

Returns:

- Filesystem of a disk layout entry or dies, if not set.

5.15.13 CFDiskGUI::getFileSystemTranslator

Description: Translates the filesystem names.
Parameter:

- fs: Filesystem to translate.

Returns:

- Translated filesystem.

5.15.14 CFDiskGUI::getPartitionTypeTranslator

Description: Translates the partition types.
Parameter:

- type: Partition type to translate.

Returns:

- Translated partition type.

5.15.15 CFDiskGUI::getDiskLayoutEntryDev

Description: Gets the device name of a disk layout entry.
Parameter:

- vDisk: Virtual (internally used) device number.
- layoutNr: Number of the disk layout entry.

Returns:

- Device name of a disk layout entry or dies, if not set.

5.15.16 CFDiskGUI::getDiskLayoutEntriesAmount

Description: Gets the amount of disk layout entries.
Parameter:

- vDisk: Virtual (internally used) device number.

Returns:

- The amount of disk layout entries.

5.15.17 CFDiskGUI::getFreeSpaceHTMLColor

Description: Gets the HTML color for marking free space.
Returns:

- HTML color for marking free space.

5.15.18 CFDiskGUI::showFreeSpaceBarBlock

Description: Shows a block for with free space in the bar visualising the partitioning of a disk.
Parameter:

- pDisk: The physical device name of the disk (eg. /dev/sda)
- start: Start point of the partition (in MB)
- end: End point of the partition (in MB)
- width: The width of the block in percent (without
- type: Type of the partition.
- 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.

5.15.19 CFDiskGUI::showEmptyDiskTable

Description: Shows a table with block for an empty disk (if the disk has no partitions)
Parameter:

- pDisk: The physical device name of the disk (eg. /dev/sda)
- 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.

Returns:

- true, if the disk has no partitions, otherwise false.

5.15.20 CFDiskGUI::getPartitionExtraIconHTML

Description: Returns HTML code for showing an icon for installation or swap partition.
Parameter:

- dev: the Device of the partition.

Returns:

- HTML code for showing an icon for installation or swap partition, if the given device is installation or swap partition. Otherwise empty string.

5.15.21 CFDiskGUI::getExtraCSSForExtended

Description: Returns CSS code for marking partitions or free space on the extended partition.
Parameter:

- type: Type of the partition.

Returns:

- CSS code for marking partitions or free space on the extended partition.

5.15.22 CFDiskGUI::showPartitionBarBlock

Description: Shows a block for with free space in the bar visualising the partitioning of a disk.
Parameter:

- pDisk: The physical device name of the disk (eg. /dev/sda)
- start: Start point of the partition (in MB)
- end: End point of the partition (in MB)
- width: The width of the block in percent (without
- type: Type of the partition.
- fileSystem: Filesystem name.
- 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.

5.15.23 CFDiskGUI::printBars

Description: prints the partitions as colored table
Parameter:

- pDisk: 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.

5.15.24 CFDiskGUI::showPartTable2

Description: Shows the partition information for a disk as table.
Parameter:

- vDisk: Virtual (internally used) device number.

5.15.25 CFDiskGUI::addDiskLayoutEntry

Description: Adds a new entry (partition or free space on a disk) to the disk layout (for rendering only).
Parameter:

- vDisk: Internal disk number (for accessing the disk information in the array).
- start: start point of the partition (in MB)
- end: end point of the partition (in MB)
- type: type of the partition (primary, logical)
- fs: File system.
- dev: Device name (optional)

5.15.26 CFDiskGUI::newDiskLayout

Description: Creates a new disk layout for a given disk.
Parameter:

- vDisk: Internal disk number (for accessing the disk information in the array).

Returns:

- true, if the disk layout will be calculated, false, if it is in the cache

5.15.27 CFDiskGUI::updateDiskLayout

Description: Updates the disk layout for a given disk, if it is not in the cache.
Parameter:

- vDisk: Internal disk number (for accessing the disk information in the array).

Returns:

- true, if the disk layout was calculated, false, if it is in the cache

5.15.28 CFDiskGUI::showCombinedFdiskGUIDialog

Description: Shows the new partition and formating screen.

5.15.29 CFDiskGUI::getCurrentDiskDev

Description: Returns the currently choosen (in the GUI) disk device name.
Returns:

- Currently choosen (in the GUI) disk device name.

5.15.30 CFDiskGUI::getCurrentDiskvDev

Description: Returns the currently choosen (in the GUI) virtual disk number.
Returns:

- Currently choosen (in the GUI) virtual disk number.

5.15.31 CFDiskGUI::getCurrentPartDev

Description: Returns the currently choosen (in the GUI) partition device name.
Returns:

- Currently choosen (in the GUI) partition device name.

5.15.32 CFDiskGUI::finalChecksAndRealPartitionAndFormatStart

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

5.15.33 CFDiskGUI::showFdiskCombinedGUIFunctions

Description: Shows the menu bar with integrated logic for CFDiskGUI::showCombinedFdiskGUIDialog.

5.15.34 CFDiskGUI::getPartitionPercent

Description: Calculates the percentual size of a selected partition in comparison to the disk size.
Parameter:

- vDisk: 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)

5.15.35 CFDiskGUI::getHTMLColorForFilesystemOrType

Description: Get HTML color code for a given filesystem or partition type.
Parameter:

- fsOrType: Name of the file system (ext3, ext2, linux-swap,...) or type of the partition (eg. CFDiskIO::PT_FREE)

Returns:

- HTML color code for the file system or partition type.

5.15.36 CFDiskGUI::getPartTableEntryNr

Description: Generates a unique number for the partitions and free spaces on a disk bar and for the partition table (used in JavaScript).
Parameter:

- vDisk: Internal disk number (for accessing the disk information in the array).
- start: Start position of the partition / free space.
- end: End position of the partition / free space.

Returns:

- Unique number

5.15.37 CFDiskGUI::getJSSelectPartTypeForNewPartitions

Description: Get the first entry of the list of partition types that can be created in a range on a disk as selectPartTypeForNewPartitions JavaScript function.
Parameter:

- vDisk: Internal disk number (for accessing the disk information in the array).
- start: Start position (in MB) of the partition to create.
- end: End position (in MB) of the partition to create.

Returns:

- Call for selectPartTypeForNewPartitions with partition type that can be created or empty, if there no partion can be created.

5.15.38 CFDiskGUI::printAllBars

Description: Shows the partition bars of all disks specified for the current client.

5.15.39 CFDiskGUI::getPartInfoIcon

Description: Generates HTML code for showing an icon with status information about a drive or partition.
Parameter:

- vDisk: 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.

Returns:

- HTML code for showing an icon with status information about the drive or partition.

5.15.40 CFDiskGUI::getPartInfoString

Description: Generates an info string, that shows information about the device name of the partition, its filesystem and bolonging to a RAID.
Parameter:

- vDisk: 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)

Returns:

- Info string.

5.15.41 CFDiskGUI::getDiskInfoString

Description: Generates an info string, that shows information about the device name of the drive and bolonging to a RAID.
Parameter:

- vDisk: Virtual (internally used) device number.

Returns:

- Info string.

5.15.42 CFDiskGUI::getDriveInfoIcon

Description: Generates HTML code for showing an icon with status information about a drive.
Parameter:

- vDisk: Virtual (internally used) device number.

Returns:

- HTML code for showing an icon with status information about the drive.

5.15.43 CFDiskGUI::listPartJobs

Description: Show all part jobs in the table.

5.15.44 CFDiskGUI::showAllPartTables

Description: Shows the partition tables of all disks for the current client.

5.15.45 CFDiskGUI::getRaidTable

Description: Get informations about the assigned real disks/partitions of a RAID.
Parameter:

- raidDev: Device name of the new drive (e.g. /dev/md0)
- withDeleteButtons: If set to true, each line with a real disk/partition gets an extra button for deleting it from the RAID.

Returns:

- HTML table with informations about the assigned real drives/partitions.

5.15.46 CFDiskGUI::showColorDefinitions

Description: Shows the color definitions for the supported filesystems.

5.15.47 CFDiskGUI::fstabAddDialog

Description: Shows a dialog (with logic) for adding fstab entries.

5.15.48 CFDiskGUI::getFstabTable

Description: Generates a HTML table with all fstab entries and logic for deleting entries.
Returns:

- HTML table with the fstab.

5.15.49 CFDiskGUI::getHDSizes

Description: Returnes the sizes of all harddisks in a string, sperated by given line separator.
Parameter:

- lineSeparator: String to separate the output entries in the output string.

Returns:

- Sizes of all harddisks in a string, sperated by given line separator.

5.16 ./inc/CFDiskIO.php

Author: Hauke Goos-Habermann (HHabermann@pc-kiel.de)
Description: Class for lowlevel organisation of partitioning and formating.

5.16.1 CFDiskIO::__construct

Description: Constructor for new CFDiskIO objects. The object holds all information about the partitioning (of a client and loads the values from the DB).
Parameter:

- in: Name or object of an existing client (to load) or data of an empty disk.
- checkMode: Check for the input variable.

5.16.2 CFDiskIO::__destruct

Description: Destructor for a CFDiskIO object. Before the object is removed from the RAM, all settings are written to the DB.

5.16.3 CFDiskIO::isTypeFreeSpace

Description: Checks, if a given partition type means "free space".
Parameter:

- type: Partition type to check.

Returns:

- true, if the given partition type means "free space", otherwise false.

5.16.4 CFDiskIO::fdiskGetProperty

Description: Returns the given variable, if it is set or dies (or returns an error value) with an error message.
Parameter:

- var: Variable to check.
- return: This will returned in case of an error, if set to another value, than 'null'.

Returns:

- Input variable (or error value).

5.16.5 CFDiskIO::getCurrentPartitioning

Description: Returns the current (physical) partitioning (of a client).
Returns:

- The current (physical) partitioning (of a client).

5.16.6 CFDiskIO::setCurrentPartitioning

Description: Sets the current (physical) partitioning (of a client).

5.16.7 CFDiskIO::fdiskLoadFromDB

Description: Loads the current (physical) partitioning of a client and CFDiskTemp values.

5.16.8 CFDiskIO::setCFDiskTemp

Description: Sets CFDiskTemp of the client.
Parameter:

- CFDiskTemp: Array with the partition and format information.

5.16.9 CFDiskIO::getDiskDevs

Description: Returns an array with all disk devices (/dev/sdX) as key and value.
Returns:

- Array with all disk devices (/dev/sdX) as key and value.

5.16.10 CFDiskIO::getDiskAmount

Description: Returns amount of disks.
Returns:

- Amount of disks.

5.16.11 CFDiskIO::getPartAmount

Description: Returns the of partitions of a given disk.
Parameter:

- vDisk: Internal disk number (for accessing the disk information in the array).

Returns:

- Amount of partitions of a given disk.

5.16.12 CFDiskIO::getpDiskAndpPartFromDev

Description: Splits a device (e.g. /dev/hda1) in the physical disk (/dev/hda) and the partition number (1).
Parameter:

- dev: The device to partition (e.g. /dev/hde1)
- pDisk: The parameter, the physical disk is written to (e.g. /dev/hda).
- pPart: The parameter, the physical partition number (e.g. 1) or false (if there is no number in the dev) is written to.
- ignoreMD: Set to true, if a "partition number" from a MD should be received.

5.16.13 CFDiskIO::isDevValidDiskPartitionOrRaid

Description: Checks, if a given device name is a valid disk, partition or RAID device.
Parameter:

- dev: The device to check.

Returns:

- true, if a given device name is a valid disk, partition or RAID device, otherwise false.

5.16.14 CFDiskIO::dev2VDiskVPart

Description: Searches for a device (e.g. /dev/sda2) and writes the virtual disk and partition numbers to the variables. These values can be used to access the array $this->wantedPartitioning.
Parameter:

- dev: The device (e.g. /dev/sda2).
- vDisk: Internal disk number (for accessing the disk information in the array) or false, if no matching (physical) disk number was found.
- vPart: Internal partition number in a disk (for accessing the disk information in the array) or false, if no matching (physical) partition number was found.

Returns:

- true if the search worked otherwise false.

5.16.15 CFDiskIO::getPreviousPartitionStep

Description: Get the previous partition steps and wanted partitioning from the last action.
Returns:

- Associative array with wanted partitioning as key 'wp' and the partition steps under key 'ps' from the last action or NULL, if there are no undo steps.

5.16.16 CFDiskIO::backToPreviousPartitionStep

Description: Jumps back in the partition history by one step.

5.16.17 CFDiskIO::findAndSetEFIBootPartDev

Description: Searches for the first vfat partition and if one is found, sets it as EFI boot partition.

5.16.18 CFDiskIO::discardUndo

Description: Discards all undo steps.

5.16.19 CFDiskIO::updateUndoMd5

Description: Updates the md5 sum of the wantedPartitioning and partitionSteps arrays.

5.16.20 CFDiskIO::getUndoMd5

Description: Gets the md5 sum of the partitionSteps arrays.

5.16.21 CFDiskIO::addUndo

Description: Adds an undo step to the undoArray.

5.16.22 CFDiskIO::addPartitionStepAtTheBeginning

Description: Adds a step at the beginning of the partitionSteps and partitionStepsForShift arrays.
Parameter:

- partJob: The job to add.

5.16.23 CFDiskIO::addPartitionStepBeforeFormat

Description: Adds a step to the partitionSteps and partitionStepsForShift arrays before the formating of the RAID device (given in the $partJob).
Parameter:

- partJob: The RAID creation commands to add.

5.16.24 CFDiskIO::addPartitionStepBeforeFormatArray

Description: Adds a step to the partitionSteps and partitionStepsForShift arrays before the formating of the RAID device (given in the $partJob).
Parameter:

- newJob: The RAID creation commands to add.
- partJobs: Pointer to the array with the partition steps (partitionSteps or partitionStepsForShift).

Returns:

- true, if there was a formating job for the RAID device. false, if the new job was placed at the end.

5.16.25 CFDiskIO::addPartitionStep

Description: Adds a step to the partitionSteps and partitionStepsForShift arrays.
Parameter:

- cmd: The commands to add.

5.16.26 CFDiskIO::shiftPartitionStep

Description: Gets the first element of the partitionSteps array and deletes it.

5.16.27 CFDiskIO::resetWantedPartitioningAndSteps

Description: Resets the wantedPartitioning array by replacing it with the contents of the client's current partitioning and resets the partition steps.

5.16.28 CFDiskIO::resetWantedPartitioning

Description: Resets the wantedPartitioning array by replacing it with the contents of the client's current partitioning.

5.16.29 CFDiskIO::fdiskSaveToDB

Description: Saves CFDiskTemp values to the DB.

5.16.30 CFDiskIO::getCurrentDiskSizesForDefinedDiskSizes

Description: Gets the disk devices and their sizes, if the client is a defined client.
Parameter:

- overwrite: Set to true, if the disk devices and their sizes should be saved even if the client is NOT a defined client.

Returns:

- Array with disk devices and their sizes, if the client is a defined client, otherwise false.

5.16.31 CFDiskIO::saveDefinedDiskSizesToDB

Description: Saves the disk devices and their sizes to the DB.

5.16.32 CFDiskIO::getDefinedDiskSizes

Description: Gets the defined disk devices and their sizes.
Returns:

- Array with disk devices and their sizes, if the set, otherwise false.

5.16.33 CFDiskIO::convertPartitioning2Array

Description: Converts old partitioning data to array.

5.16.34 CFDiskIO::isDiskRaid

Description: Checks, if a disk is a RAID.
Parameter:

- vDisk: Internal disk number (for accessing the disk information in the array).

Returns:

- true, if the disk is a RAID, otherwise false.

5.16.35 CFDiskIO::isDevRaid

Description: Checks, if a device string is a RAID.
Parameter:

- dev: The device (e.g. /dev/sda2 or /dev/md0).

Returns:

- true, if the device string is a RAID, otherwise false.

5.16.36 CFDiskIO::isNVMe

Description: Checks, if a device string is an NVMe.
Parameter:

- dev: The device (e.g. /dev/sda2 or /dev/md0).

Returns:

- true, if the device string is an NVMe, otherwise false.

5.16.37 CFDiskIO::setDiskPartLockedByRaid

Description: Sets or unsets the RAID lock of a partition or disk.
Parameter:

- vDisk: Internal disk number (for accessing the disk information in the array).
- vPart: Internal partition number in a disk (for accessing the disk information in the array).
- lock: Set to true, to set tha RAID lock, false for removing the lock.

5.16.38 CFDiskIO::isDiskOrPartLockedByRaid

Description: Checks, if a partition or disk is used as RAID.
Parameter:

- vDisk: Internal disk number (for accessing the disk information in the array).
- vPart: Internal partition number in a disk (for accessing the disk information in the array).

Returns:

- true, if the disk or partition is locked by a RAID.

5.16.39 CFDiskIO::isPartitionLockedByRaid

Description: Checks, if a partition is locked, because it is part of a RAID.
Parameter:

- vDisk: Internal disk number (for accessing the disk information in the array).
- vPart: the virtual partition number, that is used to build the variable name to access the associative array.

Returns:

- true, if the partition is locked, otherwise false.

5.16.40 CFDiskIO::isDiskLockedByRaid

Description: Checks, if a disk is locked, because it is part of a RAID.
Parameter:

- vDisk: Internal disk number (for accessing the disk information in the array).

Returns:

- true, if the disk is locked, otherwise false.

5.16.41 CFDiskIO::isDiskNotLockedByRaidAgainstCreationOfNewPartition

Description: Checks if a new partition can be created or if the complete disk is used for RAID.
Parameter:

- vDisk: Internal disk number (for accessing the disk information in the array).

Returns:

- true, if a new partition can be created, otherwise false.

5.16.42 CFDiskIO::getPartitionAmountOfType

Description: Count all partitions of a selected type on a disk.
Parameter:

- vDisk: Internal disk number (for accessing the disk information in the array).
- type: type of the partition (CFDiskIO::PT_PRIMARY, CFDiskIO::PT_EXTENDED, CFDiskIO::PT_LOGICAL, CFDiskIO::PT_EFI)

5.16.43 CFDiskIO::getExtendedVPart

Description: Searches the vPart that contains the extended partition.
Parameter:

- vDisk: Internal disk number (for accessing the disk information in the array).

Returns:

- vPart number of the extended partition, if there is an one, otherwise false.

5.16.44 CFDiskIO::getDiskSize

Description: Returns the size of the disk.
Parameter:

- vDisk: Internal disk number (for accessing the disk information in the array).

Returns:

- Size of the disk.

5.16.45 CFDiskIO::getDiskDev

Description: Returns the device name (e.g. /dev/sda) for a virtual disk.
Parameter:

- vDisk: Internal disk number (for accessing the disk information in the array).

Returns:

- Device name (e.g. /dev/sda) for a virtual disk.

5.16.46 CFDiskIO::getPartitionStart

Description: Returns the start position (in MB) of a partition.
Parameter:

- vDisk: Internal disk number (for accessing the disk information in the array).
- vPart: the virtual partition number, that is used to build the variable name to access the associative array.
- return: This will returned in case of an error, if set to another value, than ’null’.

Returns:

- Start position (in MB) of a partition.

5.16.47 CFDiskIO::getPartitionNumber

Description: Returns the physical partition number of a virtual partition.
Parameter:

- vDisk: Internal disk number (for accessing the disk information in the array).
- vPart: the virtual partition number, that is used to build the variable name to access the associative array.
- return: This will returned in case of an error, if set to another value, than 'null'.

Returns:

- Physical partition number of a virtual partition.

5.16.48 CFDiskIO::getPartitionSize

Description: Returns the size of a virtual partition.
Parameter:

- vDisk: Internal disk number (for accessing the disk information in the array).
- vPart: the virtual partition number, that is used to build the variable name to access the associative array.
- return: This will returned in case of an error, if set to another value, than 'null'.

Returns:

- Size (in MB) of a virtual partition.

5.16.49 CFDiskIO::getPartitionDev

Description: Returns the device of a partition (eg. /dev/sda1).
Parameter:

- vDisk: Internal disk number (for accessing the disk information in the array).
- vPart: the virtual partition number, that is used to build the variable name to access the associative array.

Returns:

- Device of a partition (eg. /dev/sda1).

5.16.50 CFDiskIO::getDevBypDiskpPart

Description: Returns device string with pDisk and partition number correctly combined.
Parameter:

- pDisk: Physical disk is written to (e.g. /dev/hda).
- pPart: Physical partition number (e.g. 1) or false (if there is no number in the dev).

Returns:

- Device string.

5.16.51 CFDiskIO::getPartitionDevs

Description: Returns an array with all partition devices (/dev/sdXY) as key and value.
Parameter:

- vDisk: Internal disk number (for accessing the disk information in the array).

Returns:

- Array with all partition devices (/dev/sdXY) as key and value.

5.16.52 CFDiskIO::getPartitionEnd

Description: Returns the end position (in MB) of a partition.
Parameter:

- vDisk: Internal disk number (for accessing the disk information in the array).
- vPart: the virtual partition number, that is used to build the variable name to access the associative array.
- return: This will returned in case of an error, if set to another value, than 'null'.

Returns:

- End position (in MB) of a partition.

5.16.53 CFDiskIO::getPartitionType

Description: Returns the type of a partition.
Parameter:

- vDisk: Internal disk number (for accessing the disk information in the array).
- vPart: the virtual partition number, that is used to build the variable name to access the associative array.
- return: This will returned in case of an error, if set to another value, than 'null'.

Returns:

- Type of a partition.

5.16.54 CFDiskIO::getPartitionFileSystem

Description: Returns the file system of a partition.
Parameter:

- vDisk: Internal disk number (for accessing the disk information in the array).
- vPart: the virtual partition number, that is used to build the variable name to access the associative array.
- return: This will returned in case of an error, if set to another value, than 'null'.

Returns:

- FileSystem of a partition.

5.16.55 CFDiskIO::setPartitionFileSystem

Description: Sets the file system of a partition.
Parameter:

- vDisk: Internal disk number (for accessing the disk information in the array).
- vPart: the virtual partition number, that is used to build the variable name to access the associative array.
- fs: File system (e.g. ext4)

5.16.56 CFDiskIO::virtualDelPartition

Description: Deletes a partition from the internal array.
Parameter:

- vDisk: Internal disk number (for accessing the disk information in the array).
- vPart: the virtual partition number, that is used to build the variable name to access the associative array.

5.16.57 CFDiskIO::virtualDeleteDisk

Description: Deletes a (RAID) disk and corrects the disk array.
Parameter:

- vDisk: Internal disk number (for accessing the disk information in the array).

5.16.58 CFDiskIO::getvPartBypPart

Description: Returns the virtual partition number searched by the physical partition number.
Parameter:

- vDisk: Internal disk number (for accessing the disk information in the array).
- pPart: The physical partition number

Returns:

- Virtual partition number or false, if none of the virtual partitions matches the physical partition number.

5.16.59 CFDiskIO::getLogicalpParts

Description: Builds an array with all physical partition numbers of the logical partitions.
Parameter:

- vDisk: Internal disk number (for accessing the disk information in the array).

Returns:

- Array with all physical partition numbers of the logical partitions.

5.16.60 CFDiskIO::correctLogical

Description: Corrects the order of the logical partitions after deleting a physical partition.
Parameter:

- vDisk: Internal disk number (for accessing the disk information in the array).
- pPart: The physical partition number to delete.

5.16.61 CFDiskIO::devNrExists

Description: Checks if a certain (physical) partition number exists.
Parameter:

- vDisk: Virtual (internally used) device number.
- devNr: Device numer you want to check (e.g. 2 for /dev/sda2)

Returns:

- true, when the (physical) partition number exists on the disk.

5.16.62 CFDiskIO::getBiggestLowestValueOf

Description: Gets the biggest or lowest value from all partitions of a given type.
Parameter:

- vDisk: Virtual (internally used) disk number.
- partType: Type of the partition (CFDiskIO::PT_PRIMARY, CFDiskIO::PT_EXTENDED, CFDiskIO::PT_LOGICAL, CFDiskIO::PT_EFI) or * for all types.
- varType: Variable in the associative array ($this->wantedPartitioning[$vDisk][$vPart][$varType]) to check.
- biggest: If set to true, the biggest value is searched, otherwise false.

Returns:

- Biggest or lowest value from all partitions of a given type.

5.16.63 CFDiskIO::getBiggestValueOf

Description: Gets the biggest value from all partitions of a given type.
Parameter:

- vDisk: Virtual (internally used) disk number.
- partType: Type of the partition (CFDiskIO::PT_PRIMARY, CFDiskIO::PT_EXTENDED, CFDiskIO::PT_LOGICAL, CFDiskIO::PT_EFI)
- varType: Variable in the associative array ($this->wantedPartitioning[$vDisk][$vPart][$varType]) to check.

Returns:

- Biggest value from all partitions of a given type.

5.16.64 CFDiskIO::getLowestValueOf

Description: Gets the lowest value from all partitions of a given type.
Parameter:

- vDisk: Virtual (internally used) disk number.
- partType: Type of the partition (CFDiskIO::PT_PRIMARY, CFDiskIO::PT_EXTENDED, CFDiskIO::PT_LOGICAL, CFDiskIO::PT_EFI)
- varType: Variable in the associative array ($this->wantedPartitioning[$vDisk][$vPart][$varType]) to check.

Returns:

- Lowest value from all partitions of a given type.

5.16.65 CFDiskIO::virtualAddPartition

Description: Creates a new (virtual) partition on a disk.
Parameter:

- dev: The device (e.g. /dev/sda2).
- start: Start position (in MB) of the partition to create.
- end: End position (in MB) of the partition to create.
- type: type of the partition (CFDiskIO::PT_PRIMARY, CFDiskIO::PT_EXTENDED, CFDiskIO::PT_LOGICAL, CFDiskIO::PT_EFI)

Returns:

- Physical partition number of the new partition.

5.16.66 CFDiskIO::virtualAddDisk

Description: Adds a new virtual disk.
Parameter:

- dev: Device name of the new drive (e.g. /dev/md0)
- size: Size in MB of the new drive.

Returns:

- vDisk of the new created virtual disk.

5.16.67 CFDiskIO::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.

Returns:

- The RAID device (e.g. /dev/md0) the physical partition belongs to or false, if no belonging RAID was found.

5.16.68 CFDiskIO::FDISK_dev2VDiskVPart_Legacy

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.

5.16.69 CFDiskIO::collidesWithPartitionOfType

Description: Checks if a partition (to be created) defined by start and end position would collide with existing partitions of given type.
Parameter:

- vDisk: Virtual (internally used) device number.
- type: type of the partition (primary, extended, logical, efipart, or -1 to match all partition types)
- start: start position for the partition (to be created)
- end: end position for the partition (to be created)

Returns:

- true, if the partition to create would collide, otherwise false.

5.16.70 CFDiskIO::isFreeSpaceBetween

Description: Checks if there is a free space between $start and $end and the type of partition could be created there.
Parameter:

- vDisk: Virtual (internally used) device number.
- type: type of the partition (primary, extended, logical, efipart)
- start: start position for the search
- end: end position for the search
- addErrorMsg: Set to true, if the error messages should be added.

Returns:

- true, if the there is free space, otherwise false.

5.16.71 CFDiskIO::getPossiblePartitionTypesBetween

Description: Get a list of partition types that can be created in a range on a disk.
Parameter:

- vDisk: Internal disk number (for accessing the disk information in the array).
- start: Start position (in MB) of the partition to create.
- end: End position (in MB) of the partition to create.

Returns:

- List of partition types that can be created in a range on a disk.

5.16.72 CFDiskIO::isNewPartitionOfTypeCreatable

Description: Checks if a new partition from a certain type can be created.
Parameter:

- vDisk: Internal disk number (for accessing the disk information in the array).
- type: type of the partition (CFDiskIO::PT_PRIMARY, CFDiskIO::PT_EXTENDED, CFDiskIO::PT_LOGICAL)
- addErrorMsg: Set to true, if the error messages should be added.

Returns:

- true, if a new partition can be created, otherwise false.

5.16.73 CFDiskIO::getCreatablePartitionTypes

Description: Creates an array with the types of partitions that could be created on a disk.
Parameter:

- vDisk: Internal disk number (for accessing the disk information in the array).

Returns:

- Associative array with types of partitions that could be created on a disk.

5.16.74 CFDiskIO::mayLogicalPartitionsBeChanged

Description: Checks if the logical partitions may be changed on the given disk. If one logical partition on the disk is assigned to a RAID, none of the other logical partitions may be removed or added, because the numbering of the partitions will change afterwards and bring the RAID into a faulty state.
Parameter:

- vDisk: Internal disk number (for accessing the disk information in the array).

Returns:

- true, if logical partitions may be removed or added on the disk, otherwise false.

5.16.75 CFDiskIO::mayPartitioningBeChanged

Description: Checks, if the whole disk or at least one partition on the disk is used in a RAID.
Parameter:

- vDisk: Internal disk number (for accessing the disk information in the array).

Returns:

- true, if the partitioning of the disk may be changed, otherwise false.

5.16.76 CFDiskIO::getNextFreePhysicalVirtualPartitionNumber

Description: Gets the next physical and virtual partition numbers for a new partition of given type.
Parameter:

- vDisk: Virtual (internally used) disk number.
- newPartNr: (physical) partition number for the partition to create.
- type: type of the partition (primary, extended, logical)
- start: Start position of the partition.
- end: End position of the partition.

Returns:

- Virtual (internally used) partition number.

5.16.77 CFDiskIO::nextLogicalDevNr

Description: Gets the next free logical (physical) partition number.
Parameter:

- vDisk: Virtual (internally used) disk number.

Returns:

- Next free logical (physical) partition number.

5.16.78 CFDiskIO::nextPrimaryDevNr

Description: Gets the next free primary (physical) partition number.
Parameter:

- vDisk: Virtual (internally used) disk number.

Returns:

- Next free logical (physical) partition number or -1, if there are no free partition numbers.

5.16.79 CFDiskIO::nextPrimaryDevNr

Description: Gets the next free EFI (physical) partition number.
Parameter:

- vDisk: Virtual (internally used) disk number.

Returns:

- Next free EFI (physical) partition number or -1, if there are no free partition numbers.

5.16.80 CFDiskIO::getWantedPartitioning

Description: Returns the wantedPartitioning array.
Returns:

- wantedPartitioning array.

5.16.81 CFDiskIO::sortDiskKeyByDev

Description: Sorts the elements with numeric keys (disks) of the input array by the subkey 'dev'. The numeric keys will be ascending and in the same order than the ascending subkey 'dev' with /dev/mdX at the end.
Parameter:

- in: Input array of a disk.

Returns:

- Array with sorted disks.

5.16.82 CFDiskIO::sortPartitionKeyByStart

Description: Sorts the elements with numeric keys of the input array by the subkey 'start' and copies all other elements unchanged. The numeric keys will be ascending and in the same order than the ascending subkey 'start'.
Parameter:

- in: Input array ($disk array with information about all parameters (disk info and partitions)) of a disk.

Returns:

- Array with sorted partitions.

5.16.83 CFDiskIO::getRaidLevelNumbers

Description: Returns an array with valid RAID levels.
Returns:

- Array with valid RAID levels.

5.16.84 CFDiskIO::isRaidLevelNumberValid

Description: Checks, if a number is a valid RAID level.
Parameter:

- level: RAID level to check.

Returns:

- true, if the input number is a valid RAID level.

5.16.85 CFDiskIO::getRaidDevsBuildingDisk

Description: Returns an array with all disk or partition devices building the RAID of the disk (if it is a RAID).
Parameter:

- vrDisk: Virtual (internally used) disk number.

Returns:

- Array with all disk or partition devices building the RAID of the disk (if it is a RAID) or empty array if the disk is no RAID.

5.16.86 CFDiskIO::getvrDevNrByrDev

Description: Returns the (virtual) number of the given device building the RAID.
Parameter:

- vrDisk: Virtual (internally used) disk number of the RAID.
- dev: Device to search in the RAID.

Returns:

- Virtual number of the given device building the RAID or false, if the device is not part of the RAID.

5.16.87 CFDiskIO::doesDiskHavePartitionsLockedByRaid

Description: Checks if the disk has at least one partition that was assigned to a RAID.
Parameter:

- vDisk: Virtual (internally used) disk number.

Returns:

- true, if there is at least one partition that was assigned to a RAID on the disk, otherwise false.

5.16.88 CFDiskIO::addDevToRaid

Description: Adds a device (disk or partition) to the list of devices building the RAID.
Parameter:

- vrDisk: Virtual (internally used) RAID disk number.
- dev: Device to add to the RAID.

Returns:

- true, if the device could be added, otherwise false.

5.16.89 CFDiskIO::delDevFromRaid

Description: Removes a device (disk or partition) from the list of devices building the RAID.
Parameter:

- vrDisk: Virtual (internally used) RAID disk number.
- dev: Device to remove from the RAID.

Returns:

- true, if the device could be removed, otherwise false.

5.16.90 CFDiskIO::updateRAIDSize

Description: Updates the usable size of the RAID
Parameter:

- vrDisk: Virtual (internally used) disk number.

Returns:

- true, if the RAID size could be calculated, otherwise false.

5.16.91 CFDiskIO::getRaidCompleteParameters

Description: Writes the constraints for building a RAID of a given level to the parameter variables.
Parameter:

- level: RAID level number.
- minRaidDrives: The minimum amount of devices for building the RAID.
- raidMultipleOf: The amount of devices building the RAID must be a multiple of this value.
- maxRaidDrives: Maximum amount of devices building the RAID.

Returns:

- true, if there are parameters of the given RAID level, otherwise false.

5.16.92 CFDiskIO::isRaidComplete

Description: Checks, if there are too less or too much devices building the RAID or if the amount is not a multiple of needed devices.
Parameter:

- vrDisk: Virtual (internally used) RAID disk number.

Returns:

- true, if all requirements for building the RAID are met or it is not a RAID.

5.16.93 CFDiskIO::maximumDevicesForRaidReached

Description: Checks, if the the maximum amount of devices building the RAID is reached.
Parameter:

- vrDisk: Virtual (internally used) RAID disk number.

Returns:

- true, if the maximum amount is reached or it is not a RAID.

5.16.94 CFDiskIO::areAllRaidsComplete

Description: Checks, if all RAID disk are complete (needed numbers of disks/partitions were added).
Returns:

- true, if all requirements for building all RAIDs are met.

5.16.95 CFDiskIO::getRaidDevsBuildingDiskAmount

Description: Returns the amount of the devices building the RAID.
Parameter:

- vrDisk: Virtual (internally used) RAID disk number.

Returns:

- Amount of the devices building the RAID.

5.16.96 CFDiskIO::setRaidLevel

Description: Sets the RAID level on a new RAID disk.
Parameter:

- vrDisk: Virtual (internally used) disk number.
- level: RAID level number.

Returns:

- true, if the RAID level is valid, otherwise false.

5.16.97 CFDiskIO::getDiskSize

Description: Returns the RAID level of RAID disk.
Parameter:

- vrDisk: Internal disk number (for accessing the disk information in the array).

Returns:

- RAID level or false, if it is not a RAID.

5.16.98 CFDiskIO::virtualCreateRaidDisk

Description: Creates a new RAID disk.
Parameter:

- dev: Device name for the RAID (eg. /dev/md0)
- level: RAID level.

Returns:

- Virtual RAID disk number, if the device string is suitable for a RAID, otherwise false.

5.16.99 CFDiskIO::virtualDeleteRaidDisk

Description: Deletes a RAID disk and removes the RAID locks from all devices building it.
Parameter:

- dev: Device name for the RAID (eg. /dev/md0)

Returns:

- Virtual RAID disk number, if the device string is suitable for a RAID, otherwise false.

5.16.100 CFDiskIO::getDrivesAndPartitions

Description: Generates an array that contains all disks and partitions of a given client with the disk devices as keys and the information as values.
Parameter:

- devFilter: 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.

5.16.101 CFDiskIO::getUnusedMDs

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

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

5.16.102 CFDiskIO::getUnusedDiskDev

Description: Returns an associative array with the unused device names (e.g. /dev/sda, /dev/sdb, ...) as key and value.
Returns:

- Associative array with the unused device names (e.g. /dev/sda, /dev/sdb, ...) as key and value.

5.16.103 CFDiskIO::getUsedMDs

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

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

5.16.104 CFDiskIO::getPartDevs

Description: Returns an array with the partitions (/dev/hda1, /dev/hda2, ...) of a disk or all disks.
Parameter:

- vDisk: Virtual (internally used) device number or -1, if all partitions on all devices should be listed.
- excludeType: type of partitions, not to return.
- includeTypes: Array with types of partitions, to return.

Returns:

- Array with matching partitions.

5.16.105 CFDiskIO::getNotFS

Description: Generates and returns an array with the list of not file systems (free space or unformated partitions).
Returns:

- Associative array with the list of not file systems as key and value.

5.16.106 CFDiskIO::getSupportedFS

Description: Generates and returns an array with the list of supported file systems.
Returns:

- Associative array with the list of supported file systems as key and value.

5.16.107 CFDiskIO::swapFilesystems

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

- Array with the filesystems usable for swapping.

5.16.108 CFDiskIO::installFilesystems

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

- Array with the filesystems usable for installation.

5.16.109 CFDiskIO::fdiskAddFstab

Description: Adds a new entry to the fstab array.
Parameter:

- dev: Device to mount (e.g. /dev/hda1)
- mountpoint: Location where to mount the device (e.g. /mnt/hda1)
- parameter: Mount parameter.

Returns:

- true, if the input parameter are valid, otherwise false.

5.16.110 CFDiskIO::fdiskGetEntry

Description: Gets an entry to from the fstab array. Can be called more times to get all entries.
Parameter:

- dev: Device to mount (e.g. /dev/hda1)
- mountpoint: Location where to mount the device (e.g. /mnt/hda1)
- parameter: Mount parameter.

Returns:

- true, if an entry to from the fstab array could be fetched from the array, otherwise false.

5.16.111 CFDiskIO::fdiskDelFstabEntry

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

- dev: Device to mount (e.g. /dev/hda1)

Returns:

- true, if the entry could be found, otherwise false.

5.16.112 CFDiskIO::adjustFstabParam

Description: Adjusts 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

5.16.113 CFDiskIO::genManualFstab

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

- mntPrefix: Prefix to set before the mountpoint (e.g. /mnt/m23root/)
- sourceName: Name of the client's sources list.

Returns:

- true, if fstab data in new format is used on the client, otherwise false.

5.16.114 CFDiskIO::FDISK_getOldStyleFstabArrayForHalfSister

Description: Gets the fstab entries in the old style array format that is used by HS_sysAddFstabEntries.
Returns:

- Fstab entries in the old style array format.

5.16.115 CFDiskIO::FDISK_genManualFstab

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

- fstab: Array with the fstab entries.
- mntPrefix: Prefix to set before the mountpoint (e.g. /mnt/m23root/)
- sourceName: The name of the package source list

5.16.116 CFDiskIO::fdiskGetFstabArray

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

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

5.16.117 CFDiskIO::getPartitionStepsArray

Description: Gets the partitionSteps array of a client.
Returns:

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

5.16.118 CFDiskIO::findFstabMountPointByDev

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

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

Returns:

- Mountpoint for the device or false, if there is no matching.

5.16.119 CFDiskIO::getPrimaryOrEfiPartitionType

Description: Returns a primary (on BIOS booted systems) partition type or an EFI (on UEFI booted systems).
Returns:

- Primary (on BIOS booted systems) or an EFI (on UEFI booted systems).

5.16.120 CFDiskIO::changeToEfiPartitionTypeIfInUefiMode

Description: Changes a given primary partition type to an EFI partition type on UEFI booted systems.
Parameter:

- var: Variable with the contents that may be changed.

Returns:

- The maybe unchanged type for isFreeSpaceBetween.

5.16.121 CFDiskIO::getNVMeMknodCommand

Description: Generates the mknod commands for creating the NVMe devices.
Returns:

- mknod commands for creating the NVMe devices.

5.16.122 CFDiskIO::getMknodCommand

Description: Generates the mknod command for a given /dev/sdX(Y) device (disk or partition).
Parameter:

- dev: The device (e.g. /dev/sda5) to created the mknod command for.

Returns:

- mknod command with the parameter matching the given /dev/sdX(Y).

5.17 ./inc/CFDiskTest.php

Author: Hauke Goos-Habermann (HHabermann@pc-kiel.de)
Description: Class for testing partitioning and formating functions.

5.17.1 CFDiskTest::getTestDiskDev

Description: Gets the disk device that is used for testing.
Returns:

- Disk device that is used for testing.

5.17.2 CFDiskTest::nextTurn

Description: Is called at the beginning of the next testing turn. Chooses a disk for testing, if there are more disks given.
Returns:

- true, if there are more disks for testing and a new disk could be chosen.

5.17.3 CFDiskTest::addLineToReplay

Description: Adds a line to the run log file.
Parameter:

- line: The line to add.

Returns:

- true on sucessfully writing the line, otherwise false.

5.17.4 CFDiskTest::rand

Description: Reads a random value from the replay file, if in replay mode or puts a calculated random value to the replay file.
Parameter:

- min: Minimal random value to generate.
- max: Maximal random value to generate.

Returns:

- Calculated random value or value read from the replay file.

5.17.5 CFDiskTest::getTestDiskvDisk

Description: Gets the vDisk for the testing fisk.
Returns:

- vDisk for the testing fisk.

5.17.6 CFDiskTest::getRandomPartitionDev

Description: Gets the a random device string for a partition on the test disk.
Returns:

- Random device string for a partition on the test disk.

5.17.7 CFDiskTest::getRadomStartEnd

Description: Gets random start / end positions (in MB) for e.g. creating new partitions on the test disk.

5.17.8 CFDiskTest::checkBothSetKey

Description: Checks if a key exists in virtual calculation and actual client partitioning.
Parameter:

- key: The key to check for.

Returns:

- true if the key exists in both arrays.

5.17.9 CFDiskTest::checkConsistencyDiskKey

Description: Checks for consistancy of $vDisk keys between virtual calculation and actual client partitioning. Differences will be inserted into $this->diffConsistency.
Parameter:

- vDisk: Internal disk number (for accessing the disk information in the array).
- key: The key to check for.
- bothUnsetAllowed: Set to true, if both values may be unset (used for the loop).
- tolerance: Maximal allowerd difference between the (numeric) values.

Returns:

- true if virtual calculation and actual client value are equal and both present, otherwise false.

5.17.10 CFDiskTest::checkConsistencyvDiskvPartKey

Description: Checks for consistancy of $vDisk and $vPart keys between virtual calculation and actual client partitioning. Differences will be inserted into $this->diffConsistency.
Parameter:

- vDisk: Internal disk number (for accessing the disk information in the array).
- vPart: Internal partition number in a disk (for accessing the disk information in the array).
- key: The key to check for.
- tolerance: Maximal allowerd difference between the (numeric) values.

Returns:

- true if virtual calculation and actual client value are equal and both present, otherwise false.

5.17.11 CFDiskTest::checkPartitionConsistency

Description: Checks for consistancy of the partioning between virtual calculation and actual client partitioning. Exists the script, if it is not consistent.

5.17.12 CFDiskTest::getDiskArrayFromClient

Description: Returns the disk array with all drives and their partitions.
Returns:

- Disk array with all drives and their partitions.

5.17.13 CFDiskTest::getDiskDevsFromClient

Description: Returns an array with all disk devices (/dev/sdX) as key and value from the current client.
Returns:

- Array with all disk devices (/dev/sdX) as key and value.

5.17.14 CFDiskTest::getPartInfoFromClient

Description: Fetches current partitioning and formating of the current client from a given disk.
Parameter:

- dev: The disk device (e.g. /dev/sda).

Returns:

- Array with current partitioning and formating of the current client from a given disk.

5.17.15 CFDiskTest::getRaidsFromClient

Description: Fetches current RAID information from the current client.
Parameter:

- rDevs: Array to store the devices in that build the RAIDs.
- vDisk: Last used vDisk number (from the calling function)

Returns:

- Array with the current RAID information from the current client.

5.17.16 CFDiskTest::executePartedCommands

Description: Executes all partitioning and formating commands on the given client.

5.17.17 CFDiskTest::createPartition

Description: Creates a new partition on a disk (if possible) and checks for consistancy of the partioning between virtual calculation and actual client partitioning.
Parameter:

- dev: The device (e.g. /dev/sda).
- start: Start position (in MB) of the partition to create.
- end: End position (in MB) of the partition to create.
- type: type of the partition (CFDiskIO::PT_PRIMARY, CFDiskIO::PT_EXTENDED, CFDiskIO::PT_LOGICAL)

5.17.18 CFDiskTest::randomCreatePartition

Description: Tries to create a new partition on the test disk with random type and start and end position.

5.17.19 CFDiskTest::deletePartition

Description: Deletes a partition from a disk and checks for consistancy of the partioning between virtual calculation and actual client partitioning.
Parameter:

- dev: The partition device (e.g. /dev/sda1).
- deleteBelongingRaid: If set to true, the RAID, the partition belongs to will be destroyed.

5.17.20 CFDiskTest::randomDeletePartition

Description: Tries to delete a randomly picked partition on the test disk.

5.17.21 CFDiskTest::formatPartition

Description: Formats a partition.
Parameter:

- dev: partition to format (e.g. /dev/hda1)
- fs: type of filesystem

5.17.22 CFDiskTest::randomFormatPartition

Description: Tries to format a randomly picked partition on the test disk.

5.17.23 CFDiskTest::getRandomFilesystem

Description: Returns an random filesytem.
Returns:

- Random filesytem.

5.17.24 CFDiskTest::getRandomPartitionType

Description: Returns an random partition type.
Returns:

- Random partition type.

5.17.25 CFDiskTest::createPartitionsForRaid

Description: Create partitions for usage in a RAID.

5.17.26 CFDiskTest::showDebugConsistencyClientVirtualArray

Description: Shows a debug information about the current state of virtual and client consistency.
Parameter:

- heading: The heading to show above the debug info.

5.17.27 CFDiskTest::randomPartTest

Description: Randomly creates, formates and deletes partitions of random size, type and with random file file systems.

5.17.28 CFDiskTest::getRandomRaidLevel

Description: Returns an random RAID level.
Returns:

- Random RAID level.

5.17.29 CFDiskTest::getRandomUnusedMD

Description: Returns a random unused MD.
Returns:

- Random unused MD.

5.17.30 CFDiskTest::getRandomUsedMD

Description: Returns a random used MD.
Returns:

- Random used MD.

5.17.31 CFDiskTest::deleteRaid

Description: Deletes a RAID disk.
Parameter:

- dev: Device name for the RAID (eg. /dev/md0)

Returns:

- true, if deleting of the RAID works, otherwise false.

5.17.32 CFDiskTest::areThereEnoughFreePartitionsToBuildTheRaid

Description: Checks, if there are enough unused partitions to build a RAID of a given level.
Returns:

- true, if there are enough free partitions, otherwise false.

5.17.33 CFDiskTest::randomCreateRaid

Description: Tries to create a new RAID with randomly choosen partitions and with random RAID level.

5.17.34 CFDiskTest::randomDeleteRaid

Description: Picks a random RAID and tries to delete it.

5.17.35 CFDiskTest::randomFormatRaid

Description: Picks a random RAID and tries to delete it.

5.17.36 CFDiskTest::randomRaidTest

Description: Randomly creates, formates and deletes partitions of random size, type and with random file file systems.

5.17.37 CFDiskTest::randomPartTest

Description: Randomly creates, formates and deletes partitions of random size, type and with random file file systems.

5.18 ./inc/CFirewall.php

Author: Hauke Goos-Habermann (HHabermann@pc-kiel.de)
Description: Class for managing firewall settings.

5.18.1 CFirewall::__construct

Description: Constructor for new CFDiskIO objects. The object holds all information about the partitioning (of a client and loads the values from the DB).
Parameter:

- in: Name or object of an existing client (to load) or data of an empty disk.

5.18.2 CFirewall::clearIPtablesSettings

Description: Clears all currently set iptables rules.

5.18.3 CFirewall::scriptFileExists

Description: Checks, if the iptables script file exists.
Returns:

- true, if the iptables script file exists, otherwise false.

5.18.4 CFirewall::rulesFileExists

Description: Checks, if the iptables rules file exists.
Returns:

- true, if the iptables rules file exists, otherwise false.

5.18.5 CFirewall::writeDefaultScript

Description: Writes the default iptables script, if there is no script.

5.18.6 CFirewall::getScript

Description: Gets the contents of the iptables script.
Returns:

- Contents of the iptables script.

5.18.7 CFirewall::putScript

Description: Writes script code in the iptables script.
Parameter:

- text: The iptables script code to write.

5.18.8 CFirewall::executeScript

Description: Executes the iptables script.

5.18.9 CFirewall::exportRules

Description: Exports the set iptables rules to a rule file (generated with iptables-save).

5.18.10 CFirewall::activateInInterfaces

Description: Activates auto-loading of the iptables rules in the interfaces file.
Returns:

- true, if the auto-loading is not allready active, otherwise false.

5.18.11 CFirewall::isActive

Description: Checks, if auto-loading of the iptables rules in the interfaces file is active.
Returns:

- true, if the auto-loading is active, otherwise false.

5.18.12 CFirewall::activate

Description: Activates the firewall.
Returns:

- true, if the firewall was activated, otherwise false.

5.18.13 CFirewall::deactivate

Description: Deactivates the firewall.
Returns:

- true, if the firewall was deactivated, otherwise false.

5.18.14 CFirewall::getSatusIconHTML

Description: Generates a HTML status icon to indicate the status of the firewall.

5.18.15 CFirewall::show

Description: Shows a dialog for (de)activating the firewall and changing the rules.

5.19 ./inc/CFreeIPA.php

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

5.19.1 CFreeIPA::__construct

Description: Constructor for new CFreeIPA objects.
Parameter:

- serverName: Name of the LDAP server

5.19.2 CFreeIPA::safeJSONString

Description: Makes the input string safe for using it in a JSON query.
Parameter:

- in: Input string.

Returns:

- Safe made string.

5.19.3 CFreeIPA::curlInit

Description: Prepares curl to connect to the FreeIPA server.

5.19.4 CFreeIPA::curlClose

Description: Closes the connection to the FreeIPA server.

5.19.5 CFreeIPA::curlExec

Description: Executes a command on the FreeIPA server via the JSONRPC API.
Parameter:

- post: Data (urlencoded or JSON) to send via POST to the FreeIPA server.
- json: Set to true, if the post data is in JSON format.

Returns:

- Associative array, if JSON is active, otherwise direct output of curl.

5.19.6 CFreeIPA::curlLogin

Description: Logins into the FreeIPA server via the JSONRPC API.
Returns:

- true, if login was successfully, false otherwise.

5.19.7 CFreeIPA::loadFreeIPAConfigFile

Description: Loads the FreeIPA configuration from CFreeIPA::FREEIPA_CONFIG_CACHE_FILE.
Returns:

- true, if the config file could be read, false otherwise.

5.19.8 CFreeIPA::getDomain

Description: Gets the domain that the FreeIPA server manages.
Returns:

- Domain that the FreeIPA server manages

5.19.9 CFreeIPA::getRealm

Description: Gets the Kerberos realm that the FreeIPA server manages.
Returns:

- Kerberos realm that the FreeIPA server manages

5.19.10 CFreeIPA::getServerFQDN

Description: Gets the FQDN of the FreeIPA server.
Returns:

- FQDN of the FreeIPA server

5.19.11 CFreeIPA::getm23AdminPassword

Description: Gets the password of the m23admin user on the FreeIPA server.
Returns:

- Password of the m23admin user on the FreeIPA server.

5.19.12 CFreeIPA::getServerIP

Description: Gets the IP of the FreeIPA server.
Returns:

- IP of the FreeIPA server

5.19.13 CFreeIPA::deleteConfigCacheFile

Description: Deletes the config file with cached information about the FreeIPA server.

5.19.14 CFreeIPA::generateConfigCacheFileFromFreeIPA

Description: Logs into the FreeIPA server via SSH and reads out needed information and stores them in the config file. This requires the user m23admin to be created in FreeIPA with admin right

5.19.15 CFreeIPA::saveFreeIPAConfigFile

Description: Writes/appends entries to the FreeIPA configuration file.
Parameter:

- append: Array with entries to append.

Returns:

- true, if the config file could be written, false otherwise.

5.19.16 CFreeIPA::__destruct

Description: Destructor for a CFreeIPA object.

5.19.17 CFreeIPA::getBash_checkRootAndKerberosTicket

Description: Generates BASH code, to check, if the code is run as root and if a valid Kerberos ticket is present.
Returns:

- Generated BASH code

5.19.18 CFreeIPA::getBash_addAdminRole

Description: Generates BASH code, to assign the admin role to an existing user.
Parameter:

- login: Login name

Returns:

- Generated BASH code

5.19.19 CFreeIPA::getBash_getKerberosTicketOnLogin

Description: Generates BASH code, to fetch a kerberos ticket on login automatically.
Parameter:

- login: Login name

Returns:

- Generated BASH code

5.19.20 CFreeIPA::getBash_createUser

Description: Generates BASH code, to create a user in FreeIPA.
Parameter:

- login: Login name
- firstName: Forename
- lastName: Familyname
- password: Password
- shell: Shell to use when logged in
- sshPubKey: Public SSH key to allow passwordless login

Returns:

- Generated BASH code

5.19.21 CFreeIPA::getBash_deleteUser

Description: Generates BASH code, to delete a user from FreeIPA.
Parameter:

- login: Login name

Returns:

- Generated BASH code

5.19.22 CFreeIPA::getBash_setPasswordExpiration

Description: Generates BASH code, to set expiration date for the password (far into the future so that you are not prompted to change the password).
Parameter:

- login: Login name
- password: Password
- date: Expiration date.

Returns:

- Generated BASH code

5.19.23 CFreeIPA::getFreeIPIntegrateCommands

Description: Generates commands to create an administrator account, that can be used by m23, on the FreeIPA server.
Returns:

- BASH commands to create an administrator account, that can be used by m23, on the FreeIPA server.

5.19.24 CFreeIPA::GUI_showConfigureServer

Description: Shows a dialog to integrate an existing FreeIPA server into m23.

5.19.25 CFreeIPA::isAvailable

Description: Checks, if a FreeIPA server is integrated into m23, by checking the presence ofthe configuration file on the m23 server. A presence doesn't mean, that the FreeIPA server is currently running.
Returns:

- true, if the config file is present, otherwise false.

5.19.26 CFreeIPA::createCFreeIPAObject

Description: Tries to create a CFreeIPA object. This only works if there is a configuration file on the m23 server.
Parameter:

- errorMsg: In case of an error the error message is written into this variable.

Returns:

- CFreeIPA object on success, otherwise false.

5.19.27 CFreeIPA::executeOnFreeIPA

Description: Runs a BASH script on the FreeIPA server.
Parameter:

- jobName: name of the job screen should show
- cmds: the commands of the script

Returns:

- Output of the script.

5.19.28 CFreeIPA::isOnline

Description: Checks, if the FreeIPA server, that was integrated into m23, is currently running.
Returns:

- true, if the FreeIPA server is running, false otherwise.

5.19.29 CFreeIPA::getUsedFreeIPAIPs

Description: Gets the IPs that are managed by the FreeIPA server.
Parameter:

- intIPs: Set to true, to return the IPs as int values.

Returns:

- IPs that are managed by the FreeIPA server.

5.19.30 CFreeIPA::checkCurlResult

Description: Checks the answer of a FreeIPA curl call.
Parameter:

- ret: Answer of a FreeIPA curl call.

Returns:

- true, if there are no errors in the query, otherwise false.

5.19.31 CFreeIPA::addComputer

Description: Adds an m23 client to FreeIPA as host and into FreeIPA's DNS.
Parameter:

- client: Name of the computer/client to add.
- clientIP: The IP of the client.
- mac: MAC address of the client.
- description: Description to store in FreeIPA.
- class: Host category
- locality: Locality where the computer is placed.
- location: Location where the computer is placed.
- platform: Platform (e.g. CPU, computer model, VM, etc.)
- os: The distribution, OS that the computer runs.

Returns:

- true, if the client was added successfully, otherwise (error) messages from the FreeIPA API.

5.19.32 CFreeIPA::setComputerBulkPassword

Description: Sets the bulk password of an m23 client in FreeIPA.
Parameter:

- client: Name of the computer/client to add.
- bulkPasswd: Bulk password for running ipa-client-install on the client.

Returns:

- true, if the bulk password was set successfully, otherwise (error) messages from the FreeIPA API.

5.19.33 CFreeIPA::addUser

Description: Creates a user in FreeIPA.
Parameter:

- login: Login name
- firstName: Forename
- lastName: Familyname
- password: Password
- shell: Shell to use when logged in
- sshPubKey: Public SSH key to allow passwordless login

Returns:

- true, if the user was added successfully, otherwise (error) messages from the FreeIPA API.

5.19.34 CFreeIPA::delUser

Description: Deletes a user from FreeIPA.
Parameter:

- login: Login name

Returns:

- true, if the user was deleted successfully, otherwise (error) messages from the ipa tool.

5.19.35 CFreeIPA::delComputer

Description: Deletes an m23 client form FreeIPA's host and DNS entries.
Parameter:

- client: Name of the computer/client to add.

Returns:

- true, if the client was deleted successfully, otherwise (error) messages from the ipa tool.

5.19.36 CFreeIPA::existUserGroup

Description: Checks, if a group is existing in FreeIPA.
Parameter:

- group: Name of the group to check.

Returns:

- true, if the group exists, otherwise false.

5.19.37 CFreeIPA::createUserGroup

Description: Creates a group in FreeIPA.
Parameter:

- group: Name of the group
- description: description of the group

Returns:

- true, if the group was created, otherwise false.

5.19.38 CFreeIPA::addLoginToGroup

Description: Adds an user to a group in FreeIPA.
Parameter:

- group: Name of the group
- login: Login name

Returns:

- true, if the login was added or is in the group already, otherwise false.

5.19.39 CFreeIPA::addGroupToUserGroup

Description: Adds a group to another group in FreeIPA.
Parameter:

- group: Name of the group to add to another group
- mainGroup: Name of the main group where the another group should be added

Returns:

- true, if the group was added or is in the group already, otherwise false.

5.19.40 CFreeIPA::createHostGroup

Description: Creates a host (m23 client) group in FreeIPA.
Parameter:

- group: Name of the group
- description: description of the group

Returns:

- true, if the group was created, otherwise false.

5.19.41 CFreeIPA::existHostGroup

Description: Checks, if a host group is existing in FreeIPA.
Parameter:

- group: Name of the group to check.

Returns:

- true, if the group exists, otherwise false.

5.19.42 CFreeIPA::addClientToHostGroup

Description: Adds an m23 client to a host group in FreeIPA.
Parameter:

- group: Name of the group

Returns:

- true, if the client was added or is in the group already, otherwise false.

5.19.43 CFreeIPA::addGroupToHostGroup

Description: Adds a host group to another host group in FreeIPA.
Parameter:

- group: Name of the host group to add to another host group
- mainGroup: Name of the main host group where the another host group should be added

Returns:

- true, if the host group was added or is in the host group already, otherwise false.

5.20 ./inc/CGPGSign.php

Author: Hauke Goos-Habermann (HHabermann@pc-kiel.de)
Description: Class for managing GPG keys and signing messages.

5.20.1 CGPGSign::__construct

Description: Constructor for new CGPGSign objects.
Parameter:

- mode: Save a (new) configuration file or load a (required) configuration file.

5.20.2 CGPGSign::getKeySelectionDialog

Description: Generates a dialog to choose the GPG used for signing the pools.

5.20.3 CGPGSign::showWarningAndDie

Description: Shows an "internal" warning message or a message for the m23 administrator and destroys the object afterwards.
Parameter:

- internalMsg: Internal error message text.

- userMsg: Warning message for the m23 administrator.

5.20.4 CGPGSign::exportPublicSignKey

Description: Exports the public key to the webserver directory.
Parameter:

- true, if the file was exported, othwerwise false.

5.20.5 CGPGSign::checkKey

Description: Checks, if the given GPG is valid as public and private key.
Parameter:

- privKeyID: ID of the GPG key.

Returns:

- : true, if the GPG is valid as public and private key, otherwise false.

5.20.6 CGPGSign::getKeyInfo

Description: Gets information about the used GPG key.
Returns:

- : Information about the used GPG key.

5.20.7 CGPGSign::gpgSignDetached

Description: Creates a detached signature file for a given private GPG key ID and input file.
Parameter:

- inFile: The file to create a signature for.
- outFile: The file with the detached signature.

Returns:

- : true, if the signature file was created and the input file exists, otherwise false.

5.20.8 CGPGSign::gpgSignClear

Description: Creates a clear text signature file for a given private GPG key ID and input file.
Parameter:

- inFile: The file to create a signature for.
- outFile: The file with the detached signature.

Returns:

- : true, if the signature file was created and the input file exists, otherwise false.

5.20.9 CGPGSign::hasConfigFile

Description: Checks, if the config file exists.
Returns:

- true, if the config file is present.

5.20.10 CGPGSign::loadConfigFile

Description: Loads the config file.

5.20.11 CGPGSign::writeConfigFile

Description: Writes the config file.

5.20.12 CGPGSign::setGPGID

Description: Sets the GPG ID to use.

5.20.13 CGPGSign::getGPGID

Description: Gets the GPG ID to use.
Parameter:

- allowReturnNull: Set to true, if NULL may be returned as ID (eg. there is no config file).

Returns:

- GPG ID to use or dies, if no ID is set.

5.20.14 CGPGSign::setStoreMode

Description: Sets the configuration file store or load mode.
Parameter:

- mode: Configuration file store or load mode.

5.20.15 CGPGSign::getStoreMode

Description: Gets the configuration file store or load mode.
Returns:

- mode: Configuration file store or load mode.

5.20.16 CClient::__destruct

Description: Destructor for a CGPGSign object.

5.21 ./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.

5.21.1 CHECK_login

Description: Checks, if a login name is valid.
Parameter:

- login: login name to check.
- allowEmpty: true, if empty is allowed.

Returns:

- true, if the login name is valid, otherwise false.

5.21.2 CHECK_ip

Description: Checks, if an IP is (syntactically) valid.
Parameter:

- ip: IP to check.

Returns:

- true, if the IP is valid, otherwise false.

5.21.3 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.

5.21.4 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.

5.21.5 CHECK_db2text

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

- val: String to use.

Returns:

- The safe made text.

5.21.6 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.

5.21.7 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.

5.21.8 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.

5.21.9 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.

5.21.10 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.

5.21.11 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.

5.21.12 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.

5.21.13 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.

5.21.14 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.

5.21.15 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.

5.21.16 checkIP

Description: checks if an ip is valid
Parameter:

- string: ip value to check

Returns:

- true if IP is valid, else false

5.21.17 checkMAC

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

- mac: MAC address to test.

Returns:

-

5.21.18 checkNetmask

Description: checks if a netmask is valid
Parameter:

- string: netmask value to check

Returns:

- true if netmask is valid, else false

5.21.19 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

5.21.20 checkFQDN

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

- string: string to check for special characters

5.21.21 checkFQDNOrIP

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

- string: string to check for special characters

5.21.22 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

5.22 ./inc/CIPRanges.php

Author: Hauke Goos-Habermann (HHabermann@pc-kiel.de)
Description: Class for handling ranges of IP addresses.

5.23 ./inc/CLdap.php

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

5.23.1 CLdap::__construct

Description: Constructor for new CLdap objects.
Parameter:

- serverName: Name of the LDAP server

5.23.2 CLdap::getFDAdminACLAttributes

Description: Get information (LDAP attribute, DN for user account and ACL) for creation and deletion of FusionDirectory administrators.
Parameter:

- login: the login name
- dnUser: DN where to store the FD admin account in the LDAP. (DN is written into this variable)
- dnACL: DN where to store the FD admin role in the LDAP. (DN is written into this variable)
- aclAttributes: ACL attribute that encode the admin role. (attribute is written into this variable)

5.23.3 CLdap::createFDAdmin

Description: Creates a new FusionDirectory administrator.
Parameter:

- login: the login name
- pwd: the unencrypted password
- cryptPw: set to true, if the password should be encrypted or false, if it's already encrypted.

Returns:

- true, if the administrator was created successfully, otherwise false.

5.23.4 CLdap::deleteFDAdmin

Description: Deletes a FusionDirectory administrator.
Parameter:

- login: the login name

Returns:

- true, if the administrator was deleted successfully, otherwise false.

5.23.5 CLdap::setStartRDN

Description: Sets the starting container (eg for searches or inserts).
Parameter:

- rdn: Relative DN where to start searching or inserting (eg. ou=people).

5.23.6 CLdap::getLDAPServer

Description: Gets the name of the LDAP server.
Returns:

- Name of the used LDAP server.

5.23.7 CLdap::clearStartRDN

Description: Clears the starting container RDN.

5.23.8 CLdap::getStartRDN

Description: Gets the starting container (eg for searches or inserts).
Parameter:

- rdn: Set the RDN to something else but global $this->RDN.

Returns:

- : Relative DN where to start searching or inserting (eg. ou=people).

5.23.9 CLdap::search

Description: Searches the LDAP.
Parameter:

- filter: Filter for only handling entries that match.
- attributes: Array with attributes to fetch.
- container: Container (subtree) in the DIT to search for the IDs. If empty the whole database will be searched.

Returns:

- Associative array with the LDAP search result or false in case of an error.

5.23.10 CLdap::dumpContainer

Description: Gets the contents of a LDAP container.
Parameter:

- container: Container (subtree) in the DIT to search for the IDs. If empty the whole database will be searched.

Returns:

- Associative array with the dump.

5.23.11 CLdap::getNextID

Description: Returns an array with free (posix user or group) IDs or a single free ID from the LDAP database.
Parameter:

- attribute: Name ID attribute (eg. uidnumber)
- container: Container (subtree) in the DIT to search for the IDs. If empty the whole database will be searched.

- filter: Filter for only handling entries that match.
- serverMinimumSettingsVar: The name of the variable that holds the minimum allowed ID value.
- amount: Amount of IDs to generate.
- minimumID: If set, this ID will be used as start ID instead of the ID set by server information.

Returns:

- Array with free IDs in the given amount or a single free ID as int.

5.23.12 CLdap::getFreeUserIDs

Description: Returns an array with free posix group IDs or a single free ID from the LDAP database.
Parameter:

- amount: Amount of IDs to generate.
- minimumID: If set, this ID will be used as start ID instead of the ID set by server information.

Returns:

- Array with free user IDs in the given amount or a single free user ID as int.

5.23.13 CLdap::getFreeGroupIDs

Description: Returns an array with free posix group IDs or a single free ID from the LDAP database.
Parameter:

- amount: Amount of IDs to generate.
- minimumID: If set, this ID will be used as start ID instead of the ID set by server information.

Returns:

- Array with free group IDs in the given amount or a single free group ID as int.

5.23.14 CLdap::simplifyLDAPEntries

Description: Simplifies the output of ldap_get_entries.
Parameter:

- info: Output of ldap_get_entries.

Returns:

- Array with simplified output of ldap_get_entries.

5.23.15 CLdap::filterLDAPEntries

Description: Filters the output of ldap_get_entries.
Parameter:

- info: Output of ldap_get_entries.
- filter: Only keys that match the filter will be present in the output.

Returns:

- Nummeric array with entries there the keys from ldap_get_entries match the filter.

5.23.16 CLdap::generateConfigCacheFileFromFusionDirectory

Description: Reads the FusionDirectory configuration from cn=config,ou=fusiondirectory,... and stores it into CLdap::FD_CONFIG_CACHE_FILE.

5.23.17 CLdap::loadFusionDirectoryConfigFile

Description: Loads the FusionDirectory configuration from CLdap::FD_CONFIG_CACHE_FILE.

5.23.18 CLdap::getFullDNByFdRDN

Description: Get a full DN by LDAP's base and a FusionDirectory RDN.
Parameter:

- fdConfigVar: Name of the FusionDirectory config variable holding the RDN.
- prefix: Prefix to add before the DN.
- rdn: Set the RDN to something else but global $this->RDN.

Returns:

- Full DN or false, if the FusionDirectory config variable isn't set.

5.23.19 CLdap::getKRB_realm

Description: Gets the Kerberos realm.
Returns:

- Kerberos realm or false, if not given.

5.23.20 CLdap::getKRB_server

Description: Gets the Kerberos server.
Returns:

- Kerberos server or false, if not given.

5.23.21 CLdap::getKRB_adminServer

Description: Gets the Kerberos admin server.
Returns:

- Kerberos admin server or false, if not given.

5.23.22 CLdap::getKRB_defaultDomain

Description: Gets the Kerberos default domain.
Returns:

- Kerberos default domain or false, if not given.

5.23.23 CLdap::getPasswordHash

Description: Hashes a password depending on FusionDirectory's default hash setting via MD5 or SSHA as LDAP/PAM expects it.
Parameter:

- pwd: Password to hash.

Returns:

- Hashed password.

5.23.24 CLdap::ldap_add

Description: Adds an object in the DIT.
Parameter:

- dn: Full DN where to add
- ldapAttributes: Attributes for LDAP object.
- errorMessage: Message to add in case of an error.

Returns:

- Output of ldap_add.

5.23.25 CLdap::ldap_mod_add

Description: Adds an attributes to an existing object.
Parameter:

- dn: Full DN where to add attributes
- ldapAttributes: Attributes to add to an LDAP object.
- errorMessage: Message to add in case of an error.

Returns:

- Output of ldap_mod_add.

5.23.26 CLdap::ldap_mod_del

Description: Removes attribute(s) from an existing object.
Parameter:

- dn: Full DN where to add attributes
- ldapAttributes: Attributes to remove from an LDAP object.
- errorMessage: Message to add in case of an error.

Returns:

- Output of ldap_mod_del.

5.23.27 CLdap::ldap_delete

Description: Removes an object from the DIT.
Parameter:

- dn: Full DN where to add attributes
- errorMessage: Message to add in case of an error.

Returns:

- Output of ldap_delete.

5.23.28 CLdap::ldap_rename

Description: Renames or moves a leaf in the DIT.
Parameter:

- oldDN: Full DN where the leaf is currently stored
- newRND: New (or same) RDN where the leave should be moved/renamed to.
- newParent: New parent DN (eg ou=people,dc=... without "uid=$account") where the leave should be moved to.
- errorMessage: Message to add in case of an error.

Returns:

- Output of ldap_delete.

5.23.29 CLdap::get2ndLDAPServer

Description: Gets the name of the school's LDAP server.
Returns:

- Name of the used school LDAP server.

5.23.30 CLdap::__destruct

Description: Destructor for a CLdap object.

5.24 ./inc/client$_{d}etails.php$

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

5.24.1 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.

5.24.2 CLIENT_DETAILS_endCategory

Description: Ends the previously opened icon section.

5.24.3 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.

5.24.4 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.

5.25 ./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,...

5.25.1 CLIENT_getLDAPProtocol

Description: Gets the LDAP access protocol (ldap or ldaps) by client's options.
Parameter:

- clientOptions: the client's options array

Returns:

- LDAP access protocol.

5.25.2 CLIENT_removeOldReportings

Description: Removes entries from the 'clientreporting' table that are too old.

5.25.3 CLIENT_updateReporting

Description: Stores clientname, amount of packages on the client, online status (0 = offline, 1 = pingable, 2 = SSH/HTTPs working), reboot status (0 = no reboot needed, 1 = reboot needed, but unfinished, 2 = reboot needed, but unfinished and delayed), update status (0 = no update job assigned, 1 = update job assigned, but unfinished, 2 = update job assigned, but unfinished and delayed) and the timestamp of the last finished update job into the table 'clientreporting'.

5.25.4 CLIENT_getDelayedRebootsSQL

Description: Prepares an SQL statement to get clients with delayed reboots or the delay status of a given client.
Parameter:

- clientName: Name of the client or false, if all clients that are delayed, should be found.
- maxAllowedDelay: The maximum tolerable amount of time since requesting the reboot. Only clients that exceed the limit are added to the array.

5.25.5 CLIENT_getClientsWithDelayedReboots

Description: Gets clients with delayed reboots.
Parameter:

- clientName: Name of the client.
- maxAllowedDelay: The maximum tolerable amount of time since adding the job. Only jobs that exceed the limit are added to the array.

5.25.6 CLIENT_isRebootDelayed

Description: Checks, if a client has a delayed reboot request.
Parameter:

- clientName: Name of the client.
- maxAllowedDelay: The maximum tolerable amount of time since adding the job. Only jobs that exceed the limit are added to the array.

5.25.7 CLIENT_fetchBASHScriptFromServerAndRun

Description: Generates BASH code to fetch the script from the m23 server and execute it.
Parameter:

- serverIP: IP, name, BASH variable name of the server, or NULL to get the server IP by getServerIP().
- script: Name of the PHP script under /m23/data+scripts/
- afterExport: BASH code to insert after the export line.

Returns:

- : BASH code to fetch the script from the m23 server and execute it

5.25.8 CLIENT_getClientIPArray

Description: Generates an associative array with the client names as keys and their IPs as values.
Returns:

- : Associative array with the client names as keys and their IPs as values.

5.25.9 CLIENT_rebootClientAfterJobsIfNecessary

Description: Reboots the client, if a reboot is necessary and the global server side option "m23ServerRebootClientAfterJobsIfNecess
is set.

5.25.10 CLIENT_unsetTimeStampForRebootingClientIfNOTNecessaryBASH

Description: Shows BASH code that detects, if a reboot is necessary and disables the server side check, if it's not.

5.25.11 CLIENT_rebootClientAfterJobsIfNecessaryBASH

Description: Shows BASH code for rebooting the client, if a reboot is necessary.

5.25.12 CLIENT_insertIntoClientlogs

Description: Adds an entry into the client's status log
Parameter:

- clientName: Name of the client.
- text: Status text
- status: One of CLIENTLOG_OK, CLIENTLOG_FAILURE or CLIENTLOG_UNKNOWN.

5.25.13 CLIENT_addClientlogsOk

Description: Adds an entry into the client's status log with "ok" status.
Parameter:

- clientName: Name of the client.
- text: Status text.

5.25.14 CLIENT_addClientlogsFailure

Description: Adds an entry into the client's status log with "failure" status.
Parameter:

- clientName: Name of the client.
- text: Status text.

5.25.15 CLIENT_addClientlogsUnknown

Description: Adds an entry into the client's status log with "unknown" status.
Parameter:

- clientName: Name of the client.
- text: Status text.

5.25.16 CLIENT_removeServerCache

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

- client: Name of the client.

5.25.17 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).

5.25.18 CLIENT_getNextFreeIp

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

- Next free IP address.

5.25.19 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.

5.25.20 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).

5.25.21 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.

5.25.22 CLIENT_stopLiveScreenRecording

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

- client: Name of the client.

5.25.23 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.

5.25.24 CLIENT_filterLinesFromLiveScreenRecording

Description: Filters out unwanted lines from the live log.
Parameter:

- in: Input line to give out or not.

Returns:

- Empty string, if the line should be filtered out or input line.

5.25.25 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.

5.25.26 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.

5.25.27 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.

5.25.28 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.

5.25.29 CLIENT_getDistribution

Description: Returns the distribution of a client.
Parameter:

- client: Name of the client.

Returns:

- Name of the distribution, the client uses.

5.25.30 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.

5.25.31 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.

5.25.32 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.

5.25.33 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.

5.25.34 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.

5.25.35 CLIENT_getAllClientNames

Description: Gets the names of all clients.
Returns:

- Array with the names of all clients.

5.25.36 CLIENT_getClientAmount

Description: Gets the amount of all clients.
Returns:

- Amount of all clients.

5.25.37 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.

5.25.38 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.

5.25.39 CLIENT_getClientID

Description: Returnes the ID of the calling client.

5.25.40 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.
- skipLoop: Skip loopdevices in the output array, if set to true.
- skipEQL: Skip eqldevices in the output array, if set to true.

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))

5.25.41 CLIENT_getSSHKeyorPasswordOptions

Description: Generates SSH options for using sshpass (password given) or SSH public key (no password given).
Parameter:

- password: Password to use, if it's not NULL or empty.
- sshPass: sshpass commands are written here, if a password is given.
- sshPubkeyOnlyOptions: SSH options are written here, if no password is given.

5.25.42 CLIENT_executeOnClientOrIPSimple

Description: Runs a script under a plain BASH with a given user.
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
- remoteUser: The remote user the script should be run under.
- temDir: Directory where to store the script on the remote host temporaryly.

Returns:

- Output of the script.

5.25.43 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.

5.25.44 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

5.25.45 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

Returns:

- Error message or empty string in case of success.

5.25.46 CLIENT_IPexists

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

- ip: IP to check

5.25.47 CLIENT_MACexists

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

- mac: MAC to check

5.25.48 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

5.25.49 CLIENT_getAskingParams

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

5.25.50 CLIENT_getParams

Description: returns database parameters of a special client
Parameter:

- clientName: name of the client

5.25.51 CLIENT_getClientStatus

Description: returnes the current client status
Parameter:

- client: name of the client

5.25.52 CLIENT_getProperty

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

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

5.25.53 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

5.25.54 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

5.25.55 CLIENT_acceptChanges

Description: removes packages or discards changes
Parameter:

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

5.25.56 CLIENT_showHardwareInfo

Description: prints a table with hardware informations
Parameter:

- client: name of zhe client

5.25.57 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

5.25.58 CLIENT_showWaitingJobs

Description: shows the waiting jobs for the client
Parameter:

- client: name of the client

5.25.59 CLIENT_setLastmodify

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

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

5.25.60 CLIENT_getNetmaskBits

Description: Gets the amount of set bits of a netmask.
Parameter:

- netmask: IPv4 netmask.

Returns:

- Amount of set bits of a netmask.

5.25.61 CLIENT_getNetmaskFromBitAmount

Description: Gets the netmask by the amount of set bits.
Parameter:

- bits: Amount if set bits of a netmask.

Returns:

- IPv4 netmask

5.25.62 CLIENT_getSubnet

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

- ip: ip address
- netmask: netmask

5.25.63 CLIENT_getBroadcast

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

- ip: ip address
- netmask: netmask

5.25.64 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

5.25.65 CLIENT_getIPbyName

Description: returns the ip from a selected clientname
Parameter:

- clientName: name of the client

5.25.66 CLIENT_getNamebyIP

Description: returns the clientname from a selected ip
Parameter:

- ip: ip of the client

5.25.67 CLIENT_getMACbyName

Description: returns the mac from a selected clientname
Parameter:

- clientName: name of the client

5.25.68 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)

5.25.69 CLIENT_backToRed

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

- clientName: name of the client

5.25.70 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.

5.25.71 CLIENT_recoverClientCombineJobs

Description: Recover a client: all client jobs are done again, status is set to 0, all identical jobs are combines.
Parameter:

- clientName: name of the client

5.25.72 CLIENT_wol

Description: wakes a client over the network
Parameter:

- clientName: name of the client

5.25.73 CLIENT_recalculateStatusBar

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

- clientName: name of the client

5.25.74 CLIENT_resetStatusBar

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

- clientName: name of the client

5.25.75 CLIENT_startInstall

Description: starts the installation on a client
Parameter:

- clientName: name of the client

5.25.76 CLIENT_resetAndInstall

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

- clientName: name of the client

5.25.77 CLIENT_getBootType

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

- clientName: name of the client

5.25.78 CLIENT_isrunning

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

- clientName: name of the client

5.25.79 CLIENT_reset

Description: resets a client
Parameter:

- clientName: name of the client

5.25.80 CLIENT_showLastLogError

Description: Prints the last error in the client log
Parameter:

- clientName: name of the client

5.25.81 CLIENT_showLog

Description: prints the log information of the client
Parameter:

- clientName: name of the client

5.25.82 CLIENT_getClientName

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

5.25.83 CLIENT_getAllOptions

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

- clientName: name of the client

5.25.84 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

5.25.85 CLIENT_getAllAskingOptions

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

5.25.86 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

5.25.87 CLIENT_options2HiddenForm

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

- options: name of the options array

5.25.88 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

5.25.89 CLIENT_getStatusimage

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

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

5.25.90 CLIENT_showStatusSelection

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

- client: the name of the client

5.25.91 CLIENT_listClientsInWarningMessageBox

Description: Lists clients in a warning message box with links to their detailed pages.
Parameter:

- clients: Array with the client names to list.
- header: Heading of the warning box.

Returns:

- true, if there are client names to list, false otherwise.

5.25.92 CLIENT_listClientsWithDelays

Description: Lists clients in a warning message box, that are having delayed jobs.

5.25.93 CLIENT_listClientsWithDelayedReboots

Description: Lists clients in a warning message box, that are having delayed reboot requests.

5.25.94 CLIENT_listCriticalClients

Description: lists clients with critical status'

5.25.95 CLIENT_criticalClientAmountInGroups

Description: Checks for critical clients in all groups and generates an associative array about groups with critical clients.
Returns:

- Associative array with all groupnames as key and the amount of the critical clients as value. Only groups with critical clients are added to the output array.

5.25.96 CLIENT_isInDebugMode

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

- clientName: name of the client

5.25.97 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

5.25.98 CLIENT_getStatusimage

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

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

5.25.99 CLIENT_generateHTMLDelayStatus

Description: Generates icons for showing the delayed status of a clinet.
Parameter:

- clientName: Name of the client.
- clientID: ID of the client.
- clientUpdateJobsMaxAllowedDelay: The maximum tolerable amount of running/waiting time (in minutes) to finish update jobs since adding it or 0, to disable.
- clientRebootMaxAllowedDelay: The maximum tolerable amount of time since requesting the reboot.

Returns:

- Associative array containg HTML code and tooltip

5.25.100 CLIENT_generateHTMLStatusBar

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

- clientName: Name of the m23 client
- id: ID of the m23 client
- status: Status (number, green, red, ...) of the client.
- vmRole: Role of the m23 client (host, guest, no virtualisation)
- vmSoftware: Number of the used virtualisation software.

5.25.101 CLIENT_showDebugSelection

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

- client: the name of the client

5.25.102 CLIENT_isInRescueMode

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

- clientName: the name of the client

5.25.103 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

5.25.104 CLIENT_isInDebugMode

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

5.25.105 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

5.25.106 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

5.25.107 CLIENT_getId

Description: returnes the id of a client
Parameter:

- clientName: the name of the client

5.25.108 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.

5.25.109 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)

5.25.110 CLIENT_showDelDialog

Description: Shows the dialog for deleting a client.

5.25.111 CLIENT_showAddDialog

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

- addType: defines the behaviour and appearance of the dialog

5.25.112 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.

5.25.113 CLIENT_getNames

Description: Returns an array with the names of all clients.
Parameter:

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

5.25.114 CLIENT_getIds

Description: Returns an array with all client IDs.
Parameter:

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

5.25.115 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

5.25.116 CLIENT_changeClient

Description: changes values of the clients

5.25.117 CLIENT_setAllParams

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

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

5.25.118 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.

5.25.119 CLIENT_isAssimilated

Description: Checks, if a client was assimilated.
Parameter:

- clientName: name of the client

Returns:

- true, if the client was assimilated. false, if the client was normaly installed.

5.25.120 CLIENT_isDedicatedAndReachable

Description: Checks, if a client is dedicated to this M23-Server and reachable.
Parameter:

- clientName: name of the client

Returns:

- true, if the client is dedicated and pingable. false, if any of the conditions is false.

5.25.121 CLIENT_generateHTMLDedicatedAndReachableStatus

Description: Generates HTML code and tooltip containing the status of the client showing if it's dedicated and reachable.
Parameter:

- online: One or more of the ONLINE_STATUS_* bits.

Returns:

- Associative array containg HTML code and tooltip

5.25.122 CLIENT_sendPlymouthMessage

Description: If Plymouth is running, display a message
Parameter:

- message: message to display

5.25.123 CLIENT_sendNotifySendMessage

Description: Send message to all users using notify-send
Parameter:

- message: message to display
- icon_path: icon shown
- must_inform: respect client settings or force message display? default false
- timeout_ms: timeout till message disapperas. default 0 (never)
- title: title of the message
- escape: escape the message for notify-send? default true
- timestamp: Show a timestamp at the beginning of the message. default true
- message: message to display
- icon_path: icon shown
- must_inform: respect client settings or force message display? default false

5.25.124 CLIENT_getSchool

Description: Get the school name a client belongs to.
Parameter:

- clientName: name of the client

Returns:

- Name of the school (if set), otherwise false.

5.25.125 CLIENT_isSchoolClient

Description: Checks, if a client is in school mode.
Parameter:

- clientName: name of the client

Returns:

- true, if the client is in school mode, otherwise false.

5.26 ./inc/Cm23AdminLister.php

Author: Maren Hachmann (maren@goos-habermann.de), Hauke Goos-Habermann (HHabermann@pc-kiel.de)
Description: Class for listing m23 administrators.

5.26.1 Cm23AdminLister::AdminExistsDB

Description: checks if an admin with the selected name exists in the DB and returns true if yes, otherwise false
Parameter:

- name: admin login name to check

Returns:

- True if the admin exists in the DB, else false

5.26.2 Cm23AdminLister::AdminExistsHt

Description: checks if an admin with the selected name exists in the m23 password file and returns true if yes, otherwise false
Parameter:

- name: admin login name to check

Returns:

- True if the admin exists in the password file, else false

5.26.3 Cm23AdminLister::CountAdmins

Description: counts the number of registered admins in m23 password file
Returns:

- : number of admins in m23 password file

5.26.4 Cm23AdminLister::ListAdmins

Description: lists all admins in m23 password file
Returns:

- : array of admin names in m23 password file

5.27 ./inc/Cm23Admin.php

Author: Maren Hachmann (maren@goos-habermann.de), Hauke Goos-Habermann (HHabermann@pc-kiel.de)
Description: Class for managing m23 administrators.

5.27.1 Cm23Admin::__construct

Description: Constructor for new Cm23Admin objects. The object holds all information about a single admin and loads values from the DB if the admin exists. If only name is given, an admin is loaded from the database, if name and password are given, an admin will be created, if nothing is given, the current admin is loaded and added to database if not already in.
Parameter:

- name: login name of an existing admin (to load or to create).
- password: password of a new admin (to create).

5.27.2 Cm23Admin::__destruct

Description: Destructor for a Cm23Admin object. Prints out messages and saves changed options to the database.

5.27.3 Cm23Admin::delete

Description: Deletes an m23 administrator from the database and from the password hash files for m23, backup, backuppc and phpmyldap.
Returns:

- : True if admin could be deleted, False if admin doesn't exist or cannot be deleted (one must remain, can't delete yourself, or errors with DB or files).

5.27.4 Cm23Admin::changePw

Description: Changes the password of an m23 administrator in the password hash files for m23, backup, backuppc and phpmyldap.
Parameter:

- $oldpassword: Password entered by the user, must be correct to be able to change password
- $newpassword: New password chosen by the user

Returns:

- : True if password was changed successfully, false otherwise

5.27.5 Cm23Admin::setCSS

Description: sets the CSS file
Parameter:

- $css: element indicating chosen CSS, see array $cssList

5.27.6 Cm23Admin::getCSS

Description: finds the CSS file for the CSS the user has chosen, defines selection for themeChoice
Returns:

- : name of CSS file

5.27.7 Cm23Admin::setLanguage

Description: sets the language for the admin
Parameter:

- $shortLanguage: indicates chosen language (current valid values: de, en or fr, see I18N_m23instLanguage($shortLangu

5.27.8 Cm23Admin::getLanguage

Description: finds the language the user has chosen
Returns:

- : name of language

5.27.9 Cm23Admin::addToDB

Description: Adds an m23 administrator to the DB
Parameter:

- $name: User name entered by the user
- $password: Password entered by the user

Returns:

- : True if database action was successful.

5.27.10 Cm23Admin::addToPwFiles

Description: Adds an m23 administrator to the password files
Parameter:

- $name: User name entered by the user
- $password: Password entered by the user

Returns:

- : True if file write action for m23 password file was successful.

5.27.11 Cm23Admin::delFromPwFiles

Description: Adds an m23 administrator to the password files
Parameter:

- $name: User name entered by the user

Returns:

- : True if file write action for m23 password file was successful.

5.28 ./inc/CMessageManager.php

Author: Hauke Goos-Habermann (HHabermann@pc-kiel.de)
Description: Class for managing messages in the m23 webinterface.

5.28.1 CMessageManager::addMessage

Description: Generic function for adding a message to a message stack.
Parameter:

- msg: Error message to add.
- msgVar: The reference to the variable where the massages should be stored.
- foundVar: The reference to the variable that should be set to true when there are messages in the given stack.

5.28.2 CMessageManager::addInfoMessage

Description: Adds an info message to the info message stack and sets the info flag.
Parameter:

- msg: Info message to add.

5.28.3 CMessageManager::addWarningMessage

Description: Adds a warning message to the warning message stack and sets the warning flag.
Parameter:

- msg: Warning message to add.

5.28.4 CMessageManager::addErrorMessage

Description: Adds an error message to the error message stack and sets the error flag.
Parameter:

- msg: Error message to add.

5.28.5 CMessageManager::popInfoMessagesHTML

Description: Returns all info messages and deletes the info message stack.
Returns:

- msg: The complete info message stack.

5.28.6 CMessageManager::popWarningMessagesHTML

Description: Returns all warning messages and deletes the warning message stack.
Returns:

- msg: The complete warning message stack.

5.28.7 CMessageManager::popErrorMessagesHTML

Description: Returns all error messages and deletes the error message stack.
Returns:

- msg: The complete error message stack.

5.28.8 CMessageManager::deleteAllMessages

Description: Deletes all messages from the message stack.

5.28.9 CMessageManager::hasInfos

Description: Returns if there are infos.
Returns:

- true, if there have infos been occurred otherwise false.

5.28.10 CMessageManager::hasWarnings

Description: Returns if there are warnings.
Returns:

- true, if there have warnings been occurred otherwise false.

5.28.11 CMessageManager::hasErrors

Description: Returns if there are errors.
Returns:

- true, if there have errors been occurred otherwise false.

5.28.12 CMessageManager::showInfo

Description: Shows existing info messages in an info box if there are any.
Returns:

- true, if there have info been occurred otherwise false.

5.28.13 CMessageManager::showWarning

Description: Shows (hopefully not) existing warning messages in a warning box if there are any.
Returns:

- true, if there have warnings been occurred otherwise false.

5.28.14 CMessageManager::showError

Description: Shows (hopefully not) existing error messages in an error box if there are any.
Returns:

- true, if there have errors been occurred otherwise false.

5.28.15 CMessageManager::showMessages

Description: Shows all existing messages in the according boxes.
Returns:

- true, if there have errors or warnings been occurred otherwise false.

5.29 ./inc/CObjectStorageManager.php

Author: Hauke Goos-Habermann (HHabermann@pc-kiel.de)
Description: Class for storing PHP objects in the database.

5.29.1 CObjectStorageManager::saveObject

Description: Saves or updates an object in the DB.
Parameter:

- ident: Identifier for objects belonging to e.g. a client.
- object: The object to store/update for the Identifier.

5.29.2 CObjectStorageManager::getAllObjectsByRes

Description: Fetches all objects from the DB that can be read via the given MySQL ressource ID.
Parameter:

- res: MySQL ressource ID.

Returns:

- Array with all matching objects.

5.29.3 CObjectStorageManager::getByIdent

Description: Fetches all objects from the DB that match an identifier.
Parameter:

- ident: Identifier for objects belonging to e.g. a client.

Returns:

- Array with all matching objects.

5.29.4 CObjectStorageManager::getByClass

Description: Fetches all objects from the DB that match a class.
Parameter:

- class: Class name of the objects to find.

Returns:

- Array with all matching objects.

5.29.5 CObjectStorageManager::getByIdentClass

Description: Fetches an object from the DB that match an identifier and a class.
Parameter:

- ident: Identifier for objects belonging to e.g. a client.
- class: Class name of the objects to find.

Returns:

- Found object or null, if no matching object could be found.

5.29.6 CObjectStorageManager::deleteObject

Description: Deletes an object from the DB that match an identifier and a class (given by the object itself).
Parameter:

- ident: Identifier for objects belonging to e.g. a client.
- object: The object to delete.

Returns:

- true, if it could be deleted. Otherwise false.

5.29.7 CObjectStorageManager::getObjectMethodReturnValue

Description: Checks, if an objects implements a method, calls it and returns the return value. If the method is unimplemented, an error value will be returned.
Parameter:

- object: Object to use.
- fkt: Name of the function (method)
- error: Error value to return, if the method is unimplemented.

Returns:

- Return value of the implemented method or given error value if unimplemented.

5.29.8 CObjectStorageManager::getCOSStatus

Description: Gets the status code of the object.
Parameter:

- object: Object to use or null, if $this->object should be used.

Returns:

- Status code.

5.29.9 CObjectStorageManager::getCOSStatusHumanReadable

Description: Gets the human readable status of the object.
Parameter:

- object: Object to use or null, if $this->object should be used.

Returns:

- Human readable status.

5.29.10 CObjectStorageManager::getCOSI18NVariable

Description: Gets the contents of an I18N variable for the object (if it exists).
Parameter:

- object: Object to use or null, if $this->object should be used.
- varPostfix: String to add to the name of the I18N variable.
- errMsg: Die message to display if the I18N variable is not set.

Returns:

- Contents of the the I18N variable.

5.29.11 CObjectStorageManager::getCOSDescription

Description: Gets the description for a given object class.
Parameter:

- object: Object to use or null, if $this->object should be used.

Returns:

- Description of the given object class.

5.29.12 CObjectStorageManager::getCOSName

Description: Gets the (human readable) name for a given object class.
Parameter:

- object: Object to use or null, if $this->object should be used.

Returns:

- (human readable) name of the given object class.

5.29.13 CObjectStorageManager::showList

Description: Shows a list of existing objects matching criteria set by getBy* functions.

5.30 ./inc/CObjectStorage.php

Author: Hauke Goos-Habermann (HHabermann@pc-kiel.de)
Description: Class for storing PHP objects in the database.

5.30.1 CObjectStorage::statusCodeToHumanReadable

Description: Translates a status code (COSSTATUS_*) into human readable word(s).
Parameter:

- code: The status code (COSSTATUS_*).

Returns:

- Human readable word(s) representing the status code.

5.30.2 CObjectStorage::__construct

Description: Constructor for new CObjectStorage objects. The object saves/updates or loads an object.
Parameter:

- ident: Identifier for objects belonging to e.g. a client.
- classOrObject: Class name of the object or the object itself.
- readFromDBIfObjectExists: Set to true, if an existing object should read from the DB, even if an object is given.

5.30.3 CObjectStorage::__destruct

Description: Destructor for storing changed values of the object back to the DB.

5.30.4 CObjectStorage::getObject

Description: Gets the object.
Returns:

- object.

5.30.5 CObjectStorage::destroy

Description: Destroys the
Returns:

- Human readable status.

5.31 ./inc/CPackageStatusCompare.php

Author: Hauke Goos-Habermann (HHabermann@pc-kiel.de)
Description: Class for comparing the package status between clients and files.

5.31.1 CPackageStatusCompare::__construct

Description: Constructor for new CPackageStatusCompare objects.

5.31.2 CPackageStatusCompare::__destruct

Description: Destructor for a CPackageStatusCompare object.

5.31.3 CPackageStatusCompare::save

Description: Saves parameters as hidden POST values.

5.31.4 CPackageStatusCompare::setFile

Description: Sets a file name, if the given parameter is a valid file name.
Parameter:

- var: Variable that stores a client or file name.
- file: Name of the status file.
- origVar: Variable that stores the original file name.
- origFile: Original name of the status file.

5.31.5 CPackageStatusCompare::setFile1

Description: Sets a 1st file name, if the given parameter is a valid file name.
Parameter:

- origFile: Original file name.
- file: Name of the status file.

5.31.6 CPackageStatusCompare::setFile2

Description: Sets a 2nd file name, if the given parameter is a valid file name.
Parameter:

- origFile: Original file name.
- file: Name of the status file.

5.31.7 CPackageStatusCompare::setClient

Description: Sets a client, if the given parameter is a valid client name.
Parameter:

- var: Variable that stores a client or file name.
- cl: Name of the client

5.31.8 CPackageStatusCompare::setClient1

Description: Sets a client, if the given parameter is a valid client name.
Parameter:

- cl: Name of the client

5.31.9 CPackageStatusCompare::setClient2

Description: Sets a client, if the given parameter is a valid client name.
Parameter:

- cl: Name of the client

5.31.10 CPackageStatusCompare::isClientSet

Description: Checks, if the client is set.
Parameter:

- var: Variable that stores a client or file name

Returns:

- true, if client name is set.

5.31.11 CPackageStatusCompare::isClient1Set

Description: Checks, if the 1st client is set.
Returns:

- true, if client name is set.

5.31.12 CPackageStatusCompare::isClient2Set

Description: Checks, if the 2nd client is set.
Returns:

- true, if client name is set.

5.31.13 CPackageStatusCompare::isFile1Set

Description: Checks, if the 1st combined variable stores a file name.
Returns:

- true, if the variable stores a file name, otherwise false.

5.31.14 CPackageStatusCompare::isFile2Set

Description: Checks, if the 2st combined variable stores a file name.
Returns:

- true, if the variable stores a file name, otherwise false.

5.31.15 CPackageStatusCompare::getVersionStatus

Description: Gets an array with, the package file name, the version and the status.
Parameter:

- file: File that stores the package information.
- package: The package to get the information for.

Returns:

- Array with, the package file name, the version and the status.

5.31.16 CPackageStatusCompare::getStatusFile

Description: Gets the file name of a package status file. If a client name is give, a temporary package status file will be written.
Parameter:

- clientOrFile: Client or package status file name.
- isFile: Set to true, if $clientOrFile is a file, otherwise false.

Returns:

- Array with, the package file name, the version and the status.

5.31.17 CPackageStatusCompare::showStatusRow

Description: Gets the file name of a package status file. If a client name is give, a temporary package status file will be written.
Parameter:

- left: Left package name.
- leftVersion: Left version of the package.
- leftStatus: Left status of the package.
- type: diff type.
- right: Right package name.
- rightVersion: Right version of the package.
- rightStatus: Right status of the package.

Returns:

- Array with, the package file name, the version and the status.

5.31.18 CPackageStatusCompare::diff

Description: Compares two package status files.

5.31.19 CPackageStatusCompare::show

Description: Shows the comparing dialog.

5.32 ./inc/CPoolFromClientDebsGUI.php

Author: Hauke Goos-Habermann (HHabermann@pc-kiel.de)
Description: GUI class for building pools from client packages.

5.32.1 CPoolFromClientDebsGUI::__construct

Description: Constructor for new CPoolFromClientDebsGUI objects. The object creates a new pool that stores all packages that are needed to install the client.
Parameter:

- clientName: Name of the client to create the pool from.

5.32.2 CPoolFromClientDebsGUI::addm23BuildPoolFromClientDebsJob

Description: Adds a m23BuildPoolFromClientDebs job to the client.
Parameter:

- clientName: Name of the client, the pool should be build from.

5.32.3 CPoolFromClientDebsGUI::DEFINE_storableCheckboxForAddingm23BuildPoolFromC

Description: Defines a checkbox, that adds a m23BuildPoolFromClientDebs job when checked.
Parameter:

- htmlName: Name of the HTML element and constant.
- clientName: Name of the client, the pool should be build from.
- prefKey: Variable name of the preference the dialog element stands for.
- storePointer: Additional pointer to the variable where to store the entered value.

5.33 ./inc/CPoolFromClientGUI.php

Author: Hauke Goos-Habermann (HHabermann@pc-kiel.de)
Description: GUI class for building pools from client packages.

5.33.1 CPoolFromClientGUI::DEFINE_checkboxForAddingm23BuildPoolFromClientPackage

Description: Shows a checkbox, that adds a m23BuildPoolFromClient job when checked.
Parameter:

- htmlName: Name of the HTML element and constant.
- clientName: Name of the client, the pool should be build from.

5.33.2 CPoolFromClientGUI::setImportPoolName

Description: Sets the name of this pool by the client name.
Parameter:

- clientName: Name of the client, the pool is build from.

5.33.3 CPoolFromClientGUI::getImportPoolName

Description: Returns the name of this pool.
Returns:

- The name of this pool.

5.33.4 CPoolFromClientGUI::__construct

Description: Constructor for new CPoolFromClientGUI objects. The object creates a new pool that stores all packages that are needed to install the client.
Parameter:

- clientName: Name of the client to create the pool from.

5.33.5 CPoolFromClientGUI::getCOSStatusHumanReadable

Description: Translates the status code (COSSTATUS_*) into human readable word(s).
Returns:

- Human readable word(s) representing the status code.

5.33.6 CPoolFromClientGUI::getCOSStatus

Description: Gets the status code (COSSTATUS_*).
Returns:

- Status code of this object.

5.33.7 CPoolFromClientGUI::runCOSLoop

Description: Function that is called on every run of CObjectStorageManager::getAllObjectsByRes.

5.33.8 CPoolFromClientGUI::saveInObjectStorage

Description: Saves this object in the object storage.

5.34 ./inc/CPoolGUI.php

Author: Hauke Goos-Habermann (HHabermann@pc-kiel.de)
Description: GUI class for building pools.

5.34.1 CPoolGUI::show

Description: Shows the start dialog for creating, adding, deleting and changing pools.

5.34.2 CPoolGUI::getHelpPage

Description: Returns the current help page.
Returns:

- Current help page.

5.34.3 CPoolGUI::getHeading

Description: Returns the current heading.
Returns:

- Current heading.

5.34.4 POOL_showReadCD

Description: shows a dialog for copying the CD contents to the pool
Parameter:

- poolName: name of the pool

5.34.5 CPoolGUI::DEFINE_convertPackagesToRepositoryStatus

Description: Defines dialog elements for the status of the Packages* and sources.list generation of the currently generated pool.
Parameter:

- BUT_refresh: HTML constant name for the refresh button.
- BUT_step0: HTML constant name for the go back to start button.
- LA_convertPackagesToRepositoryStatus: HTML constant name for showing the conversation status.

5.34.6 CPoolGUI::DIALOG_convertPackagesToRepositoryStatus

Description: Shows information (status of the Packages* generation, sources.list) about the currently generated pool.

5.34.7 CPoolGUI::DEFINE_showDownloadStatus

Description: Defines HTML elements for the package download status of a pool.
Parameter:

- BUT_step2: HTML constant name for the go to the pool generation dialog button.
- BUT_refresh: HTML constant name for the refresh button.
- LA_downloadStatus: HTML constant name for showing the download status.
- SPAN_poolSize: HTML constant name for showing the size of the pool.

5.34.8 CPoolGUI::DIALOG_showDownloadStatus

Description: Shows the package download status of a pool.

5.34.9 CPoolGUI::DIALOG_start

Description: Shows the start dialog for creating, adding, deleting and changing pools.

5.34.10 CPoolGUI::DEFINE_nextStepCopyDownloadPackages

Description: Defines a button for going to the next step (copy or download of packages).
Parameter:

- BUT_copyDownloadPackages: HTML constant name for the copy or download of packages button.

5.34.11 CPoolGUI::DEFINE_updatePackageIndexAndReSignPool

Description: Defines a button for re-creating the Packages files and to re-sign the Release file.
Parameter:

- BUT_reSignPool: HTML constant name for the re-sign button.

5.34.12 CPoolGUI::DEFINE_changePoolDescription

Description: Defines dialog elements for changing the pool description.
Parameter:

- TA_poolDescription: HTML constant name for the pool description text box.
- BUT_poolSaveChanges: HTML constant name for the pool saving button.
- LA_poolSourcesList: HTML constant name for showing sourceslist of the pool.

5.34.13 CPoolGUI::DEFINE_loadDeletePool

Description: Defines dialog elements for loading or deleting a pool.
Parameter:

- SEL_loadDeletePoolname: HTML constant name for the pool name.
- BUT_loadPool: HTML constant name for the pool loading button.
- BUT_deletePool: HTML constant name for the pool deletion button.

5.34.14 CPoolGUI::DEFINE_createBasicPool

Description: Defines dialog elements for creating a basic pool.
Parameter:

- ED_createPoolname: HTML constant name for the pool name input field.
- RB_createPooltype: HTML constant name for the pool type selection radio buttons.
- RB_createPoolarch: HTML constant name for the pool architecture selection radio buttons.
- BUT_createPool: HTML constant name for the pool creation button.

5.35 ./inc/CPoolLister.php

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

5.36 ./inc/CPool.php

Author: Hauke Goos-Habermann (HHabermann@pc-kiel.de)
Description: GUI class for building pools.

5.36.1 CPool::__construct

Description: Constructor for new CPool objects that loads existing pools or creates a new basic pool.
Parameter:

- poolName: Name of the pool (if a pool with the given name exists => load)
- poolType: Type of the pool (POOL_TYPE_CD or CPoolLister::POOL_TYPE_DOWNLOAD).
- poolArch: Architecture of the pool (POOL_ARCH_I386 or CPoolLister::POOL_ARCH_AMD64).

5.36.2 CPool::signRelease

Description: Signs the Release file as Release.gpg and InRelease.
Returns:

- : Name the log file.

5.36.3 CPool::getConvertPackagesToRepositoryLogName

Description: Returns the full file name of the convert packages to repository log file.
Returns:

- : Name the log file.

5.36.4 CPool::getConvertPackagesToRepositoryLogNewLines

Description: Gets the last (new) lines of the (growing) convert packages to repository log file.
Returns:

- : UTF8-encoded new lines of the log file.

5.36.5 CPool::isConvertPackagesToRepositoryRunning

Description: Checks if the conversation of downloaded packages to a repository is running.
Returns:

- : true, if it is running, otherwise false.

5.36.6 CPool::convertPackagesToRepository

Description: Generates a package source from packages stored in one directory.
Parameter:

- returnCommands: If set to true, the commands for downloading the packages will be returned instead of executed.
- runInScreen: Set to true if the execution should be done in "screen". False executes it under the normal BASH.

Returns:

- Commands for creating the package source when $returnCommands is true, true when the screen session for creating the package source is started and false on errors.

5.36.7 CPool::isPackageTaskRunning

Description: Checks if a packages task (download to the pool or creating the pool from downloaded files) is running.
Parameter:

- singleScreenName: Name of the single screen, that is used when downloading and creation of the pool are two seperate screen sessions.
-

Returns:

- : true, if the task is running, otherwise false.

5.36.8 CPool::isDownloadRunning

Description: Checks if a download of packages to the pool is running.
Returns:

- : true, if download is running, otherwise false.

5.36.9 CPool::resetDownloadLog

Description: Deletes the aptDownload.log file and resets the line number of the last read line.

5.36.10 CPool::getDownloadLogNewLines

Description: Gets the last (new) lines of the (growing) download log file.
Returns:

- : UTF8-encoded new lines of the log file.

5.36.11 CPool::getDownloadLogContents

Description: Gets the contents of the download log file.
Returns:

- : Contents of the download log file.

5.36.12 CPool::getPoolImportedFromSourceslist

Description: Gets the complete sourceslist that was used to download the packages into the pool.
Returns:

- : Sourceslist that was used to download the packages into the pool.

5.36.13 CPool::setPoolImportedFromSourceslist

Description: Sets the complete sourceslist that was used to download the packages into the pool.
Parameter:

- importedFromImportedFromSourceslist: Complete sourceslist that was used to download the packages into the pool.

5.36.14 CPool::hasPoolDownloadBasePackages

Description: Checks, if base packages should be downloaded.
Returns:

- : true, if base packages should be downloaded otherwise false.

5.36.15 CPool::setPoolDownloadBasePackages

Description: Sets, if base packages should be downloaded.
Parameter:

- downloadBasePackages: true, when base packages should be downloaded otherwise false.

5.36.16 CPool::getPoolImportedPackageList

Description: Gets the list of packages that were downloaded (or have to be downloaded) into the pool.
Returns:

- : PackageList of the pool.

5.36.17 CPool::setPoolImportedPackageList

Description: Sets the list of packages that were downloaded (or have to be downloaded) into the pool.
Parameter:

- packageList: PackageList of the pool.

5.36.18 CPool::getPoolDistribution

Description: Gets the distribution value of the pool.
Returns:

- : Distribution of the pool.

5.36.19 CPool::setPoolDistribution

Description: Sets the distribution value of the pool.
Parameter:

- distribution: Distribution of the pool.

5.36.20 CPool::createBasicPool

Description: Sets the name, type and architecture of the pool and creates the pool directory.
Parameter:

- poolName: Name of the pool.
- poolType: Type of the pool (POOL_TYPE_CD or CPoolLister::POOL_TYPE_DOWNLOAD).
- poolArch: Architecture of the pool (POOL_ARCH_I386 or CPoolLister::POOL_ARCH_AMD64).

5.36.21 CPool::getPoolDir

Description: Gets the directory of the pool.
Returns:

- : Directory of the pool.

5.36.22 CPool::setPoolName

Description: Sets the name of the pool and create the pool directory.
Parameter:

- poolName: Name of the pool.

5.36.23 CPool::getPoolName

Description: Gets the name of the pool.
Parameter:

- returnEmptyIfNull: Returns an empty string, if $this->poolName is null.

Returns:

- : Name of the pool.

5.36.24 CPool::getPoolSourceslist

Description: Gets the sourceslist value of the pool.
Returns:

- : Sourceslist of the pool.

5.36.25 CPool::setPoolSourceslist

Description: Sets the sourceslist value of the pool.
Parameter:

- sourceslist: Sourceslist of the pool.

5.36.26 CPool::getPoolDescription

Description: Gets the description value of the pool.
Returns:

- : Description of the pool.

5.36.27 CPool::setPoolDescription

Description: Sets the description value of the pool.
Parameter:

- description: Description of the pool.

5.36.28 CPool::getPoolRelease

Description: Gets the release value of the pool.
Returns:

- : Release of the pool.

5.36.29 CPool::setPoolRelease

Description: Sets the release value of the pool.
Parameter:

- release: Release of the pool.

5.36.30 CPool::getPoolType

Description: Gets the type value of the pool.
Returns:

- : Type of the pool.

5.36.31 CPool::setPoolType

Description: Sets the type value of the pool.
Parameter:

- type: Type of the pool (POOL_TYPE_CD or CPoolLister::POOL_TYPE_DOWNLOAD or CPoolLister::POOL_TYPE_USECLIENTDEBS).

5.36.32 CPool::getPoolArch

Description: Gets the architecture value of the pool.
Returns:

- : Architecture of the pool.

5.36.33 CPool::setPoolArch

Description: Sets the architecture value of the pool.
Parameter:

- arch: Architecture of the pool (POOL_ARCH_I386 or CPoolLister::POOL_ARCH_AMD64).

5.36.34 CPool::setProperty

Description: Writes the contents of a property file.
Parameter:

- property: name of the pool property
- value: value to write in the pool property file

5.36.35 CPool::getProperty

Description: Reads the contents of a property file.
Parameter:

- property: Name of the pool property

Returns:

- Contents of a property file

5.36.36 CPool::getPoolSize

Description: Calculates the disk usage of a pool.
Returns:

- Size of the pool in MB

5.36.37 CPool::destroyPool

Description: Deletes the pool
Parameter:

- poolName: Name of the pool (can optionally be set here)

5.36.38 CPool::checkForDistributionSpecificPackageFunction

Description: Checks for a distribution specific package function in the distribution's packages.php file and generates an error message in case it cannot be found.
Parameter:

- fkt: Name of the function to check for.
- errorMsg: Error message to add, if the function cannot be found.

5.36.39 CPool::preparePool

Description: Prepares the pool to make it able to store software packages.
Returns:

- false on errors, true on success.

5.36.40 CPool::stopDownloadToPool

Description: Stops the download of packages to the pool.

5.36.41 CPool::downloadDebsFromClient

Description: Checks, if all pre-requirements for downloading packages to the pool are satisfied. Then starts the routine to download the packages directly from the client.
Parameter:

- clientName: Name of the client to download the packages from.
- returnCommands: If set to true, the commands for downloading the packages will be returned instead of executed.

Returns:

- true, if the download was started otherwise false.

5.36.42 CPool::startDownloadToPool

Description: Checks, if all pre-requirements for downloading packages to the pool are satisfied. Then starts the distribution specific download routine.
Parameter:

- returnCommands: If set to true, the commands for downloading the packages will be returned instead of executed.

Returns:

- true, if the download was started otherwise false.

5.36.43 CPool::getPoolStatusFileName

Description: Returns the name of the pool status file.
Returns:

- Name of the pool status file.

5.36.44 CPool::addPoolStatusFileCommand

Description: Generates BASH code that sets a status in the pool status file.
Parameter:

- status: Status to set in the file.

Returns:

- BASH code that sets a status in the pool status file.

5.36.45 CPool::getPoolStatusFileContents

Description: Gets the contents of the pool status file.
Returns:

- Contents of the pool status file.

5.36.46 CPool::downloadPackagesAndCreatePool

Description: Downloads packages and creates a pool from them in one combined screen session.
Parameter:

- clientName: Name of the client (only needed when downloadDebsFromClient is used)

5.37 ./inc/cron.php

Author: Hauke Goos-Habermann (HHabermann@pc-kiel.de)
Description: Functions for managing cron jobs.

5.37.1 CRON_genCronEntry

Description: Creates a cron line to insert into crontab.
Parameter:

- min: Minute or * for every minute to execute the cron job. (0-59)
- hour: Hour or * for every hour to execute the cron job. (0-23)
- dayOfMonth: The day in the month or * for all days to execute the cron job. (1-31)
- month: The month or * for every months to execute the cron job. (1-12)
- dayOfWeek: The day in the week or * for every week days to execute the cron job. (0-7 (0 or 7 is Sun))
- user: The user the cron job should be run under.
- cmd: The command to exectute.
- identifier: A string to identify the cron entry (for deletion)
- runInScreen: Set to true if the command should be run in a screen.

Returns:

- The crontab line that can be inserted.

5.37.2 CRON_addJobMinutely

Description: Runs a command every N minutes.
Parameter:

- intervall: Amount of minutes to wait between calls.
- user: The user the cron job should be run under.
- cmd: The command to exectute.
- identifier: A string to identify the cron entry (for deletion)
- runInScreen: Set to true if the command should be run in a screen.

5.37.3 CRON_addJobBasic

Description: Adds a command to the crontab.
Parameter:

- min: Minute or * for every minute to execute the cron job. (0-59)
- hour: Hour or * for every hour to execute the cron job. (0-23)
- dayOfMonth: The day in the month or * for all days to execute the cron job. (1-31)
- month: The month or * for every months to execute the cron job. (1-12)
- dayOfWeek: The day in the week or * for every week days to execute the cron job. (0-7 (0 or 7 is Sun))

- user: The user the cron job should be run under.
- cmd: The command to exectute.
- identifier: A string to identify the cron entry (for deletion)
- runInScreen: Set to true if the command should be run in a screen.

5.37.4 CRON_rmJob

Description: Removes an entry from the crontab.
Parameter:

- identifier: A string to identify the cron entry (for deletion)

5.37.5 CRON_reloadConfig

Description: Reloads cron with new crontab.

5.37.6 CRON_addJobHourly

Description: Runs a command every N hours.
Parameter:

- min: Minute or * for every minute to execute the cron job. (0-59)
- user: The user the cron job should be run under.
- cmd: The command to exectute.
- identifier: A string to identify the cron entry (for deletion)
- runInScreen: Set to true if the command should be run in a screen.

5.37.7 CRON_addJobDayly

Description: Runs a command every day at a specified time.
Parameter:

- min: Minute or * for every minute to execute the cron job. (0-59)
- hour: Hour or * for every hour to execute the cron job. (0-23)
- user: The user the cron job should be run under.
- cmd: The command to exectute.
- identifier: A string to identify the cron entry (for deletion)
- runInScreen: Set to true if the command should be run in a screen.

5.37.8 CRON_addJobWeekly

Description: Runs a command every week on a specified week day at a specified time.
Parameter:

- min: Minute or * for every minute to execute the cron job. (0-59)
- hour: Hour or * for every hour to execute the cron job. (0-23)
- dayOfWeek: The day in the week or * for every week days to execute the cron job. (0-7 (0 or 7 is Sun))

- user: The user the cron job should be run under.
- cmd: The command to exectute.
- identifier: A string to identify the cron entry (for deletion)
- runInScreen: Set to true if the command should be run in a screen.

5.37.9 CRON_getTimeBaseArray

Description: Returns an associative array that contains the timebases for cron that are supported by m23.
Returns:

- Associative array with the timebases for cron that are supported by m23 with one-character name as key and language specific human readable string as value.

5.37.10 CRON_getDayOfWeekArray

Description: Returns an associative array that contains the week day names for cron.
Returns:

- Associative array with the week day names for cron with three letter cron name as key and language specific human readable week day as value.

5.37.11 CRON_checkMinute

Description: Checks if a minute value is valid.
Parameter:

- min: Minute value to check.

Returns:

- True if it is valid otherwise false.

5.37.12 CRON_checkHour

Description: Checks if a hour value is valid.
Parameter:

- hour: Hour value to check.

Returns:

- True if it is valid otherwise false.

5.37.13 CRON_isEntryPresent

Description: Checks, if a crontab entry is present with a given identifier.
Parameter:

- identifier: A string to identify the cron entries

Returns:

- true, if a crontab entry is present with a given identifier, otherwise false.

5.37.14 CRON_getEntriesByIdentifier

Description: Parses the crontab for all lines matching the identifier.
Parameter:

- identifier: A string to identify the cron entries

Returns:

- Associative array with the crontab lines that match the identifier and named by the parameters.

5.37.15 CRON_getNextIdentifierNr

Description: Calculates the next higher identifier number from a given identifier.
Parameter:

- identifier: A string to identify the cron entries.

Returns:

- Identifier with next higher number attaced.

5.37.16 CRON_cronManagementDialog

Description: Shows a dialog for viewing, adding and deleting crontab entries for a given user, command and identifier.
Parameter:

- user: The user the cron job should be run under.
- cmd: The command to exectute.
- identifier: A string to identify the cron entry (for deletion)
- runInScreen: Set to true if the command should be run in a screen.

5.37.17 CRON_cronEntryDeletionDialog

Description: Shows a a list of crontab entries matching the identifier with deletion option.
Parameter:

- identifier: A string to identify the cron entry (for deletion)

5.37.18 CRON_translateEveryIntervallValue

Description: Translates a time value with possible intervall into a human readable string.
Parameter:

- val: Time value (e.g. 2/2)

Returns:

- Human readable string for a given time value.

5.38 ./inc/CSchoolManagement.php

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

5.38.1 CSchoolManagement::__construct

Description: Constructor for new CSchoolManagement objects.
Parameter:

- schoolName: Name of the school.

5.38.2 CSchoolManagement::GUI_showSchoolChooser

Description: Shows a dialog for selecting the school.
Parameter:

- generic: If set to true show a generic school chooser, otherwise show the school management edition.
- hideIfButtonPressed: No dialog will be shown, if set to true and the save button is clicked.

Returns:

- Short name of the selected school, if button is pressed, otherwise empty string.

5.38.3 CSchoolManagement::GUI_showSchoolUserCreator

Description: Shows a dialog for creating a user (admin, teacher, pupil) in the school.

5.38.4 CSchoolManagement::GUI_showCreateMegaAdmin

Description: Shows a dialog for creating a admin.

5.38.5 CSchoolManagement::GUI_showDeleteMegaAdmin

Description: Shows a dialog for deleting a admin.

5.39 ./inc/CSchool-old.php

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

5.39.1 CSchool::__construct

Description: Constructor for new CSchool objects.
Parameter:

- schoolName: Name of the school.

5.39.2 CSchool::createm23Subgroups

Description: Create client groups with the name of the school and different postfixes.
Parameter:

- schoolName: Name of the school.
- description: Base description of the school.

5.39.3 CSchool::getm23SubgroupsPostfixes

Description: Creates an array with the postfixes of the m23 subgroups.
Parameter:

- Array with the postfixes of the m23 subgroups as key and their description as value.

5.39.4 CSchool::checkNewgroupBeforeClientAdd

Description: Checks, if the needed groups are selected in the needed amount
Parameter:

- schoolName: Short name of the school.
- newgroup: Array with selected groups.

Returns:

- Error message(s) in case of errors or empty string when all is ok.

5.39.5 CSchool::getSchoolGrades

Description: Get the grades of a school.
Parameter:

- schoolName: Short name of the school.

Returns:

- Array with the school grades as key and value.

5.39.6 CSchool::getUserSchoolRoles

Description: Gets an array with the school roles.
Parameter:

- : addAdmin: If set to true, the admin role will be included in the array.

Returns:

- Array with the school roles: Nummeric value as key, description as value.

5.39.7 CSchool::isSchoolModeActive

Description: Checks, if the school mode is active.
Returns:

- true, if the school mode is active, otherwise false.

5.39.8 CSchool::getSchoolNameWithDescription

Description: Get an array with the short school names as key and the short school names with description as value.
Returns:

- Associative array with short school names and their descriptions.

5.39.9 CSchool::getSchoolNames

Description: Get the names of all known schools.
Returns:

- Associative array with the names of all known schools as key and value.

5.39.10 CSchool::getLongSchoolName

Description: Get the long name of a school.
Parameter:

- schoolName: Short name of the school.

Returns:

- Long name of the given school.

5.39.11 CSchool::readSchoolInfoFile

Description: Reads and parses the school info file for a given school.
Parameter:

- schoolName: Name of the school or NULL, if all schools should be given out.

Returns:

- Information about the given school or all schools

5.39.12 CSchool::isSchoolExisting

Description: Checks, if a school is existing in LDAP.
Parameter:

- schoolName: Name of the school.

5.39.13 CSchool::setSchool

Description: Sets the name of the school (must be a DC in the LDAP) under the base.
Parameter:

- schoolName: Name of the school.

5.39.14 CSchool::addServer

Description: Adds a client to LDAP as FusionDirectory server.
Parameter:

- clientName: Name of the client
- ip: IP address of the client
- mac: MAC address of the client
- serial: Serial number
- description: Description for the client
- location: Location of the client
- pwd: the unencrypted password

Returns:

- true, if the client could be added, otherwise false.

5.39.15 CSchool::addWorkstation

Description: Adds a client to LDAP as FusionDirectory workstation.
Parameter:

- clientName: Name of the client
- ip: IP address of the client
- mac: MAC address of the client
- serial: Serial number
- description: Description for the client
- location: Location of the client
- pwd: the unencrypted password

Returns:

- true, if the client could be added, otherwise false.

5.39.16 CSchool::addComputer

Description: Adds a computer to LDAP as FusionDirectory object.
Parameter:

- fdConfigVar: Name of the FusionDirectory config variable holding the RDN.
- fdObjectClass: FusionDirectory object class of the new element.
- computerName: Name of the client
- ip: IP address of the client
- mac: MAC address of the client
- serial: Serial number
- desciption: Description for the client
- location: Location of the client

- pwd: the (un)encrypted password
- cryptPw: set to true, if the password should be encrypted or false, if it's already encrypted

Returns:

- true, if the client could be added, otherwise false.

5.39.17 CSchool::deleteComputer

Description: Deletes a computer object from LDAP.
Parameter:

- fdConfigVar: Name of the FusionDirectory config variable holding the RDN.
- fdObjectClass: FusionDirectory object class of the new element.
- computerName: Name of the client

Returns:

- true, if the client could be added, otherwise false.

5.39.18 CSchool::deleteServer

Description: Deletes a client/server from LDAP.
Parameter:

- clientName: Name of the client

Returns:

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

5.39.19 CSchool::deleteWorkstation

Description: Deletes a client/workstation from LDAP.
Parameter:

- clientName: Name of the client

Returns:

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

5.39.20 CSchool::createUser

Description: Adds a posix account to the LDAP server and encrypts the password with MD5.
Parameter:

- 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
- cryptPw: set to true, if the password should be encrypted or false, if it's already encrypted

Returns:

- true or false in case of an error.

5.39.21 CSchool::deleteUser

Description: Removes a posix account from the LDAP server.
Parameter:

- account: the login name

Returns:

- true or false in case of an error.

5.39.22 CSchool::deleteGroup

Description: Deletes a posix group from the LDAP server.
Parameter:

- groupname: Name of the group.

Returns:

- true or error message string in case of an error.

5.39.23 CSchool::delKerberosPrincipalFromExistingLDAP

Description: Removess Kerberos principal attributes from an existing LDAP object.
Parameter:

- dn: DN where to delete the attributes from.
- account: the login name

Returns:

- Output of kadmin.local delprinc...

5.39.24 CSchool::addKerberosPrincipalToExistingLDAP

Description: Adds Kerberos principal attributes to an existing LDAP object.
Parameter:

- dn: DN where to add the attributes.
- account: the login name
- pwd: the unencrypted password

Returns:

- Output of kadmin.local addprinc...

5.39.25 CSchool::addKerberosWorkstation

Description: Adds a Kerberos entries to an existing FusionDirectory workstation object in LDAP.
Parameter:

- computerName: Name of the client
- pwd: the (un)encrypted password

Returns:

- true, if the Kerberos entries could be added, otherwise false.

5.39.26 CSchool::addKerberosServer

Description: Adds a Kerberos entries to an existing FusionDirectory server object in LDAP.
Parameter:

- computerName: Name of the client
- pwd: the (un)encrypted password

Returns:

- true, if the Kerberos entries could be added, otherwise false.

5.39.27 CSchool::addKerberosComputer

Description: Adds a Kerberos entries to an existing FusionDirectory computer object in LDAP.
Parameter:

- fdConfigVar: Name of the FusionDirectory config variable holding the RDN.
- computerName: Name of the client
- pwd: the (un)encrypted password

Returns:

- true, if the Kerberos entries could be added, otherwise false.

5.39.28 CSchool::getHostPrincipalName

Description: Gets the Kerberos host principal for a given computer.
Parameter:

- computerName: Name of the client

Returns:

- Kerberos host principal.

5.39.29 CSchool::getHostPrincipalKeytab

Description: Gets the Kerberos keytab for a given computer in base64 encoded format.
Parameter:

- computerName: Name of the client

Returns:

- Kerberos keytab.

5.39.30 CSchool::createGroup

Description: Creates a posix group on the LDAP server.
Parameter:

- groupname: Name of the group.
- gid: Linux group ID
- description: Description of the group.

Returns:

- true or error message string in case of an error.

5.39.31 CSchool::addAccountToGroup

Description: Adds a posix account to a posix group on the LDAP server.
Parameter:

- groupname: Name of the group.
- gid: Linux group ID

Returns:

- true or error message string in case of an error.

5.39.32 CSchool::delAccountFromGroup

Description: Removes a posix account from a posix group on the LDAP server.
Parameter:

- groupname: Name of the group.
- gid: Linux group ID

Returns:

- true or error message string in case of an error.

5.39.33 CSchool::createSchoolStructure

Description: Creates the LDAP structure for a new school.
Parameter:

- schoolName: Short name of the school.
- description: Full school name.

5.39.34 CSchool::moveUserToSchool

Description: Moves a posix account from the current school to another.
Parameter:

- account: the login name
- newSchool: Name of the new (existing) school.

Returns:

- true or error message string in case of an error.

5.39.35 CSchool::getNextFreeIP

Description: Returns an array with free IPs or a single free IP that can be used for the next client(s).
Parameter:

- amount: Amount of IDs to generate.
- rangeName: Name of the range (with min and max) defined in /m23/inc/schoolInfoConf.php.

Returns:

- Array with free IPs in the given amount or a single free IP.

5.39.36 CSchool::getRangeInfo

Description: Get the value of an option from a range in the school info.
Parameter:

- rangeName: Name of the range defined in /m23/inc/schoolInfoConf.php.
- option: Name of the option.

Returns:

- Value of the option (or dies if no value is present).

5.39.37 CSchool::getNetmask

Description: Get the netmask for m23 clients.
Parameter:

- rangeName: Name of the range defined in /m23/inc/schoolInfoConf.php.

Returns:

- Netmask for m23 clients.

5.39.38 CSchool::getGateway

Description: Get the gateway for m23 clients.
Parameter:

- rangeName: Name of the range defined in /m23/inc/schoolInfoConf.php.

Returns:

- Gateway for m23 clients.

5.39.39 CSchool::getDNS

Description: Get the DNS server for m23 clients.
Parameter:

- rangeName: Name of the range defined in /m23/inc/schoolInfoConf.php.

Returns:

- DNS server for m23 clients.

5.39.40 CSchool::addPrinter

Description: Adds a printer to LDAP in FusionDirectory style.
Parameter:

- printerName: Name of the printer
- ip: IP address of the printer
- mac: MAC address of the printer
- description: Description for the printer
- location: Location of the client
- labeledURI: The printer URI.

Returns:

- true, if the printer could be added, otherwise false.

5.39.41 CSchool::addNetDevices

Description: Adds a network device to LDAP in FusionDirectory style.
Parameter:

- deviceName: Name of the netdevice
- ip: IP address of the netdevice
- mac: MAC address of the netdevice
- description: Description for the netdevice

Returns:

- true, if the netdevice could be added, otherwise false.

5.39.42 CSchool::getUsedLDAPIPs

Description: Get an array with IP addresses that are used for computers and devices in the LDAP.
Parameter:

- intIPs: Set to true, if the IPs should be converted to integer values.

Returns:

- Array with IP addresses that are used for computers and devices in the LDAP

5.39.43 CSchool::IPexists

Description: Checks if an IP with the selected IP exists in LDAP for the current school
Parameter:

- ip: IP to check
- addErrorMessageIfIPExists: Set to true, if an error message should be added to the message stack, if the IP exists

Returns:

- true, if the IP exists (is used), otherwise false.

5.39.44 CSchool::createRootUser

Description: Adds a posix account that should act as root to the LDAP server and encrypts the password with MD5.
Parameter:

- 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
- cryptPw: set to true, if the password should be encrypted or false, if it's already encrypted

Returns:

- true if all is ok or false in case of an error.

5.39.45 CSchool::deleteRootUser

Description: Deletes a posix account that should act as root from the LDAP server
Returns:

- true if all is ok or false in case of an error.

5.39.46 CSchool::createMegaAdmin

Description: Creates a new account, that is administrator in FusionDirectory and m23 and that can login into m23 clients and become root via sudo.
Parameter:

- loginName: the login name
- password: the unencrypted password

Returns:

- true, if the mega administrator was created successfully, otherwise false.

5.39.47 CSchool::deleteMegaAdmin

Description: Deletes a mega administrator.
Parameter:

- loginName: the login name

Returns:

- true, if the mega administrator was deleted successfully, otherwise false.

5.39.48 CSchool::getMegaAdmins

Description: Generates an array with all mega administrators.
Returns:

- Associative array with all mega administrators as key and value.

5.40 ./inc/CSchool.php

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

5.40.1 CSchool::__construct

Description: Constructor for new CSchool objects.
Parameter:

- schoolName: Name of the school.

5.40.2 CSchool::createm23Subgroups

Description: Create client groups with the name of the school and different postfixes.
Parameter:

- schoolName: Name of the school.
- description: Base description of the school.

5.40.3 CSchool::getm23SubHostGroupsPostfixes

Description: Creates an array with the postfixes of the m23 subgroups for hosts.
Parameter:

- Array with the postfixes of the m23 subgroups as key and their description as value.

5.40.4 CSchool::checkNewgroupBeforeClientAdd

Description: Checks, if the needed groups are selected in the needed amount
Parameter:

- schoolName: Short name of the school.
- newgroup: Array with selected groups.

Returns:

- Error message(s) in case of errors or empty string when all is ok.

5.40.5 CSchool::getSchoolGrades

Description: Get the grades of a school.
Parameter:

- schoolName: Short name of the school.

Returns:

- Array with the school grades as key and value.

5.40.6 CSchool::getUserSchoolRoles

Description: Gets an array with the school roles.
Parameter:

- : addAdmin: If set to true, the admin role will be included in the array.

Returns:

- Array with the school roles: Nummeric value as key, description as value.

5.40.7 CSchool::isSchoolModeActive

Description: Checks, if the school mode is active.
Returns:

- true, if the school mode is active, otherwise false.

5.40.8 CSchool::getSchoolNameWithDescription

Description: Get an array with the short school names as key and the short school names with description as value.
Returns:

- Associative array with short school names and their descriptions.

5.40.9 CSchool::getSchoolNames

Description: Get the names of all known schools.
Returns:

- Associative array with the names of all known schools as key and value.

5.40.10 CSchool::getLongSchoolName

Description: Get the long name of a school.
Parameter:

- schoolName: Short name of the school.

Returns:

- Long name of the given school.

5.40.11 CSchool::readSchoolInfoFile

Description: Reads and parses the school info file for a given school.
Parameter:

- schoolName: Name of the school or NULL, if all schools should be given out.

Returns:

- Information about the given school or all schools

5.40.12 CSchool::getRangeInfo

Description: Get the value of an option from a range in the school info.
Parameter:

- rangeName: Name of the range defined in /m23/inc/schoolInfoConf.php.

- option: Name of the option.

Returns:

- Value of the option (or dies if no value is present).

5.40.13 CSchool::getNetmask

Description: Get the netmask for m23 clients.
Parameter:

- rangeName: Name of the range defined in /m23/inc/schoolInfoConf.php.

Returns:

- Netmask for m23 clients.

5.40.14 CSchool::getGateway

Description: Get the gateway for m23 clients.
Parameter:

- rangeName: Name of the range defined in /m23/inc/schoolInfoConf.php.

Returns:

- Gateway for m23 clients.

5.40.15 CSchool::getDNS

Description: Get the DNS server for m23 clients.
Parameter:

- rangeName: Name of the range defined in /m23/inc/schoolInfoConf.php.

Returns:

- DNS server for m23 clients.

5.40.16 CSchool::getNextFreeIP

Description: Returns an array with free IPs or a single free IP that can be used for the next client(s).
Parameter:

- amount: Amount of IDs to generate.
- rangeName: Name of the range (with min and max) defined in /m23/inc/schoolInfoConf.php.

Returns:

- Array with free IPs in the given amount or a single free IP.

5.40.17 CSchool::getUsedIPs

Description: Gets the IPs that are managed by the Kerberos/LDAP or the FreeIPA server.
Parameter:

- intIPs: Set to true, to return the IPs as int values.

Returns:

- Array with used IPs.

5.40.18 CSchool::IPexists

Description: Checks if an IP with the selected IP exists in Kerberos/LDAP or the FreeIPA for the current school.
Parameter:

- ip: IP to check
- addErrorMessageIfIPExists: Set to true, if an error message should be added to the message stack, if the IP exists

Returns:

- true, if the IP exists (is used), otherwise false.

5.40.19 CSchool::usesLDAPKerberos

Description: Checks, if LDAP/Kerberos with FusionDirectory is used for user and groups management.
Returns:

- true, if LDAP/Kerberos with FusionDirectory is used for user and groups management, otherwise false.

5.40.20 CSchool::usesFreeIPA

Description: Checks, if FreeIPA is used for user and groups management.
Returns:

- true, if FreeIPA is used for user and groups management, otherwise false.

5.40.21 CSchool::createCSchoolObject

Description: Tries to create a CSchool object.
Parameter:

- schoolName: Name if the school to create the object for.
- errorMsg: In case of an error the error message is written into this variable.

Returns:

- CSchool object on success, otherwise false.

5.41 ./inc/CScredit.php

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

5.41.1 CScredit::__construct

Description: Constructor for new CScredit objects.

5.41.2 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.

5.41.3 CScredit::getCurrentScriptFilename

Description: Gets the current script filename.
Returns:

- Current script filename.

5.41.4 CScredit::getCurrentScriptFilenameFullPath

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

- Current script filename full path.

5.41.5 CScredit::getCurrentScriptFilenameWithoutInstallPhp

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

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

5.41.6 CScredit::getLocalScriptFilenames

Description: Gets the filenames of local scripts.
Returns:

- Array with the filenames of local scripts.

5.41.7 CScredit::updateOnlineScriptInfo

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

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

5.41.8 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.

5.41.9 CScredit::loadOnlineScript

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

- scriptIdToLoad: Id of the script to load.

5.41.10 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.

5.41.11 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.

5.41.12 CScredit::getNewScriptTemplate

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

- Text of the script template.

5.41.13 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.

5.41.14 CScredit::getCurrentScript

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

- Current text of the editor window.

5.41.15 CScredit::setCurrentScript

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

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

5.41.16 CScredit::deleteCurrentScript

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

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

5.41.17 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.

5.41.18 CScredit::saveScript

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

5.41.19 CScredit::show

Description: Shows a script editor with syntax highlighting.

5.42 ./inc/CSystemProxy.php

Author: Hauke Goos-Habermann (HHabermann@pc-kiel.de)
Description: Class for managing systemwide proxies.

5.42.1 CSYSTEMPROXY_getProxySettingsFromAPT

Description: Parses the (maybe existing) proxy settings from the local APT config file.
Returns:

- Array with the proxy settings.

5.42.2 CSYSTEMPROXY_addCurlProxySettings

Description: Sets proxy settings for curl, if a proxy is active.

5.42.3 CSYSTEMPROXY_getUserPasswordString

Description: Creates a string with the user/password combination ($user:$pass@).
Parameter:

- connector: Character to connect the user/password combination with the following words.

Returns:

- String with the user/password combination ($user:$pass@) or empty string, if no proxy authentifictaion is used.

5.42.4 CSYSTEMPROXY_getEnvironmentVariables

Description: Generates BASH proxy variables.
Parameter:

- getAlways: If set to true, the variables will always be returned.

Returns:

- BASH proxy variables.

5.42.5 CSYSTEMPROXY_getAptGetProxyParamter

Description: Generates the apt-get proxy parameters.
Returns:

- : apt-get proxy parameters, if the system proxy is acive otherwise empty string.

5.42.6 CSystemProxy::__construct

Description: Constructor for new CSystemProxy objects.

5.42.7 CSystemProxy::writeEtcProfiles

Description: Writes the proxy settings to the environment file.

5.42.8 CSystemProxy::writeAPTProxyConf

Description: Writes the proxy settings to the APT configuration file or comments them out.

5.42.9 CSystemProxy::writeSquidConf

Description: Writes the (parent) proxy settings to the Squid configuration or removed them.

5.42.10 CSystemProxy::writeACNGConf

Description: Write Configuration for apt-cacher-ng

5.42.11 CSystemProxy::save

Description: Saves the proxy settings in all configuration files.

5.42.12 CSystemProxy::showProxyDialog

Description: Shows a dialog for editing the the proxy settings.

5.42.13 CSystemProxy::loadProxySettings

Description: Loads the proxy settings (from APT config file).

5.42.14 CSystemProxy::isProxyActive

Description: Checks, if a proxy is actively used.
Returns:

- true, if a proxy is actively used, otherwise false.

5.42.15 CSystemProxy::setProxyActive

Description: Activated or deactivates a proxy.
Parameter:

- active: Set to true to activate or false to deactivate proxy usage.

5.42.16 CSystemProxy::areProxySettingsAvailable

Description: Checks, if proxy settings are available.
Returns:

- true, if proxy settings are available, otherwise false.

5.42.17 CSystemProxy::setProxyHost

Description: Sets the proxy host/servername.
Parameter:

- hos: Proxy host/servername.

5.42.18 CSystemProxy::getProxyHost

Description: Gets the proxy host/servername.
Returns:

- Proxy host/servername.

5.42.19 CSystemProxy::getProxyHostIP

Description: Gets the proxy IP.
Returns:

- Proxy IP or false, if the proxy IP could not be detected.

5.42.20 CSystemProxy::usesUserPassword

Description: Checks, if the proxy uses user/password authetification.
Returns:

- true, if the proxy uses user/password authetification, otherwise false.

5.42.21 CSystemProxy::setProxyUser

Description: Sets the proxy authentification username.
Parameter:

- user: The username for proxy authentification.

5.42.22 CSystemProxy::getProxyUser

Description: Gets the proxy authentification username.
Returns:

- Proxy authentification username.

5.42.23 CSystemProxy::setProxyPassword

Description: Sets the proxy authentification password.
Parameter:

- pass: Proxy authentification password.

5.42.24 CSystemProxy::getProxyPassword

Description: Gets the proxy authentification password.
Returns:

- Proxy authentification password.

5.42.25 CSystemProxy::setProxyScheme

Description: Sets the proxy scheme (http/ftp).
Parameter:

- scheme: Proxy scheme (http/ftp)

5.42.26 CSystemProxy::getProxyScheme

Description: Gets the proxy scheme (http/ftp).
Returns:

- Proxy scheme (http/ftp)

5.42.27 CSystemProxy::setProxyPort

Description: Sets the proxy port.
Parameter:

- port: Proxy port.

5.42.28 CSystemProxy::getProxyPort

Description: Gets the proxy port.
Returns:

- Proxy port.

5.42.29 CSystemProxy::getUserPasswordString

Description: Creates a string with the user/password combination ($user:$pass@).
Parameter:

- connector: Character to connect the user/password combination with the following words.

Returns:

- String with the user/password combination ($user:$pass@) or empty string, if no proxy authentific-taion is used.

5.43 ./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

5.43.1 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.

5.43.2 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.

5.43.3 isMySQL3used

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

5.43.4 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

5.43.5 getClientLanguage

5.43.6 getInstDev

Description: fetch the device for installation
Parameter:

- id: package ID

5.43.7 DB_getConnection

Description: Gets the MySQLi connection.
Returns:

- MySQLi connection.

5.43.8 DB_setConnection

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

- conn: MySQLi connection to use globally.

5.43.9 DB_isConnectionValid

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

- true on valid MySQLi connection, otherwise false.

5.43.10 dbConnect

Description: connects to the m23 database

5.43.11 dbClose

Description: closes the connection to the m23 database

5.43.12 DB_getSuperUserName

Description: Returns the name of the super MySQL/MariaDB user.
Returns:

- Name of the super MySQL/MariaDB user.

5.43.13 DB_getSuperUserPassword

Description: Returns the password (or empty) of the super MySQL/MariaDB user.
Returns:

- Password (or empty) of the super MySQL/MariaDB user.

5.43.14 getServerIP

Description: returnes the IP of the m23 server
Returns:

- IP of the m23 server.

5.43.15 getServerNetmask

Description: returnes the netmask of the m23 server

5.43.16 getServerNetwork

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

- Network IP of the m23 server.

5.43.17 getDNSServers

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

5.43.18 getServerGateway

Description: Returnes the gateway of the m23 server

5.43.19 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

5.43.20 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

5.43.21 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

5.43.22 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

5.43.23 deleteClientLogs

Description: deletes the installation logs
Parameter:

- clientName: name of the client to delete all logs

5.43.24 workPhpName

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

- Unique name for storing the work.php file.

5.43.25 executeNextWork

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

5.43.26 DB_query

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

- sql: sql query

5.43.27 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.

5.43.28 DB_getErrorMessage

Description: Gets the error message from the last query.
Returns:

- Error message from the last query.

5.43.29 DB_genPassword

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

- length: length of password

5.43.30 getClientIP

Description: returnes the IP of the calling client

5.43.31 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

5.43.32 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

5.43.33 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

5.43.34 isProgrammInstalled

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

- progName: name of the programm

5.43.35 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: IP to ping

Returns:

- true, if the IP can be pinged, otherwise false.

5.43.36 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

5.43.37 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.

5.43.38 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.

5.43.39 DB_getArrayAssoc

Description: Fetches all results from a query.
Parameter:

- result: MySQLi result.

Returns:

- Array with continuous numbers as keys and associative arrays with the result rows as value.

5.43.40 DB_getTableColumns

Description: Creates an associative array with table and column names and their data types and codepages (collation).
Parameter:

- dbName: Name of the database.

Returns:

- Associative array with table and column names and their data types and codepages (collation).

5.43.41 DB_changeAllCollations

Description: Changes the codepage for the collation of all text fields in all tables of a database.
Parameter:

- destCodepage: Wanted codepage.
- dbName: Name of the database.

5.43.42 DB_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

5.43.43 DB_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

5.43.44 DB_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

5.43.45 DB_getNextID

Description: Returns the next higher than the highest ID
Parameter:

- type: "user" for user IDs

5.43.46 DB_getNextUserID

Description: Returns the next higher than the highest user ID

5.43.47 DB_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

5.43.48 DB_deleteID

Description: Delets an ID from the table of used IDs
Parameter:

- type: "user" for user IDs
- id: number of the new ID

5.43.49 DB_addNewUserID

Description: Adds a new user ID to the table
Parameter:

- id: number of the new ID

5.43.50 DB_addNewGroupID

Description: Adds a new user ID to the table
Parameter:

- id: number of the new ID

5.43.51 DB_getNextGroupID

Description: Returns the next higher than the highest group ID

5.44 ./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.

5.44.1 DHCP_exportDHCPSettingsForExternalDHCPServer

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

5.44.2 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.

5.44.3 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

5.44.4 DHCP_delSubnetDefinition

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

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

5.44.5 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.

5.44.6 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.

5.44.7 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.

5.44.8 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.

5.44.9 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.

5.44.10 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.

5.44.11 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.

5.44.12 DHCP_addLineToDHCPDConf

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

- line: Line to add.

5.44.13 DHCP_restartDHCPserver

Description: Restarts the DHCP server.
Returns:

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

5.44.14 DHCP_rmClient

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

- clientName: name of the client

5.44.15 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)

5.44.16 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)

5.44.17 DHCP_calcPXEIP

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

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

5.44.18 DHCP_writePXEcfg

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

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

5.44.19 DHCP_removePXEcfg

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

- clientName: name of the client

5.44.20 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.

5.45 ./inc/distr/debian/clientConfigCommon.php

Author: Hauke Goos-Habermann (HHabermann@pc-kiel.de)
Description: Common management funtions shared by Linux distributions.

5.45.1 CLCFG_enableFreeIPA

Description: Shows BASH commands to enable FreeIPA on a client.
Parameter:

- clientName: Name of the client.

Returns:

- true, if the BASH commands could be prepared, false otherwise.

5.45.2 CLCFG_enableLDAPplusKerberos

Description: Enables login via Kerberos for a client.
Parameter:

- clientOptions: the client's options array
- clientOptions[KRB_server]: IP or hostname of the Kerberos server.
- clientOptions[KRB_adminServer]: IP or hostname of the Kerberos admin server.
- clientOptions[KRB_realm]: Kerberos realm.

5.45.3 CLCFG_enablePAMModule

Description: Enables a PAM modules (if it wasn't enabled before)

5.45.4 CLCFG_enableShutdownByUsersGroup

Description: Allows all users of the group users to run shutdown as root.

5.45.5 CLCFG_enableSudoByM23sudoGroup

Description: Allows all users of the group m23sudo to run command as root.

5.45.6 CLCFG_blacklistPackageInstallation

Description: Blacklists a package for APT to make it uninstallable.
Parameter:

- package: Name of the package.

5.45.7 CLCFG_executeNextWorkEveryMinute

Description: Runs executeNextWork in a loop every minute and shows a status message about elapsed waiting time.
Parameter:

- lang: language for the messages

5.45.8 CLCFG_setSystime

Description: Sets the system time of the client to about the same time of the m23 server.

5.45.9 CLCFG_disableAptSystemdDaily

Description: Deactivates systemd apt-daily.timer.

5.45.10 CLCFG_addPAMtoDM

Description: Adds PAM modules (if present) to the pam.d configuration files.
Parameter:

- dmFile: Name of the login manager pam.d config file.

5.45.11 CLCFG_disableSudoRootLogin

Description: Disables getting root rights of normal users via sudo.

5.45.12 CLCFG_installFirmware

Description: Installs available firmware packages.

5.45.13 CLCFG_monoRemove

Description: Removes mono packages and installs gnote when tomboy was installed before.

5.45.14 CLCFG_setDebConfDM

Description: Sets the display manager in debconf.
Parameter:

- dm: Name of the display manager (kdm, lightdm, mdm, ...).

5.45.15 CLCFG_installXDM

Description: Installs the XDM display manager.

5.45.16 CLCFG_installSDDM

Description: Installs the SDDM display manager.

5.45.17 CLCFG_installMintDM

Description: Installs the Linux Mint DM display manager.

5.45.18 CLCFG_copyMBRToAllDevices

Description: Copies the MBR code from the boot device to all other devices.
Parameter:

- bootDevice: Device name of the device with the original MBR (e.g. /dev/sda).

5.45.19 CLCFG_activateBOOT_DEGRADED

Description: Activates BOOT_DEGRADED on Ubuntu 12.04 to allow booting from degraded RAIDs.

5.45.20 CLCFG_installLightDM

Description: Installs the light DM display manager.
Parameter:

- session: Name of the session to select by default.
- addSessionWrapper: Set to true, if an additional line with "session-wrapper=/etc/X11/Xsession" should be added.

5.45.21 CLCFG_setDebConfDirect

Description: Sets debconf settings.
Parameter:

- debconf: Debconf settings to add.

5.45.22 TRINITY_installLoginManager

Description: Installs the Trinity login manager KDM.
Parameter:

- lang: short language

5.45.23 TRINITY_install

Description: Installs a minimalistic Trinity desktop.
Parameter:

- lang: short language

5.45.24 CLCFG_makeDev

Description: Creates the device nodes in /dev by downloading and extracting an archive containing the device nodes and if this fails running MAKEDEV.

5.45.25 CLCFG_disablePlymouth

Description: Disables the plymouth.

5.45.26 CLCFG_disableAvahiDaemon

Description: Disables the avahi-daemon.

5.45.27 CLCFG_configUpstartForNormalUsage

Description: Configures upstart for normal running in an installed system.

5.45.28 CLCFG_configUpstartForChroot

Description: Configures upstart to make it not fail installation.

5.45.29 CLCFG_createScreenRC

Description: Creates the (under Ubuntu) needed settings for screen.

5.45.30 CLCFG_addGrubPassword

Description: Adds a password line to the grub configuration to lock the edit line of bootmanager.

5.45.31 CLCFG_addLiloPassword

Description: Adds a password line to the LiLo configuration to lock the edit line of bootmanager.

5.45.32 GNOME_prepare

Description: Prepares the GNOME installation

5.45.33 LXDE_install

Description: Installs the LXDE desktop.
Parameter:

- lang: short language
- fullInstall: Set to true, if the full desktop with all applications should be installed. Otherwise a minimal desktop will be installed

5.45.34 GNOME3_install

Description: Installs the GNOME 3 desktop.
Parameter:

- lang: short language
- fullInstall: Set to true, if the full desktop with all applications should be installed. Otherwise a minimal desktop will be installed

5.45.35 GNOME_install

Description: Installs a minimalistic GNOME desktop.
Parameter:

- lang: short language

5.45.36 GNOME_installLoginManager

Description: Installs the GNOME login manager GDM.
Parameter:

- lang: short language

5.45.37 KDE_prepare

Description: Prepares the KDE installation

5.45.38 KDE_install

Description: Installs a minimalistic KDE desktop.
Parameter:

- lang: short language
- ver: KDE version 3 or 4 (minor releases are depending in the on the used distribution)

5.45.39 KDE_installLoginManager

Description: Installs the KDE/Trinity login manager KDM.
Parameter:

- lang: short language
- ver: KDE version 3 or 4 (minor releases are depending in the on the used distribution)
- trinity: Set to true, if the Trinity Desktop should be configured.

5.45.40 CLCFG_lilo2Grub

Description: Installs and runs the lilo.conf to Grub's menu.lst converter

5.45.41 CLCFG_getMbrPart

Description: Returns the device to install the MBR of the bootloader in.
Parameter:

- installPart: Partition to install the OS on.
- clientOptions: Array with information about client options.

Returns:

- Device to install the MBR of the bootloader in.

5.45.42 CLCFG_setTimeZone

Description: Sets the timezone a a client.
Parameter:

- timezone: POSIX timezone as defined at /usr/share/zoneinfo/posix/

5.45.43 CLCFG_writeHosts

Description: writes the /etc/hosts file for the client

5.45.44 CLCFG_addUser

Description: generates the commands to add a user on the client. it adds the user account, creates a home directory, copies the m23 skel files and sets the
Parameter:

- userName: the username for the account
- password: the unecrypted password for the account
- groups: the groups the user should be added to
- skelDir: directory to the skeleton files
- uid: Optional user ID of the new user.
- gid: Optional group ID of the new user.

5.45.45 CLCFG_createBootDeviceNode

Description: Makes sure the needed device nodes for the boot device are created.

5.45.46 CLCFG_efi

Description: Configures the client for UEFI booting.
Parameter:

- CFDiskIOO: Object to with partition information for the calling client.

Returns:

- true, if the client uses EFI, otherwise false.

5.45.47 CLCFG_genFstab

Description: generates the commands to auto detect the partitions and generate the fstab file and installs a boot manager
Parameter:

- bootDevice: the device the bootloader should be installed on (e.g. /dev/hda)
- rootDevice: the path to the installation partition (e.g. /dev/hda1)
- bootloader: Name of the boot loader to install and configure (actually grub only)
- ignoreErrors: Set to true, if some errors should be ignored instead of logging them.
- isReconfiguredWithExtraDistr: Set to true, the function is called to reconfigure a distribution that was installed by imaging.

5.45.48 CLCFG_interfaces

Description: generates a script that writes the /etc/network/interfaces file.
Parameter:

- clientParams: Associated array with the parameters of the client.

5.45.49 CLCFG_hostname

Description: generates a script that writes the /etc/hostname file.
Parameter:

- clientName: the name of the client

5.45.50 CLCFG_resolvConf

Description: generates a script that writes the /etc/resolv.conf file.
Parameter:

- DNSServers: list of DNS servers

5.45.51 CLCFG_aptConf

Description: generates a script that writes the /etc/apt/apt.conf.d/70debconf file.
Parameter:

- proxyServer: IP or name of the proxy server
- proxyPort: port the proxy server listens on

5.45.52 CLCFG_sourceslist

Description: generates a script that writes the /etc/apt/sources.list file.
Parameter:

- clientIP: IP of the client
- clientName: name of the client
- serverIP: IP of the server

5.45.53 CLCFG_sourceslistCreateConfigFiles

Description: Creates config files for the package manager.
Parameter:

- sourceName: The name of the package source list

5.45.54 CLCFG_hwdetect

Description: updates and runs the hardware detection

5.45.55 CLCFG_language

Description: sets the language for keyboard in console and X11 and console language
Parameter:

- lang: 2 letter language code (de,fr,it,en)
- release: The release name of the distribution (for special handling).

5.45.56 CLCFG_setRootPassword

Description: sets the root password of the client
Parameter:

- cryptedPassword: sets an allready crypted root password

5.45.57 CLCFG_setAuthorized_keys

Description: sets the ssh authorized_file for remote login into the clients
Parameter:

- serverIP: IP of the server
- pathToKeyFile: path on the server where to get the key file from
- acceptyDSSKeys: Set to true, if SSH-DSS keys should be re-enabled

5.45.58 CLCFG_copyClientPackageStatus

Description: copies the package installation status file from the lient to the cache directory of the server
Parameter:

- clientName: the name of the client

5.45.59 CLCFG_aptGet

Description: Executes an APT command with parameters and tries to install/remove as many packages as possible. If the packages could not get installed in a bunch try to install the packges one by one.
Parameter:

- command: install or remove
- packages: white space seperated list of packages to install or remove

5.45.60 CLCFG_importLocalPoolKey

Description: Generates BASH code to import the local package pool key (if it exists) on the client's APT system.
Returns:

- BASH code to import the local package pool key or empty string, if there is no package pool sign key.

5.45.61 CLCFG_installBasePackages

Description: installs needed base packages
Parameter:

- packagelist: the list of the packages to install
- keyring: name of the keyring package

5.45.62 CLCFG_setDebconf

Description: installs the debconf packages and sets debconf configuration
Parameter:

- serverIP: IP of the server, where to download the config file
- debconfFile: path and filename of the debconf file

5.45.63 CLCFG_debootstrap

Description: bootstraps a Debian system
Parameter:

- suite: select the Debian suite (sarge, sid, woody, potato)
- DNSServers: DNS server for resolving the names of the installation server
- gateway: gateway for fetching the packages
- packageProxy: the ip of the proxy the packages should be fetched from
- packagePort: the proxy port
- mirror: the Debian mirror the packages should be fetched from
- arch: the computer architecture of the client
- exclude: packages to exclude (as comma-seperated list)
- include: additional packages to include
- isCritical: selects if debootstrap errors should be critical
- additionalPackages: space seperated list of additional packages that should be installed during bootstrapping

5.45.64 CLCFG_downgradeExt

Description: downgrades all ext* partitions so the fsck.ext* of woody can understand it

5.45.65 CLCFG_mountRootDir

Description: create a new directory for mounting the root partition of the client, mount the installation directory to this mount point and create a temp directory
Parameter:

- rootDev: the root device (e.g. /dev/hda1)
- mountPoint: directory under /mnt to mount the device in
- CFDiskIOO: Client object (used for mounting the EFI boot partition).

5.45.66 CLCFG_activateDMA

Description: try to activate DMA to speed up installation. this short script tries to detect

5.45.67 CLCFG_fetchm23BasicTools

Description: fetches the basic m23 tools and fsize

5.45.68 CLCFG_dialogGaugeProcPos

Description: Generates the BASH code for showing a dialog with status bar that uses /proc/pid/fdinfo for current file position
Parameter:

- backtitle: title on top of the screen
- title: message title
- message: message to show over the status bar
- infofilecmd: BASH code to figure out the needed status file in the proc filesystem
- fullsize: The full size of the file that should be written.
- force: Forces showing of the dialog box

5.45.69 CLCFG_dialogInfoBox

Description: generates the BASH code for showing an dialog infobox
Parameter:

- backtitle: title on top of the screen
- title: message title
- message: the message itself
- force forces showing of the dialog box

5.45.70 CLCFG_dialogMsgBox

Description: generates the BASH code for showing an dialog message box
Parameter:

- backtitle: title on top of the screen
- title: message title
- message: the message itself
- force forces showing of the dialog box

5.45.71 CLCFG_dialogAllBox

Description: generates the BASH code for showing different types of dialog boxes
Parameter:

- backtitle: title on top of the screen
- title: message title
- message: the message itself
- type: type of the dialog box

5.45.72 CLCFG_executeAfterChroot

Description: prepares for pivot_root and does it and executes afterChroot

5.45.73 CLCFG_writeCrontabm23fetchjobEvery5Minutes

Description: Adds entries to crontab to check every 5 minutes for new jobs.

5.45.74 CLCFG_writeM23fetchjob

Description: generates the m23fetchjob script
Parameter:

- release: Name of the distribution release for special handling of some releases.

5.45.75 CLCFG_hideKernelWarnings

Description: hides the kernel warnungs

5.45.76 CLCFG_getRootDeviceFS

Description: Gets the filesystem of the root device.
Parameter:

- rootDevice: the device, the kernel should be installed on
- clientName: the name of the client

Returns:

- Filesystem of the root device.

5.45.77 CLCFG_genFakeFstab

Description: generates a fake /etc/fstab that only contains the lines for proc and the root partition (this is used to make the kernel install correctly)
Parameter:

- rootDevice: the device, the kernel should be installed on
- clientName: the name of the client

5.45.78 CLCFG_copySSLCert

Description: fetches the SSL certificate from the server and copies it to the client
Parameter:

- rootPath: the path to where the root directory is mounted
- disableSSLCertCheck: Disables the SSL certificate check of wget.

5.45.79 CLCFG_changeUser

Description: changes the settings of an useraccount on a client
Parameter:

- userName: the (old) username for the account
- password: the new unecrypted password for the account
- newUserName: the new username

5.45.80 CLCFG_patchNsswitchForLDAP

Description: Patches /etc/nsswitch.conf for usage with LDAP.

5.45.81 CLCFG_enableNFSHome

Description: enables storing of home directories on a NFS server
Parameter:

- nfsURL: URL to the NFS storage (e.g. 192.168.1.23:/nfs-homes)

5.45.82 CLCFG_disableNFSHome

Description: Disables storing of home directories on a NFS server

5.45.83 CLCFG_installDesktopLanguagePackage

Description: Installs some additional language packages for (KDE / Gnome) desktops.
Parameter:

- lang: short language
- kde: Install KDE language packages too.
- gnome: Install Gnome language packages too.

5.45.84 CLCFG_installApplicationLanguagePackages

Description: Installs some additional language packages for installed applications with seperate language packs.
Parameter:

- lang: short language

5.45.85 CLCFG_updateDebootstrapScripts

Description: Updates the debootstrap scripts for Debian or Ubuntu and returns the www path to the files

5.45.86 CLCFG_updateDebootstrapScriptsDebian

Description: Updates the debootstrap scripts for Debian and returns the www path to the files

5.46 ./inc/distr/debian/clientConfig.php

Author: Hauke Goos-Habermann (HHabermann@pc-kiel.de)
Description: Debian specific management functions.

5.46.1 DEBIAN_desktopInstall

Description: Installs a Debian desktop.
Parameter:

- desktop: Desktop constant.

5.46.2 CLCFG_listDebianReleasesGeneric

Description: Generates an array of the different releases (e.g. sarge, sid, woody, potato) of a distribution.
Parameter:

- selName: the name of the selection
- first: the release to show first
- distr: distribution directory

Returns:

- : Array with release names.

5.46.3 CLCFG_listDebianReleasesGeneric

Description: generates a selection of the different releases (e.g. sarge, sid, woody, potato) of a distribution.
Parameter:

- selName: the name of the selection
- first: the release to show first
- distr: distribution directory

Returns:

- : Selection with release names.

5.46.4 CLCFG_listDebianReleases

Description: generates a selection of the different Debian releases (sarge, sid, woody, potato)
Parameter:

- selName: the name of the selection
- first: the release to show first

5.46.5 CLCFG_addDistributionSpecificOptions

Description: adds distribution specific settings to an option array and returns the new array
Parameter:

- $options: the options array with some values

5.46.6 CLCFG_showDistributionSpecificOptions

Description: shows distribution specific options and returns false, if there was an error
Parameter:

- options: options array
- distr: The name the distribution to use.

5.46.7 CLIENT_enableLDAP

Description: enables LDAP login for a client.
Parameter:

- clientOptions: the client's options array

5.47 ./inc/distr/debian/packages.php

Author: Hauke Goos-Habermann (HHabermann@pc-kiel.de)
Description: Debian/Ubuntu specific package handling functions.

5.47.1 PKG_updatePackageSearchCacheFile

Description: Updates the compressed package search file.
Parameter:

- packagesource: Name of the package sources list.

Returns:

- Full path to the log file.

5.47.2 PKG_getSearchCacheFileName

Description: Returns the full path to the compressed package search file.
Parameter:

- dir: Directory where the package information is stored.
- arch: Architecture for the packages in the search file.

Returns:

- Full path to the compressed package search file.

5.47.3 PKG_genPackageSearchCacheFileCMD

Description: Generates commands to create a compressed package search file.
Parameter:

- dir: Directory where the package information is stored.
- arch: Architecture for the packages in the search file.

Returns:

- BASH commands for creating a compressed package search file.

5.47.4 PKG_searchPackageSearchCacheFileCMD

Description: Generates BASH commands to search a compressed package search file.
Parameter:

- dir: Directory where the package information is stored.
- arch: Architecture for the packages in the search file.
- search: Search terms.

Returns:

- BASH commands to search a compressed package search file.

5.47.5 PKG_ncTarDebsFromClientToServer_Client

Description: Client to send the Debian packages to the m23 client.

5.47.6 PKG_ncTarDebsFromClientToServer_Server

Description: Starts a server to receive the Debian packages from an m23 client.
Parameter:

- poolDir: The directory on the server to store the Debian packages.

5.47.7 PKG_rsyncDebsFromClientToServer

Description: Copies all the Debian packages from a client to a given directory on the server via rsync.
Parameter:

- clientIP: The IP of the client.
- poolDir: The directory on the server to store the Debian packages.

5.47.8 PKG_preparePool

Description: Generates the needed configuration file for reprepro.
Parameter:

- release: Release of the pool.
- distr: Distribution of the pool.
- arch: Architecture of the pool (POOL_ARCH_I386 or CPoolLister::POOL_ARCH_AMD64).
- poolName: Name of the pool (if a pool with the given name exists => load)
- poolDir: Directory of the pool with full path.

5.47.9 PKG_convertPackagesToRepository

Description: Generates commands for creating a package source from packages stored in one directory.
Parameter:

- poolDir: Complete path to the directory that stores the pool.
- logFile: File (with full path) to store the pool generation log in.
- poolName: Name of the pool.
- sourceslist: Variable to store the sources list in.

Returns:

- Commands for creating a package source from packages stored in one directory.

5.47.10 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.

5.47.11 PKG_fastGetNewInstalledPackages

Description: Gets new installed packages by comparing status files (before and after the installation).
Parameter:

- oldStatusFile: File with the list of all installed packages before the installation of the new packages
- newStatusFile: File with the list of all installed packages after the installation of the new packages
- storeFile: File name to store the list of new installed on the client.

5.47.12 PKG_searchFor

Description: searches for a package and returnes a file descriptor
Parameter:

- key: search key
- distr: the distribution name
- packagesource: name of the package source
- arch: Array of architectures to get package infos for.

5.47.13 PKG_getDescription

Description: gets the description of a package
Parameter:

- line: line containing package name and description

5.47.14 PKG_getPackageName

Description: gets the name of a package
Parameter:

- line: line containing package name and description

5.47.15 PKG_getLongPackageDescription

Description: Gets the description of a package.
Parameter:

- package: Name of the package.
- distr: the distribution name
- packagesource: name of the package source

Returns:

- Description of the package.

5.47.16 PKG_getPackageDescriptionSize

Description: Returnes the description and size of a package
Parameter:

- distr: the name of the distribution
- packagesource: the name of the package source list
- package: name if the package

5.47.17 PKG_printStatus

Description: prints the package status, e.g. amount of packages,...
Parameter:

- distr: the name of the distribution
- packagesource: the name of the package source list

5.47.18 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)
- completeDescription: If set to true, the full package description and sizes are fetched (time consuming)

Returns:

- : Amount of found packages.

5.47.19 PKG_getAptArchOptions

Description: Generates options to specify the architecture of a client that can be appended to an apt-get line.
Parameter:

- arch: Architecture to get package infos for.

Returns:

- : Parameter for architecture specific apt-get commands, if architecture is not i386.

5.47.20 PKG_addAPTConfigFiles

Description: Creates the config files for the package manager on the m23 server.
Parameter:

- sourceName: The name of the package source list
- dir: The directory on the m23 server where the config files should be created.

5.47.21 PKG_preparePackageDir

Description: creates the needed files + sources list in a directory to use it for "local apt".
Parameter:

- dir: the directory to prepare
- packagesource: sources list
- logFile: Name of the file the messages from apt-get should be written to.
- returnCmd: Set to true, if the apt command should be returned or to false to execute it in this function.
- arch: Architecture to get package infos for.
- sourceName: The name of the package source list
- hardTryUpdate: Set to true, if the apt-get update should be run until it succeeds

Returns:

- : Error text on error or empty string on success.

5.47.22 PKG_updatePackageInfo

Description: Updates the package information to make it searchable.
Parameter:

- distr: the distribution name
- packagesource: name of the package source
- force: set to true if you want to update the package info and the time is not over
- arch: Architecture to get package infos for.
- changeTime: Writes the time when the status file was last modified to.

5.47.23 PKG_previewInstall

Description: shows what happens if packages get 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

5.47.24 PKG_getKernels

Description: Generates an associative array with the available kernels for an architecture and distribution as keys and values, where kernels containing the term 'generic' are at the front.
Parameter:

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

Returns:

- Associative array with the available kernels for an architecture and distribution as keys and values with kernels containing the term 'generic' at the front, sorted by length and then by alphabet.

5.47.25 PKG_downloadPool

Description: Generates commands to download packages from a sources list to a directory.
Parameter:

- destDir: the directory the packages should be downloaded to
- sourceslist: package source list
- packagesArr: array of the lists that contain packages to download (seperated by blanks)
- arch: CPU architecture to download the packages for.
- release: Select the Debian/Ubuntu suite (squeeze, sarge, sid, precise).

5.47.26 PKG_getDebootStrapBasePackages

Description: Returns the list of base packages that are downloaded by debootstrap for a special release.
Parameter:

- release: release name of the distribution version (e.g. sarge)

5.48 ./inc/distr/halfSister/clientConfig.php

Author: Hauke Goos-Habermann (HHabermann@pc-kiel.de)
Description: halfSister specific management functions.

5.48.1 CLCFG_copyClientPackageStatus

Description: Not used, but needs to be defined here.

5.48.2 CLCFG_listHSReleasesGeneric

Description: Generates an array of the different releases (e.g. sarge, sid, woody, potato) of a distribution.
Parameter:

- selName: the name of the selection
- first: the release to show first
- distr: distribution directory

Returns:

- : Array with release names.

5.48.3 CLCFG_listHSReleasesGeneric

Description: generates a selection of the different releases (e.g. sarge, sid, woody, potato) of a distribution.
Parameter:

- selName: the name of the selection
- first: the release to show first
- distr: distribution directory

Returns:

- : Selection with release names.

5.48.4 CLCFG_listHSReleases

Description: generates a selection of the different HS releases (sarge, sid, woody, potato)
Parameter:

- selName: the name of the selection
- first: the release to show first

5.48.5 CLCFG_addDistributionSpecificOptions

Description: adds distribution specific settings to an option array and returns the new array
Parameter:

- $options: the options array with some values

5.48.6 CLCFG_showDistributionSpecificOptions

Description: shows distribution specific options and returns false, if there was an error
Parameter:

- options: options array

5.49 ./inc/distr/halfSister/packages.php

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

5.49.1 PKG_updatePackageSearchCacheFile

Description: Updates the compressed package search file.
Parameter:

- packagesource: Name of the package sources list.

Returns:

- The name of the package cache.

5.49.2 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.

5.49.3 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.

5.49.4 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)

Returns:

- : Amount of found packages.

5.49.5 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

5.49.6 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.

5.49.7 PKG_translateClientPackageStatus

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

- status: status code you want to translate

5.50 ./inc/distr/imaging/clientConfigCommon.php

Author: Hauke Goos-Habermann (HHabermann@pc-kiel.de)
Description: Common management funtions shared by Linux distributions.

5.50.1 CLCFG_enableFreeIPA

Description: Shows BASH commands to enable FreeIPA on a client.
Parameter:

- clientName: Name of the client.

Returns:

- true, if the BASH commands could be prepared, false otherwise.

5.50.2 CLCFG_enableLDAPplusKerberos

Description: Enables login via Kerberos for a client.
Parameter:

- clientOptions: the client's options array
- clientOptions[KRB_server]: IP or hostname of the Kerberos server.
- clientOptions[KRB_adminServer]: IP or hostname of the Kerberos admin server.
- clientOptions[KRB_realm]: Kerberos realm.

5.50.3 CLCFG_enablePAMModule

Description: Enables a PAM modules (if it wasn't enabled before)

5.50.4 CLCFG_enableShutdownByUsersGroup

Description: Allows all users of the group users to run shutdown as root.

5.50.5 CLCFG_enableSudoByM23sudoGroup

Description: Allows all users of the group m23sudo to run command as root.

5.50.6 CLCFG_blacklistPackageInstallation

Description: Blacklists a package for APT to make it uninstallable.
Parameter:

- package: Name of the package.

5.50.7 CLCFG_executeNextWorkEveryMinute

Description: Runs executeNextWork in a loop every minute and shows a status message about elapsed waiting time.
Parameter:

- lang: language for the messages

5.50.8 CLCFG_setSystime

Description: Sets the system time of the client to about the same time of the m23 server.

5.50.9 CLCFG_disableAptSystemdDaily

Description: Deactivates systemd apt-daily.timer.

5.50.10 CLCFG_addPAMtoDM

Description: Adds PAM modules (if present) to the pam.d configuration files.
Parameter:

- dmFile: Name of the login manager pam.d config file.

5.50.11 CLCFG_disableSudoRootLogin

Description: Disables getting root rights of normal users via sudo.

5.50.12 CLCFG_installFirmware

Description: Installs available firmware packages.

5.50.13 CLCFG_monoRemove

Description: Removes mono packages and installs gnote when tomboy was installed before.

5.50.14 CLCFG_setDebConfDM

Description: Sets the display manager in debconf.
Parameter:

- dm: Name of the display manager (kdm, lightdm, mdm, ...).

5.50.15 CLCFG_installXDM

Description: Installs the XDM display manager.

5.50.16 CLCFG_installSDDM

Description: Installs the SDDM display manager.

5.50.17 CLCFG_installMintDM

Description: Installs the Linux Mint DM display manager.

5.50.18 CLCFG_copyMBRToAllDevices

Description: Copies the MBR code from the boot device to all other devices.
Parameter:

- bootDevice: Device name of the device with the original MBR (e.g. /dev/sda).

5.50.19 CLCFG_activateBOOT_DEGRADED

Description: Activates BOOT_DEGRADED on Ubuntu 12.04 to allow booting from degraded RAIDs.

5.50.20 CLCFG_installLightDM

Description: Installs the light DM display manager.
Parameter:

- session: Name of the session to select by default.
- addSessionWrapper: Set to true, if an additional line with "session-wrapper=/etc/X11/Xsession" should be added.

5.50.21 CLCFG_setDebConfDirect

Description: Sets debconf settings.
Parameter:

- debconf: Debconf settings to add.

5.50.22 TRINITY_installLoginManager

Description: Installs the Trinity login manager KDM.
Parameter:

- lang: short language

5.50.23 TRINITY_install

Description: Installs a minimalistic Trinity desktop.
Parameter:

- lang: short language

5.50.24 CLCFG_makeDev

Description: Creates the device nodes in /dev by downloading and extracting an archive containing the device nodes and if this fails running MAKEDEV.

5.50.25 CLCFG_disablePlymouth

Description: Disables the plymouth.

5.50.26 CLCFG_disableAvahiDaemon

Description: Disables the avahi-daemon.

5.50.27 CLCFG_configUpstartForNormalUsage

Description: Configures upstart for normal running in an installed system.

5.50.28 CLCFG_configUpstartForChroot

Description: Configures upstart to make it not fail installation.

5.50.29 CLCFG_createScreenRC

Description: Creates the (under Ubuntu) needed settings for screen.

5.50.30 CLCFG_addGrubPassword

Description: Adds a password line to the grub configuration to lock the edit line of bootmanager.

5.50.31 CLCFG_addLiloPassword

Description: Adds a password line to the LiLo configuration to lock the edit line of bootmanager.

5.50.32 GNOME_prepare

Description: Prepares the GNOME installation

5.50.33 LXDE_install

Description: Installs the LXDE desktop.
Parameter:

- lang: short language
- fullInstall: Set to true, if the full desktop with all applications should be installed. Otherwise a minimal desktop will be installed

5.50.34 GNOME3_install

Description: Installs the GNOME 3 desktop.
Parameter:

- lang: short language
- fullInstall: Set to true, if the full desktop with all applications should be installed. Otherwise a minimal desktop will be installed

5.50.35 GNOME_install

Description: Installs a minimalistic GNOME desktop.
Parameter:

- lang: short language

5.50.36 GNOME_installLoginManager

Description: Installs the GNOME login manager GDM.
Parameter:

- lang: short language

5.50.37 KDE_prepare

Description: Prepares the KDE installation

5.50.38 KDE_install

Description: Installs a minimalistic KDE desktop.
Parameter:

- lang: short language
- ver: KDE version 3 or 4 (minor releases are depending in the on the used distribution)

5.50.39 KDE_installLoginManager

Description: Installs the KDE/Trinity login manager KDM.
Parameter:

- lang: short language
- ver: KDE version 3 or 4 (minor releases are depending in the on the used distribution)
- trinity: Set to true, if the Trinity Desktop should be configured.

5.50.40 CLCFG_lilo2Grub

Description: Installs and runs the lilo.conf to Grub's menu.lst converter

5.50.41 CLCFG_getMbrPart

Description: Returns the device to install the MBR of the bootloader in.
Parameter:

- installPart: Partition to install the OS on.
- clientOptions: Array with information about client options.

Returns:

- Device to install the MBR of the bootloader in.

5.50.42 CLCFG_setTimeZone

Description: Sets the timezone a a client.
Parameter:

- timezone: POSIX timezone as defined at /usr/share/zoneinfo/posix/

5.50.43 CLCFG_writeHosts

Description: writes the /etc/hosts file for the client

5.50.44 CLCFG_addUser

Description: generates the commands to add a user on the client. it adds the user account, creates a home directory, copies the m23 skel files and sets the
Parameter:

- userName: the username for the account
- password: the unecrypted password for the account
- groups: the groups the user should be added to
- skelDir: directory to the skeleton files
- uid: Optional user ID of the new user.
- gid: Optional group ID of the new user.

5.50.45 CLCFG_createBootDeviceNode

Description: Makes sure the needed device nodes for the boot device are created.

5.50.46 CLCFG_efi

Description: Configures the client for UEFI booting.
Parameter:

- CFDiskIOO: Object to with partition information for the calling client.

Returns:

- true, if the client uses EFI, otherwise false.

5.50.47 CLCFG_genFstab

Description: generates the commands to auto detect the partitions and generate the fstab file and installs a boot manager
Parameter:

- bootDevice: the device the bootloader should be installed on (e.g. /dev/hda)
- rootDevice: the path to the installation partition (e.g. /dev/hda1)
- bootloader: Name of the boot loader to install and configure (actually grub only)
- ignoreErrors: Set to true, if some errors should be ignored instead of logging them.
- isReconfiguredWithExtraDistr: Set to true, the function is called to reconfigure a distribution that was installed by imaging.

5.50.48 CLCFG_interfaces

Description: generates a script that writes the /etc/network/interfaces file.
Parameter:

- clientParams: Associated array with the parameters of the client.

5.50.49 CLCFG_hostname

Description: generates a script that writes the /etc/hostname file.
Parameter:

- clientName: the name of the client

5.50.50 CLCFG_resolvConf

Description: generates a script that writes the /etc/resolv.conf file.
Parameter:

- DNSServers: list of DNS servers

5.50.51 CLCFG_aptConf

Description: generates a script that writes the /etc/apt/apt.conf.d/70debconf file.
Parameter:

- proxyServer: IP or name of the proxy server
- proxyPort: port the proxy server listens on

5.50.52 CLCFG_sourceslist

Description: generates a script that writes the /etc/apt/sources.list file.
Parameter:

- clientIP: IP of the client
- clientName: name of the client
- serverIP: IP of the server

5.50.53 CLCFG_sourceslistCreateConfigFiles

Description: Creates config files for the package manager.
Parameter:

- sourceName: The name of the package source list

5.50.54 CLCFG_hwdetect

Description: updates and runs the hardware detection

5.50.55 CLCFG_language

Description: sets the language for keyboard in console and X11 and console language
Parameter:

- lang: 2 letter language code (de,fr,it,en)
- release: The release name of the distribution (for special handling).

5.50.56 CLCFG_setRootPassword

Description: sets the root password of the client
Parameter:

- cryptedPassword: sets an allready crypted root password

5.50.57 CLCFG_setAuthorized_keys

Description: sets the ssh authorized_file for remote login into the clients
Parameter:

- serverIP: IP of the server
- pathToKeyFile: path on the server where to get the key file from
- acceptyDSSKeys: Set to true, if SSH-DSS keys should be re-enabled

5.50.58 CLCFG_copyClientPackageStatus

Description: copies the package installation status file from the lient to the cache directory of the server
Parameter:

- clientName: the name of the client

5.50.59 CLCFG_aptGet

Description: Executes an APT command with parameters and tries to install/remove as many packages as possible. If the packages could not get installed in a bunch try to install the packges one by one.
Parameter:

- command: install or remove
- packages: white space seperated list of packages to install or remove

5.50.60 CLCFG_importLocalPoolKey

Description: Generates BASH code to import the local package pool key (if it exists) on the client's APT system.
Returns:

- BASH code to import the local package pool key or empty string, if there is no package pool sign key.

5.50.61 CLCFG_installBasePackages

Description: installs needed base packages
Parameter:

- packagelist: the list of the packages to install
- keyring: name of the keyring package

5.50.62 CLCFG_setDebconf

Description: installs the debconf packages and sets debconf configuration
Parameter:

- serverIP: IP of the server, where to download the config file
- debconfFile: path and filename of the debconf file

5.50.63 CLCFG_debootstrap

Description: bootstraps a Debian system
Parameter:

- suite: select the Debian suite (sarge, sid, woody, potato)
- DNSServers: DNS server for resolving the names of the installation server
- gateway: gateway for fetching the packages
- packageProxy: the ip of the proxy the packages should be fetched from
- packagePort: the proxy port
- mirror: the Debian mirror the packages should be fetched from
- arch: the computer architecture of the client
- exclude: packages to exclude (as comma-seperated list)
- include: additional packages to include
- isCritical: selects if debootstrap errors should be critical
- additionalPackages: space seperated list of additional packages that should be installed during bootstrapping

5.50.64 CLCFG_downgradeExt

Description: downgrades all ext* partitions so the fsck.ext* of woody can understand it

5.50.65 CLCFG_mountRootDir

Description: create a new directory for mounting the root partition of the client, mount the installation directory to this mount point and create a temp directory
Parameter:

- rootDev: the root device (e.g. /dev/hda1)
- mountPoint: directory under /mnt to mount the device in
- CFDiskIOO: Client object (used for mounting the EFI boot partition).

5.50.66 CLCFG_activateDMA

Description: try to activate DMA to speed up installation. this short script tries to detect

5.50.67 CLCFG_fetchm23BasicTools

Description: fetches the basic m23 tools and fsize

5.50.68 CLCFG_dialogGaugeProcPos

Description: Generates the BASH code for showing a dialog with status bar that uses /proc/pid/fdinfo for current file position
Parameter:

- backtitle: title on top of the screen
- title: message title
- message: message to show over the status bar
- infofilecmd: BASH code to figure out the needed status file in the proc filesystem
- fullsize: The full size of the file that should be written.
- force: Forces showing of the dialog box

5.50.69 CLCFG_dialogInfoBox

Description: generates the BASH code for showing an dialog infobox
Parameter:

- backtitle: title on top of the screen
- title: message title
- message: the message itself
- force forces showing of the dialog box

5.50.70 CLCFG_dialogMsgBox

Description: generates the BASH code for showing an dialog message box
Parameter:

- backtitle: title on top of the screen
- title: message title
- message: the message itself
- force forces showing of the dialog box

5.50.71 CLCFG_dialogAllBox

Description: generates the BASH code for showing different types of dialog boxes
Parameter:

- backtitle: title on top of the screen
- title: message title
- message: the message itself
- type: type of the dialog box

5.50.72 CLCFG_executeAfterChroot

Description: prepares for pivot_root and does it and executes afterChroot

5.50.73 CLCFG_writeCrontabm23fetchjobEvery5Minutes

Description: Adds entries to crontab to check every 5 minutes for new jobs.

5.50.74 CLCFG_writeM23fetchjob

Description: generates the m23fetchjob script
Parameter:

- release: Name of the distribution release for special handling of some releases.

5.50.75 CLCFG_hideKernelWarnings

Description: hides the kernel warnungs

5.50.76 CLCFG_getRootDeviceFS

Description: Gets the filesystem of the root device.
Parameter:

- rootDevice: the device, the kernel should be installed on
- clientName: the name of the client

Returns:

- Filesystem of the root device.

5.50.77 CLCFG_genFakeFstab

Description: generates a fake /etc/fstab that only contains the lines for proc and the root partition (this is used to make the kernel install correctly)
Parameter:

- rootDevice: the device, the kernel should be installed on
- clientName: the name of the client

5.50.78 CLCFG_copySSLCert

Description: fetches the SSL certificate from the server and copies it to the client
Parameter:

- rootPath: the path to where the root directory is mounted
- disableSSLCertCheck: Disables the SSL certificate check of wget.

5.50.79 CLCFG_changeUser

Description: changes the settings of an useraccount on a client
Parameter:

- userName: the (old) username for the account
- password: the new unecrypted password for the account
- newUserName: the new username

5.50.80 CLCFG_patchNsswitchForLDAP

Description: Patches /etc/nsswitch.conf for usage with LDAP.

5.50.81 CLCFG_enableNFSHome

Description: enables storing of home directories on a NFS server
Parameter:

- nfsURL: URL to the NFS storage (e.g. 192.168.1.23:/nfs-homes)

5.50.82 CLCFG_disableNFSHome

Description: Disables storing of home directories on a NFS server

5.50.83 CLCFG_installDesktopLanguagePackage

Description: Installs some additional language packages for (KDE / Gnome) desktops.
Parameter:

- lang: short language
- kde: Install KDE language packages too.
- gnome: Install Gnome language packages too.

5.50.84 CLCFG_installApplicationLanguagePackages

Description: Installs some additional language packages for installed applications with seperate language packs.
Parameter:

- lang: short language

5.50.85 CLCFG_updateDebootstrapScripts

Description: Updates the debootstrap scripts for Debian or Ubuntu and returns the www path to the files

5.50.86 CLCFG_updateDebootstrapScriptsDebian

Description: Updates the debootstrap scripts for Debian and returns the www path to the files

5.51 ./inc/distr/imaging/clientConfig.php

Author: Hauke Goos-Habermann (HHabermann@pc-kiel.de)
Description: Imaging specific management functions.

5.51.1 CLCFG_addDistributionSpecificOptions

Description: adds distribution specific settings to an option array and returns the new array
Parameter:

- $options: the options array with some values

5.51.2 CLCFG_showDistributionSpecificOptions

Description: shows distribution specific options and returns false, if there was an error
Parameter:

- options: options array
- client: Name of the client.

5.52 ./inc/distr/imaging/packages.php

Author: Hauke Goos-Habermann (HHabermann@pc-kiel.de)
Description: Debian/Ubuntu specific package handling functions.

5.52.1 PKG_updatePackageSearchCacheFile

Description: Updates the compressed package search file.
Parameter:

- packagesource: Name of the package sources list.

Returns:

- Full path to the log file.

5.52.2 PKG_getSearchCacheFileName

Description: Returns the full path to the compressed package search file.
Parameter:

- dir: Directory where the package information is stored.
- arch: Architecture for the packages in the search file.

Returns:

- Full path to the compressed package search file.

5.52.3 PKG_genPackageSearchCacheFileCMD

Description: Generates commands to create a compressed package search file.
Parameter:

- dir: Directory where the package information is stored.
- arch: Architecture for the packages in the search file.

Returns:

- BASH commands for creating a compressed package search file.

5.52.4 PKG_searchPackageSearchCacheFileCMD

Description: Generates BASH commands to search a compressed package search file.
Parameter:

- dir: Directory where the package information is stored.
- arch: Architecture for the packages in the search file.
- search: Search terms.

Returns:

- BASH commands to search a compressed package search file.

5.52.5 PKG_ncTarDebsFromClientToServer_Client

Description: Client to send the Debian packages to the m23 client.

5.52.6 PKG_ncTarDebsFromClientToServer_Server

Description: Starts a server to receive the Debian packages from an m23 client.
Parameter:

- poolDir: The directory on the server to store the Debian packages.

5.52.7 PKG_rsyncDebsFromClientToServer

Description: Copies all the Debian packages from a client to a given directory on the server via rsync.
Parameter:

- clientIP: The IP of the client.
- poolDir: The directory on the server to store the Debian packages.

5.52.8 PKG_preparePool

Description: Generates the needed configuration file for reprepro.
Parameter:

- release: Release of the pool.
- distr: Distribution of the pool.
- arch: Architecture of the pool (POOL_ARCH_I386 or CPoolLister::POOL_ARCH_AMD64).
- poolName: Name of the pool (if a pool with the given name exists => load)
- poolDir: Directory of the pool with full path.

5.52.9 PKG_convertPackagesToRepository

Description: Generates commands for creating a package source from packages stored in one directory.
Parameter:

- poolDir: Complete path to the directory that stores the pool.
- logFile: File (with full path) to store the pool generation log in.
- poolName: Name of the pool.
- sourceslist: Variable to store the sources list in.

Returns:

- Commands for creating a package source from packages stored in one directory.

5.52.10 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.

5.52.11 PKG_fastGetNewInstalledPackages

Description: Gets new installed packages by comparing status files (before and after the installation).
Parameter:

- oldStatusFile: File with the list of all installed packages before the installation of the new packages
- newStatusFile: File with the list of all installed packages after the installation of the new packages
- storeFile: File name to store the list of new installed on the client.

5.52.12 PKG_searchFor

Description: searches for a package and returnes a file descriptor
Parameter:

- key: search key
- distr: the distribution name
- packagesource: name of the package source
- arch: Array of architectures to get package infos for.

5.52.13 PKG_getDescription

Description: gets the description of a package
Parameter:

- line: line containing package name and description

5.52.14 PKG_getPackageName

Description: gets the name of a package
Parameter:

- line: line containing package name and description

5.52.15 PKG_getLongPackageDescription

Description: Gets the description of a package.
Parameter:

- package: Name of the package.
- distr: the distribution name
- packagesource: name of the package source

Returns:

- Description of the package.

5.52.16 PKG_getPackageDescriptionSize

Description: Returnes the description and size of a package
Parameter:

- distr: the name of the distribution
- packagesource: the name of the package source list
- package: name if the package

5.52.17 PKG_printStatus

Description: prints the package status, e.g. amount of packages,...
Parameter:

- distr: the name of the distribution
- packagesource: the name of the package source list

5.52.18 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)
- completeDescription: If set to true, the full package description and sizes are fetched (time consuming)

Returns:

- : Amount of found packages.

5.52.19 PKG_getAptArchOptions

Description: Generates options to specify the architecture of a client that can be appended to an apt-get line.
Parameter:

- arch: Architecture to get package infos for.

Returns:

- : Parameter for architecture specific apt-get commands, if architecture is not i386.

5.52.20 PKG_addAPTConfigFiles

Description: Creates the config files for the package manager on the m23 server.
Parameter:

- sourceName: The name of the package source list
- dir: The directory on the m23 server where the config files should be created.

5.52.21 PKG_preparePackageDir

Description: creates the needed files + sources list in a directory to use it for "local apt".
Parameter:

- dir: the directory to prepare
- packagesource: sources list
- logFile: Name of the file the messages from apt-get should be written to.
- returnCmd: Set to true, if the apt command should be returned or to false to execute it in this function.
- arch: Architecture to get package infos for.
- sourceName: The name of the package source list
- hardTryUpdate: Set to true, if the apt-get update should be run until it succeeds

Returns:

- : Error text on error or empty string on success.

5.52.22 PKG_updatePackageInfo

Description: Updates the package information to make it searchable.
Parameter:

- distr: the distribution name
- packagesource: name of the package source
- force: set to true if you want to update the package info and the time is not over
- arch: Architecture to get package infos for.
- changeTime: Writes the time when the status file was last modified to.

5.52.23 PKG_previewInstall

Description: shows what happens if packages get 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

5.52.24 PKG_getKernels

Description: Generates an associative array with the available kernels for an architecture and distribution as keys and values, where kernels containing the term 'generic' are at the front.
Parameter:

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

Returns:

- Associative array with the available kernels for an architecture and distribution as keys and values with kernels containing the term 'generic' at the front, sorted by length and then by alphabet.

5.52.25 PKG_downloadPool

Description: Generates commands to download packages from a sources list to a directory.
Parameter:

- destDir: the directory the packages should be downloaded to
- sourceslist: package source list
- packagesArr: array of the lists that contain packages to download (seperated by blanks)
- arch: CPU architecture to download the packages for.
- release: Select the Debian/Ubuntu suite (squeeze, sarge, sid, precise).

5.52.26 PKG_getDebootStrapBasePackages

Description: Returns the list of base packages that are downloaded by debootstrap for a special release.
Parameter:

- release: release name of the distribution version (e.g. sarge)

5.53 ./inc/distr.php

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

5.53.1 DISTR_getUbuntuUserGroups

Description: Returns the default groups an user of a Ubuntu system should be in.
Parameter:

- arrayOrSeparator: If set to true, the groups are returned as array, otherwise the value is used as separator character.

Returns:

- Default groups as array or as string, where the groups are separates by a separator character.

5.53.2 DISTR_getDebianUserGroups

Description: Returns the default groups an user of a Debian system should be in.
Parameter:

- arrayOrSeparator: If set to true, the groups are returned as array, otherwise the value is used as separator character.

Returns:

- Default groups as array or as string, where the groups are separates by a separator character.

5.53.3 DISTR_releaseVersionTranslator

Description: Adds the version number to a Debian or Ubuntu release.
Parameter:

- release: Release name (e.g. etch)

Returns:

- Combination of release code name and version number.

5.53.4 DISTR_listDistributions

Description: returns a list of the directory names of the distributions
Parameter:

- addEmpty: set to true if an empty fake distribution shoul be put on top of the list

Returns:

- An array that contains the system names of all distributions.

5.53.5 DISTR_getDescriptionValues

Description: gets the valuest stored in the info.txt file of the distributions and returns a associative array
Parameter:

- shortName: the short name of the distribution

Returns:

- : Associative array with information about the distribution.

5.53.6 DISTR_DistributionsSelections

Description: generates a selection/option form of the available distributions
Parameter:

- selName: name of the selection
- first: shortName of the distribution to show first

5.53.7 DISTR_geti18nValue

Description: returns a value of the info file and tries to get it in the selected language. if it doesn't exist it is returned in the default language
Parameter:

- lang: language as 2 letter code (e.g. de)
- variable: the name of the variable you want to get (e.g. Description)
- values: the associate array of the info.txt values

5.53.8 DISTR_listCommaSeperated

Description: returns a normal array with the values of the specified variable (e.g. var: GUIs => result: [0] => Textmode [1] => KDE3 [2] => KDEwoody ...)
Parameter:

- variable: the variable to search for
- values: the values of the distribution text file (info.txt) read with DISTR_getDescriptionValues

5.53.9 DISTR_commaSeperatedSelections

Description: returns a selection with certain values (specified thru $variable) from the distribution text file
Parameter:

- selName: name of the selection
- first: item to show first
- variable: the variable to search for
- values: the values of the distribution text file (info.txt) read with DISTR_getDescriptionValues

5.53.10 DISTR_getDesktopsCBList

Description: returns a checkbox list with desktops for a certain distribution. Desktops included in the array $selectedDesktops are checked.
Parameter:

- distr: distribution of the desktops should be shown
- selectedDesktops: array with the desktops that should be checked

5.53.11 DISTR_getDesktopDescription

Description: Returns the description for the given desktop in the given distribution preferedly in the language of the m23 webinterface.
Parameter:

- distr: Name of the distribution.
- desktop: Name of the desktop.

5.53.12 DISTR_getSelectedDesktopsArr

Description: returns an array with selected desktops from the list generated by DISTR_getDesktopsCBList

5.53.13 DISTR_getSelectedDesktopsStr

Description: returns a string with selected desktops (seperated by "###") from the list generated by DISTR_getDesktopsCBList

5.54 ./inc/distr/ubuntu/clientConfigCommon.php

Author: Hauke Goos-Habermann (HHabermann@pc-kiel.de)
Description: Common management funtions shared by Linux distributions.

5.54.1 CLCFG_enableFreeIPA

Description: Shows BASH commands to enable FreeIPA on a client.
Parameter:

- clientName: Name of the client.

Returns:

- true, if the BASH commands could be prepared, false otherwise.

5.54.2 CLCFG_enableLDAPplusKerberos

Description: Enables login via Kerberos for a client.
Parameter:

- clientOptions: the client's options array
- clientOptions[KRB_server]: IP or hostname of the Kerberos server.
- clientOptions[KRB_adminServer]: IP or hostname of the Kerberos admin server.
- clientOptions[KRB_realm]: Kerberos realm.

5.54.3 CLCFG_enablePAMModule

Description: Enables a PAM modules (if it wasn't enabled before)

5.54.4 CLCFG_enableShutdownByUsersGroup

Description: Allows all users of the group users to run shutdown as root.

5.54.5 CLCFG_enableSudoByM23sudoGroup

Description: Allows all users of the group m23sudo to run command as root.

5.54.6 CLCFG_blacklistPackageInstallation

Description: Blacklists a package for APT to make it uninstallable.
Parameter:

- package: Name of the package.

5.54.7 CLCFG_executeNextWorkEveryMinute

Description: Runs executeNextWork in a loop every minute and shows a status message about elapsed waiting time.
Parameter:

- lang: language for the messages

5.54.8 CLCFG_setSystime

Description: Sets the system time of the client to about the same time of the m23 server.

5.54.9 CLCFG_disableAptSystemdDaily

Description: Deactivates systemd apt-daily.timer.

5.54.10 CLCFG_addPAMtoDM

Description: Adds PAM modules (if present) to the pam.d configuration files.
Parameter:

- dmFile: Name of the login manager pam.d config file.

5.54.11 CLCFG_disableSudoRootLogin

Description: Disables getting root rights of normal users via sudo.

5.54.12 CLCFG_installFirmware

Description: Installs available firmware packages.

5.54.13 CLCFG_monoRemove

Description: Removes mono packages and installs gnote when tomboy was installed before.

5.54.14 CLCFG_setDebConfDM

Description: Sets the display manager in debconf.
Parameter:

- dm: Name of the display manager (kdm, lightdm, mdm, ...).

5.54.15 CLCFG_installXDM

Description: Installs the XDM display manager.

5.54.16 CLCFG_installSDDM

Description: Installs the SDDM display manager.

5.54.17 CLCFG_installMintDM

Description: Installs the Linux Mint DM display manager.

5.54.18 CLCFG_copyMBRToAllDevices

Description: Copies the MBR code from the boot device to all other devices.
Parameter:

- bootDevice: Device name of the device with the original MBR (e.g. /dev/sda).

5.54.19 CLCFG_activateBOOT_DEGRADED

Description: Activates BOOT_DEGRADED on Ubuntu 12.04 to allow booting from degraded RAIDs.

5.54.20 CLCFG_installLightDM

Description: Installs the light DM display manager.
Parameter:

- session: Name of the session to select by default.
- addSessionWrapper: Set to true, if an additional line with "session-wrapper=/etc/X11/Xsession" should be added.

5.54.21 CLCFG_setDebConfDirect

Description: Sets debconf settings.
Parameter:

- debconf: Debconf settings to add.

5.54.22 TRINITY_installLoginManager

Description: Installs the Trinity login manager KDM.
Parameter:

- lang: short language

5.54.23 TRINITY_install

Description: Installs a minimalistic Trinity desktop.
Parameter:

- lang: short language

5.54.24 CLCFG_makeDev

Description: Creates the device nodes in /dev by downloading and extracting an archive containing the device nodes and if this fails running MAKEDEV.

5.54.25 CLCFG_disablePlymouth

Description: Disables the plymouth.

5.54.26 CLCFG_disableAvahiDaemon

Description: Disables the avahi-daemon.

5.54.27 CLCFG_configUpstartForNormalUsage

Description: Configures upstart for normal running in an installed system.

5.54.28 CLCFG_configUpstartForChroot

Description: Configures upstart to make it not fail installation.

5.54.29 CLCFG_createScreenRC

Description: Creates the (under Ubuntu) needed settings for screen.

5.54.30 CLCFG_addGrubPassword

Description: Adds a password line to the grub configuration to lock the edit line of bootmanager.

5.54.31 CLCFG_addLiloPassword

Description: Adds a password line to the LiLo configuration to lock the edit line of bootmanager.

5.54.32 GNOME_prepare

Description: Prepares the GNOME installation

5.54.33 LXDE_install

Description: Installs the LXDE desktop.
Parameter:

- lang: short language
- fullInstall: Set to true, if the full desktop with all applications should be installed. Otherwise a minimal desktop will be installed

5.54.34 GNOME3_install

Description: Installs the GNOME 3 desktop.
Parameter:

- lang: short language
- fullInstall: Set to true, if the full desktop with all applications should be installed. Otherwise a minimal desktop will be installed

5.54.35 GNOME_install

Description: Installs a minimalistic GNOME desktop.
Parameter:

- lang: short language

5.54.36 GNOME_installLoginManager

Description: Installs the GNOME login manager GDM.
Parameter:

- lang: short language

5.54.37 KDE_prepare

Description: Prepares the KDE installation

5.54.38 KDE_install

Description: Installs a minimalistic KDE desktop.
Parameter:

- lang: short language
- ver: KDE version 3 or 4 (minor releases are depending in the on the used distribution)

5.54.39 KDE_installLoginManager

Description: Installs the KDE/Trinity login manager KDM.
Parameter:

- lang: short language
- ver: KDE version 3 or 4 (minor releases are depending in the on the used distribution)
- trinity: Set to true, if the Trinity Desktop should be configured.

5.54.40 CLCFG_lilo2Grub

Description: Installs and runs the lilo.conf to Grub's menu.lst converter

5.54.41 CLCFG_getMbrPart

Description: Returns the device to install the MBR of the bootloader in.
Parameter:

- installPart: Partition to install the OS on.
- clientOptions: Array with information about client options.

Returns:

- Device to install the MBR of the bootloader in.

5.54.42 CLCFG_setTimeZone

Description: Sets the timezone a a client.
Parameter:

- timezone: POSIX timezone as defined at /usr/share/zoneinfo/posix/

5.54.43 CLCFG_writeHosts

Description: writes the /etc/hosts file for the client

5.54.44 CLCFG_addUser

Description: generates the commands to add a user on the client. it adds the user account, creates a home directory, copies the m23 skel files and sets the
Parameter:

- userName: the username for the account
- password: the unecrypted password for the account
- groups: the groups the user should be added to
- skelDir: directory to the skeleton files
- uid: Optional user ID of the new user.
- gid: Optional group ID of the new user.

5.54.45 CLCFG_createBootDeviceNode

Description: Makes sure the needed device nodes for the boot device are created.

5.54.46 CLCFG_efi

Description: Configures the client for UEFI booting.
Parameter:

- CFDiskIOO: Object to with partition information for the calling client.

Returns:

- true, if the client uses EFI, otherwise false.

5.54.47 CLCFG_genFstab

Description: generates the commands to auto detect the partitions and generate the fstab file and installs a boot manager
Parameter:

- bootDevice: the device the bootloader should be installed on (e.g. /dev/hda)
- rootDevice: the path to the installation partition (e.g. /dev/hda1)
- bootloader: Name of the boot loader to install and configure (actually grub only)
- ignoreErrors: Set to true, if some errors should be ignored instead of logging them.
- isReconfiguredWithExtraDistr: Set to true, the function is called to reconfigure a distribution that was installed by imaging.

5.54.48 CLCFG_interfaces

Description: generates a script that writes the /etc/network/interfaces file.
Parameter:

- clientParams: Associated array with the parameters of the client.

5.54.49 CLCFG_hostname

Description: generates a script that writes the /etc/hostname file.
Parameter:

- clientName: the name of the client

5.54.50 CLCFG_resolvConf

Description: generates a script that writes the /etc/resolv.conf file.
Parameter:

- DNSServers: list of DNS servers

5.54.51 CLCFG_aptConf

Description: generates a script that writes the /etc/apt/apt.conf.d/70debconf file.
Parameter:

- proxyServer: IP or name of the proxy server
- proxyPort: port the proxy server listens on

5.54.52 CLCFG_sourceslist

Description: generates a script that writes the /etc/apt/sources.list file.
Parameter:

- clientIP: IP of the client
- clientName: name of the client
- serverIP: IP of the server

5.54.53 CLCFG_sourceslistCreateConfigFiles

Description: Creates config files for the package manager.
Parameter:

- sourceName: The name of the package source list

5.54.54 CLCFG_hwdetect

Description: updates and runs the hardware detection

5.54.55 CLCFG_language

Description: sets the language for keyboard in console and X11 and console language
Parameter:

- lang: 2 letter language code (de,fr,it,en)
- release: The release name of the distribution (for special handling).

5.54.56 CLCFG_setRootPassword

Description: sets the root password of the client
Parameter:

- cryptedPassword: sets an allready crypted root password

5.54.57 CLCFG_setAuthorized_keys

Description: sets the ssh authorized_file for remote login into the clients
Parameter:

- serverIP: IP of the server
- pathToKeyFile: path on the server where to get the key file from
- acceptyDSSKeys: Set to true, if SSH-DSS keys should be re-enabled

5.54.58 CLCFG_copyClientPackageStatus

Description: copies the package installation status file from the lient to the cache directory of the server
Parameter:

- clientName: the name of the client

5.54.59 CLCFG_aptGet

Description: Executes an APT command with parameters and tries to install/remove as many packages as possible. If the packages could not get installed in a bunch try to install the packges one by one.
Parameter:

- command: install or remove
- packages: white space seperated list of packages to install or remove

5.54.60 CLCFG_importLocalPoolKey

Description: Generates BASH code to import the local package pool key (if it exists) on the client's APT system.
Returns:

- BASH code to import the local package pool key or empty string, if there is no package pool sign key.

5.54.61 CLCFG_installBasePackages

Description: installs needed base packages
Parameter:

- packagelist: the list of the packages to install
- keyring: name of the keyring package

5.54.62 CLCFG_setDebconf

Description: installs the debconf packages and sets debconf configuration
Parameter:

- serverIP: IP of the server, where to download the config file
- debconfFile: path and filename of the debconf file

5.54.63 CLCFG_debootstrap

Description: bootstraps a Debian system
Parameter:

- suite: select the Debian suite (sarge, sid, woody, potato)
- DNSServers: DNS server for resolving the names of the installation server
- gateway: gateway for fetching the packages
- packageProxy: the ip of the proxy the packages should be fetched from
- packagePort: the proxy port
- mirror: the Debian mirror the packages should be fetched from
- arch: the computer architecture of the client
- exclude: packages to exclude (as comma-seperated list)
- include: additional packages to include
- isCritical: selects if debootstrap errors should be critical
- additionalPackages: space seperated list of additional packages that should be installed during bootstrapping

5.54.64 CLCFG_downgradeExt

Description: downgrades all ext* partitions so the fsck.ext* of woody can understand it

5.54.65 CLCFG_mountRootDir

Description: create a new directory for mounting the root partition of the client, mount the installation directory to this mount point and create a temp directory
Parameter:

- rootDev: the root device (e.g. /dev/hda1)
- mountPoint: directory under /mnt to mount the device in
- CFDiskIOO: Client object (used for mounting the EFI boot partition).

5.54.66 CLCFG_activateDMA

Description: try to activate DMA to speed up installation. this short script tries to detect

5.54.67 CLCFG_fetchm23BasicTools

Description: fetches the basic m23 tools and fsize

5.54.68 CLCFG_dialogGaugeProcPos

Description: Generates the BASH code for showing a dialog with status bar that uses /proc/pid/fdinfo for current file position
Parameter:

- backtitle: title on top of the screen
- title: message title
- message: message to show over the status bar
- infofilecmd: BASH code to figure out the needed status file in the proc filesystem
- fullsize: The full size of the file that should be written.
- force: Forces showing of the dialog box

5.54.69 CLCFG_dialogInfoBox

Description: generates the BASH code for showing an dialog infobox
Parameter:

- backtitle: title on top of the screen
- title: message title
- message: the message itself
- force forces showing of the dialog box

5.54.70 CLCFG_dialogMsgBox

Description: generates the BASH code for showing an dialog message box
Parameter:

- backtitle: title on top of the screen
- title: message title
- message: the message itself
- force forces showing of the dialog box

5.54.71 CLCFG_dialogAllBox

Description: generates the BASH code for showing different types of dialog boxes
Parameter:

- backtitle: title on top of the screen
- title: message title
- message: the message itself
- type: type of the dialog box

5.54.72 CLCFG_executeAfterChroot

Description: prepares for pivot_root and does it and executes afterChroot

5.54.73 CLCFG_writeCrontabm23fetchjobEvery5Minutes

Description: Adds entries to crontab to check every 5 minutes for new jobs.

5.54.74 CLCFG_writeM23fetchjob

Description: generates the m23fetchjob script
Parameter:

- release: Name of the distribution release for special handling of some releases.

5.54.75 CLCFG_hideKernelWarnings

Description: hides the kernel warnungs

5.54.76 CLCFG_getRootDeviceFS

Description: Gets the filesystem of the root device.
Parameter:

- rootDevice: the device, the kernel should be installed on
- clientName: the name of the client

Returns:

- Filesystem of the root device.

5.54.77 CLCFG_genFakeFstab

Description: generates a fake /etc/fstab that only contains the lines for proc and the root partition (this is used to make the kernel install correctly)
Parameter:

- rootDevice: the device, the kernel should be installed on
- clientName: the name of the client

5.54.78 CLCFG_copySSLCert

Description: fetches the SSL certificate from the server and copies it to the client
Parameter:

- rootPath: the path to where the root directory is mounted
- disableSSLCertCheck: Disables the SSL certificate check of wget.

5.54.79 CLCFG_changeUser

Description: changes the settings of an useraccount on a client
Parameter:

- userName: the (old) username for the account
- password: the new unecrypted password for the account
- newUserName: the new username

5.54.80 CLCFG_patchNsswitchForLDAP

Description: Patches /etc/nsswitch.conf for usage with LDAP.

5.54.81 CLCFG_enableNFSHome

Description: enables storing of home directories on a NFS server
Parameter:

- nfsURL: URL to the NFS storage (e.g. 192.168.1.23:/nfs-homes)

5.54.82 CLCFG_disableNFSHome

Description: Disables storing of home directories on a NFS server

5.54.83 CLCFG_installDesktopLanguagePackage

Description: Installs some additional language packages for (KDE / Gnome) desktops.
Parameter:

- lang: short language
- kde: Install KDE language packages too.
- gnome: Install Gnome language packages too.

5.54.84 CLCFG_installApplicationLanguagePackages

Description: Installs some additional language packages for installed applications with seperate language packs.
Parameter:

- lang: short language

5.54.85 CLCFG_updateDebootstrapScripts

Description: Updates the debootstrap scripts for Debian or Ubuntu and returns the www path to the files

5.54.86 CLCFG_updateDebootstrapScriptsDebian

Description: Updates the debootstrap scripts for Debian and returns the www path to the files

5.55 ./inc/distr/ubuntu/clientConfig.php

Author: Hauke Goos-Habermann (HHabermann@pc-kiel.de)
Description: Ubuntu specific management functions.

5.55.1 CLCFG_showDistributionSpecificOptionsUbuntu

Description: Shows additional distribution specific options for Ubuntu.
Parameter:

- options: options array

Returns:

- Options array or false, if there was an error

5.55.2 UBUNTU_installLanguagePacks

Description: Installs the language packs on Ubuntu.
Parameter:

- lang: short language

5.55.3 UBUNTU_desktopInstall

Description: Installs a desktop environment.
Parameter:

- desktop: Desktop constant.
- globalMenu: If set to true Unity's global menus are activated otherwise deactivated.
- normalButtonPosition: If set to true the window buttons are shown on the top right instead of top left.
- normalScrollBars: If set to true the normal scroll bars are usesed instead of the Unity bars.
- addDesktopIcons: If set to true the home, network, volumes and trash icons are shown on the desktop.
- removeUbuntuOne: If set to true Ubuntu One is removed.
- removeMono: If set to true Mono is removed.
- installLangPacks: If set to true language packs will be installed.

5.55.4 UBUNTU_fixBeforeBaseInstall

Description: Corrects the settings for Ubuntu before the base packages are installed.
Parameter:

- release: Name of the Ubuntu release.

5.55.5 UBUNTU_fixAfterBaseInstall

Description: Corrects the settings for Ubuntu after the base packages are installed.
Parameter:

- release: Name of the Ubuntu release.

5.55.6 CLCFG_listUbuntuReleases

Description: generates a selection of the different Ubuntu releases (breezy, hoary, warty)
Parameter:

- selName: the name of the selection
- first: the release to show first

5.55.7 CLCFG_updateDebootstrapScriptsUbuntu

Description: Updates the debootstrap scripts for Ubuntu and returns the www path to the files

5.55.8 CLCFG_enableKerberosWithHostTickets

Description: Enables Kerberos host/user login for a client.
Parameter:

- clientOptions: the client's options array

5.55.9 CLIENT_enableLDAP

Description: enables LDAP login for a client.
Parameter:

- clientOptions: the client's options array

5.56 ./inc/distr/ubuntu/packages.php

Author: Hauke Goos-Habermann (HHabermann@pc-kiel.de)
Description: Debian/Ubuntu specific package handling functions.

5.56.1 PKG_updatePackageSearchCacheFile

Description: Updates the compressed package search file.
Parameter:

- packagesource: Name of the package sources list.

Returns:

- Full path to the log file.

5.56.2 PKG_getSearchCacheFileName

Description: Returns the full path to the compressed package search file.
Parameter:

- dir: Directory where the package information is stored.
- arch: Architecture for the packages in the search file.

Returns:

- Full path to the compressed package search file.

5.56.3 PKG_genPackageSearchCacheFileCMD

Description: Generates commands to create a compressed package search file.
Parameter:

- dir: Directory where the package information is stored.
- arch: Architecture for the packages in the search file.

Returns:

- BASH commands for creating a compressed package search file.

5.56.4 PKG_searchPackageSearchCacheFileCMD

Description: Generates BASH commands to search a compressed package search file.
Parameter:

- dir: Directory where the package information is stored.
- arch: Architecture for the packages in the search file.
- search: Search terms.

Returns:

- BASH commands to search a compressed package search file.

5.56.5 PKG_ncTarDebsFromClientToServer_Client

Description: Client to send the Debian packages to the m23 client.

5.56.6 PKG_ncTarDebsFromClientToServer_Server

Description: Starts a server to receive the Debian packages from an m23 client.
Parameter:

- poolDir: The directory on the server to store the Debian packages.

5.56.7 PKG_rsyncDebsFromClientToServer

Description: Copies all the Debian packages from a client to a given directory on the server via rsync.
Parameter:

- clientIP: The IP of the client.
- poolDir: The directory on the server to store the Debian packages.

5.56.8 PKG_preparePool

Description: Generates the needed configuration file for reprepro.
Parameter:

- release: Release of the pool.
- distr: Distribution of the pool.
- arch: Architecture of the pool (POOL_ARCH_I386 or CPoolLister::POOL_ARCH_AMD64).
- poolName: Name of the pool (if a pool with the given name exists => load)
- poolDir: Directory of the pool with full path.

5.56.9 PKG_convertPackagesToRepository

Description: Generates commands for creating a package source from packages stored in one directory.
Parameter:

- poolDir: Complete path to the directory that stores the pool.
- logFile: File (with full path) to store the pool generation log in.
- poolName: Name of the pool.
- sourceslist: Variable to store the sources list in.

Returns:

- Commands for creating a package source from packages stored in one directory.

5.56.10 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.

5.56.11 PKG_fastGetNewInstalledPackages

Description: Gets new installed packages by comparing status files (before and after the installation).
Parameter:

- oldStatusFile: File with the list of all installed packages before the installation of the new packages
- newStatusFile: File with the list of all installed packages after the installation of the new packages
- storeFile: File name to store the list of new installed on the client.

5.56.12 PKG_searchFor

Description: searches for a package and returnes a file descriptor
Parameter:

- key: search key
- distr: the distribution name
- packagesource: name of the package source
- arch: Array of architectures to get package infos for.

5.56.13 PKG_getDescription

Description: gets the description of a package
Parameter:

- line: line containing package name and description

5.56.14 PKG_getPackageName

Description: gets the name of a package
Parameter:

- line: line containing package name and description

5.56.15 PKG_getLongPackageDescription

Description: Gets the description of a package.
Parameter:

- package: Name of the package.
- distr: the distribution name
- packagesource: name of the package source

Returns:

- Description of the package.

5.56.16 PKG_getPackageDescriptionSize

Description: Returnes the description and size of a package
Parameter:

- distr: the name of the distribution
- packagesource: the name of the package source list
- package: name if the package

5.56.17 PKG_printStatus

Description: prints the package status, e.g. amount of packages,...
Parameter:

- distr: the name of the distribution
- packagesource: the name of the package source list

5.56.18 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)
- completeDescription: If set to true, the full package description and sizes are fetched (time consuming)

Returns:

- : Amount of found packages.

5.56.19 PKG_getAptArchOptions

Description: Generates options to specify the architecture of a client that can be appended to an apt-get line.
Parameter:

- arch: Architecture to get package infos for.

Returns:

- : Parameter for architecture specific apt-get commands, if architecture is not i386.

5.56.20 PKG_addAPTConfigFiles

Description: Creates the config files for the package manager on the m23 server.
Parameter:

- sourceName: The name of the package source list
- dir: The directory on the m23 server where the config files should be created.

5.56.21 PKG_preparePackageDir

Description: creates the needed files + sources list in a directory to use it for "local apt".
Parameter:

- dir: the directory to prepare
- packagesource: sources list
- logFile: Name of the file the messages from apt-get should be written to.
- returnCmd: Set to true, if the apt command should be returned or to false to execute it in this function.
- arch: Architecture to get package infos for.
- sourceName: The name of the package source list
- hardTryUpdate: Set to true, if the apt-get update should be run until it succeeds

Returns:

- : Error text on error or empty string on success.

5.56.22 PKG_updatePackageInfo

Description: Updates the package information to make it searchable.
Parameter:

- distr: the distribution name
- packagesource: name of the package source
- force: set to true if you want to update the package info and the time is not over
- arch: Architecture to get package infos for.
- changeTime: Writes the time when the status file was last modified to.

5.56.23 PKG_previewInstall

Description: shows what happens if packages get 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

5.56.24 PKG_getKernels

Description: Generates an associative array with the available kernels for an architecture and distribution as keys and values, where kernels containing the term 'generic' are at the front.
Parameter:

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

Returns:

- Associative array with the available kernels for an architecture and distribution as keys and values with kernels containing the term 'generic' at the front, sorted by length and then by alphabet.

5.56.25 PKG_downloadPool

Description: Generates commands to download packages from a sources list to a directory.
Parameter:

- destDir: the directory the packages should be downloaded to
- sourceslist: package source list
- packagesArr: array of the lists that contain packages to download (seperated by blanks)
- arch: CPU architecture to download the packages for.
- release: Select the Debian/Ubuntu suite (squeeze, sarge, sid, precise).

5.56.26 PKG_getDebootStrapBasePackages

Description: Returns the list of base packages that are downloaded by debootstrap for a special release.
Parameter:

- release: release name of the distribution version (e.g. sarge)

5.57 ./inc/edit.php

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

5.57.1 EDIT_sedSearchAddInsert

Description: Generates a sed command line to insert text before (SED_insertBefore) or after (SED_insertAfter) a searched line.
Parameter:

- file: Name of the file to edit.
- search: Search pattern to find the line for inserting before or after.
- text: The text to insert.
- insertBeforeOrAfter: Parameter to select, if the new text should inserted before (SED_insertBefore) or after (SED_insertAfter) the found line

Returns:

- sed code for inserting text before (SED_insertBefore) or after (SED_insertAfter) a searched line.

5.57.2 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.

5.57.3 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.

5.57.4 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

5.57.5 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)

5.57.6 EDIT_commentoutAll

Description: Comments out all matching lines.
Parameter:

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

5.57.7 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

5.57.8 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")

5.57.9 EDIT_uncomment

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

- file: the name of the file

5.57.10 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:

5.57.11 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.

5.57.12 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.

5.57.13 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.

5.57.14 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

5.57.15 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

5.57.16 EDIT_searchNextLineNumber

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

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

5.57.17 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.

5.57.18 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

5.57.19 EDIT_savePerms

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

- file: the name to the file

5.57.20 EDIT_restorePerms

Description: restores previously saved file permissions and owner

5.57.21 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

5.57.22 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

5.57.23 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

5.57.24 EDIT_deleteMatching

Description: Deletes all lines matching the regular expression
Parameter:

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

5.57.25 EDIT_replaceLineElseAdd

Description: Deletes all lines, that match regular expression, afterwards adds a given line
Parameter:

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

5.57.26 EDIT_appendToFile

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

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

Returns:

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

5.58 ./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.

5.58.1 FDISK_showFdiskCombinedGUIFunctions

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

5.58.2 FDISK_showCombinedFdiskGUIDialog

Description: Shows the new partition and formating screen.

5.58.3 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)).

5.58.4 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).

5.58.5 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.

5.58.6 FDISK_swapFilesystems

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

5.58.7 FDISK_formatInstallAndSwappart

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

5.58.8 FDISK_finalChecksAndRealPartitionAndFormatStart

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

5.58.9 FDISK_installFilesystems

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

5.58.10 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.

5.58.11 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.

5.58.12 FDISK_printAllBars2

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

5.58.13 FDISK_showAllPartTables

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

5.58.14 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.

5.58.15 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.

5.58.16 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.

5.58.17 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.

5.58.18 FDISK_fdiskSessionClient

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

- The client name to partition and format.

5.58.19 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.

5.58.20 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.

5.58.21 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.

5.58.22 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.

5.58.23 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).

5.58.24 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.

5.58.25 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.

5.58.26 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.

5.58.27 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.

5.58.28 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.

5.58.29 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.

5.58.30 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.

5.58.31 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.
- partitionsOnly: Set to true, if only partitions should be listed.

Returns:

- String with the HTML selection.

5.58.32 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

5.58.33 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

5.58.34 FDISK_getFirstDrive

Description: return the first drive as installation drive
Parameter:

- param: parameter string containing status informations about the harddisks

5.58.35 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

5.58.36 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"])

5.58.37 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

5.58.38 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.

5.58.39 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.

5.58.40 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

5.58.41 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

5.58.42 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.

5.58.43 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)

5.58.44 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

5.58.45 FDISK_installExistingDialog

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

- param: parameter string containing status informations about the harddisks

5.58.46 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)

5.58.47 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.

5.58.48 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.

5.58.49 FDISK_fstabAddDialog2

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

5.58.50 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.

5.58.51 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

5.58.52 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/)

5.58.53 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.

5.58.54 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.

5.58.55 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.

5.58.56 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.

5.58.57 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.

5.58.58 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.

5.58.59 FDISK_getAllDrives

Description: gets all drives of the client
Parameter:

- partitions: associative array containing status information about the harddisks

5.58.60 FDISK_colorFS

Description: get color for a selected filesystem
Parameter:

- fsName: name of the file system: ext3, ext2, linux-swap,...

5.58.61 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)

5.58.62 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

5.58.63 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

5.58.64 FDISK_getPartitions

Description: get the partition info for the client from db
Parameter:

- client: name of the client

5.58.65 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.

5.58.66 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.

5.58.67 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.

5.58.68 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.

5.58.69 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.

5.58.70 FDISK_getSupportedFS

Description: Generates and returns an array with the list of supported file systems.
Returns:

- Array with the list of supported file systems.

5.58.71 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.

5.58.72 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.

5.58.73 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

5.58.74 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

5.58.75 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

5.58.76 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

5.58.77 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

5.58.78 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

5.58.79 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

5.58.80 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)

5.58.81 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

5.58.82 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

5.58.83 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)

5.58.84 FDISK_getFreeSpaces

Description: get free spaces as array
Parameter:

- param: parameter string containing status informations about the harddisks
- dev: selected device (e.g. hda)

5.58.85 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.

5.58.86 FDISK_printColorDefinitions

Description: prints the color definitions for the filesystems

5.58.87 FDISK_showDiskDefine

Description: shows a dialog for defining the type and size of the fake drive for the clientBuilder
Parameter:

- client: client name

5.58.88 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.

5.58.89 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.

5.58.90 FDISK_rereadPartTable

Description: Let the OS re-read the partition table.
Parameter:

- path: The device that was changed/created (e.g. /dev/sda5).

5.58.91 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 .

5.58.92 FDISK_listPartJobs

Description: print all part jobs in the table
Parameter:

- partJobs: string with information about all created partition jobs

5.58.93 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)

5.58.94 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

5.58.95 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

5.58.96 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

5.58.97 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

5.58.98 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)

5.58.99 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)

5.58.100 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.

5.58.101 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.

5.58.102 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.

5.58.103 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.

5.58.104 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.

5.58.105 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

5.58.106 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, ...)

5.58.107 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

5.58.108 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

5.58.109 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.

5.59 ./inc/groups.php

Author: Hauke Goos-Habermann (HHabermann@pc-kiel.de)
Description: group management funtions

5.59.1 GRP_safeSettings

Description: Makes the contents of the input variables safe for SQL usage.
Parameter:

- groupName: name of the group
- type: Type of the group setting (can be used to identify the places where it should be used)
- var: Name of the group setting.
- val: Its value.

Returns:

- group ID on success, false otherwise.

5.59.2 GRP_setSetting

Description: Adds or changes a group setting.
Parameter:

- groupName: name of the group to add/change the setting for
- type: Type of the group setting (can be used to identify the places where it should be used)
- var: Name of the group setting.
- val: Its value.

Returns:

- true on successful adding/changing a group setting, false otherwise.

5.59.3 GRP_unsetSetting

Description: Removes a group setting.
Parameter:

- groupName: name of the group to add/change the setting for
- type: Type of the group setting (can be used to identify the places where it should be used)
- var: Name of the group setting.

Returns:

- true on successful removal of a group setting, false otherwise.

5.59.4 GRP_getSetting

Description: Gets the variable names, values and group setting types of a group setting.
Parameter:

- groupName: name of the group (required)
- type: (optional) Type of the group setting (can be used to identify the places where it should be used)
- var: (optional) Name of the group setting.

Returns:

- All variable names, values and group setting types of a group setting in an associative array or false if it's unset. ($out[$i]['var'], $out[$i]['val'], $out[$i]['type'])

5.59.5 GRP_getSettingsForClient

Description: Gets the variable names, values and group setting types of all group settings that are assigned to a client.
Parameter:

- clientName: name of the client (required)
- type: (optional) Type of the group setting (can be used to identify the places where it should be used)
- var: (optional) Name of the group setting.

Returns:

- All variable names, values and group setting types of a group setting in an associative array or false if the client is in no group. ($out[$i]['var'], $out[$i]['val'], $out[$i]['type'])

5.59.6 GRP_editSettingsDialog

Description: Shows a dialog for editing group settings.
Parameter:

- groupName: name of the group.

5.59.7 GRP_getSingleSetting

Description: Gets the value of a single group setting.
Parameter:

- groupName: name of the group
- type: Type of the group setting (can be used to identify the places where it should be used)
- var: Name of the group setting.

Returns:

- Value of a single group setting or false, if it's unset.

5.59.8 GRP_runSettingsForClient

Description: Run group settings for a given client.
Parameter:

- clientName: name of the client (required)
- type: (optional) Type of the group setting (can be used to identify the places where it should be used)
- var: (optional) Name of the group setting.

Returns:

- true, if there were group settings for the clients, otherwise false.

5.59.9 GRP_exists

Description: checks, if a group exists
Parameter:

- groupName: name of the group to check

5.59.10 GRP_add

Description: Adds a group with description.
Parameter:

- groupName: name of the group to add
- groupDescription: description of the group to add
- showInfo: Set to true, if an info message about the successfully creation should be shown.

Returns:

- true, of the group was added, otherwise false.

5.59.11 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.

5.59.12 GRP_getNameById

Description: gets the groupname of an Id
Parameter:

- groupId: Index of the group

Returns:

- Group name or false, if no matching group is found.

5.59.13 GRP_del

Description: deletes all clients from the group and the group itself
Parameter:

- groupName: name of the group

5.59.14 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

5.59.15 GRP_addClientToGroup

Description: adds a client to a group
Parameter:

- clientName: name of the client
- groupName: name of the group

5.59.16 GRP_delClientFromGroup

Description: removes a client from a group
Parameter:

- clientName: name of the client
- groupName: name of the group

5.59.17 GRP_setDescrGroup

Description: Sets the description of a client group.
Parameter:

- groupName: Name of the group
- groupDescription: New description of the group

5.59.18 GRP_getDescrGroup

Description: Gets the description of a client group.
Returns:

- Description of the group.

5.59.19 GRP_listGroupsAndCount

Description: returnes a associative array with all groupnames and the amount of clients in each group

5.59.20 GRP_showGroupsAndCount

Description: generates a table with all groupnames and the amount of clients in each group

5.59.21 GRP_ren

Description: renames a group
Parameter:

- groupName: name of the group
- newGroupName: name of the new group

5.59.22 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

5.59.23 GRP_countClients

Description: returns the amount of client of a certain group
Parameter:

- groupName: name of the group

5.59.24 GRP_showGeneralInfo

Description: shows a table with general information about the group
Parameter:

- groupName: name of the group

5.59.25 GRP_showRenDialog

Description: shows a dialog to rename a group
Parameter:

- groupName: name of the group

5.59.26 GRP_showChangeDescriptionDialog

Description: shows a dialog to change the group description
Parameter:

- groupName: name of the group

5.59.27 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

5.59.28 GRP_listGroups

Description: returnes all groups in an array

5.59.29 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

5.59.30 GRP_showDelDialog

Description: shows a dialog for deleting a group
Parameter:

- groupName: name of the group

5.59.31 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)

5.59.32 GRP_listClientGroups

Description: returnes an array containing all groups a client is in
Parameter:

- clientName: name of the client

5.59.33 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.

5.59.34 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

5.59.35 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

5.59.36 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 stored in an array or NULL, if groups should be ignored and all distributions and sources lists should be shown.
- 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

5.59.37 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
- withAutoUpdateJob: If set to true, only clients with auto update job will be listed.

Returns:

- Array with the found client names.

5.59.38 GRP_HTMLBackToOverview

Description: generates HTML code to return to the group overview page

5.59.39 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

5.59.40 GRP_desasterRecovery

Description: recovers all selected clients and shows a message afterwards
Parameter:

- clients: an array containing all clients that should be recovered

5.60 ./inc/halfSister.php

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

5.60.1 HS_ASSI_getClientSettingsCommand

Description: Generates the commands to send the client infos to the server (This has the same functionality as MSR_getClientSettingsCommand) if this is an halfSister client to be assimilated.

5.60.2 HS_getClientSettingsCommand

Description: Generates the commands to send the client infos to the server (This has the same functionality as MSR_getClientSettingsCommand).

5.60.3 HS_ASSI_statusFileCommand

Description: Generates the commands to send the package infos to the server (This has the same functionality as MSR_statusFileCommand) if this is an halfSister client to be assimilated.

5.60.4 HS_ASSI_prepareClient

Description: Prepares a halfSister client for assimilisation.

5.60.5 HS_getm23HSAdminPath

Description: Calculates the complete local path (including the file name) to m23HSAdmin for a choosen distribution release.
Parameter:

- release: The release name of the distribution

Returns:

- The complete local path (including the file name) to m23HSAdmin.

5.60.6 HS_getPackageCacheName

Description: Returns the complete name (including path) of the package cache.
Parameter:

- packagesourcename: name of the package source
- arch: CPU architecture of the client.

Returns:

- The name of the package cache.

5.60.7 HS_pkgUpdateCacheOnServer

Description: updates the package list on the server (for editing package selections) and saves it gzipped in /m23/data+scripts/distr/halfSister/packages-$packagesourcename-$arch.gz
Parameter:

- packagesourcename: name of the package source
- force: force the update (or not)
- arch: CPU architecture of the client.

Returns:

- The name of the package cache.

5.60.8 HS_fetchAndExtractOSImage

Description: Downloads and extracts a halfSister distribution.
Parameter:

- distr: halfSister distribution to install.
- arch: The architecture of the distribution (amd64 or i386)
- DNSServers: DNS server for resolving the names of the installation server
- gateway: gateway for fetching the packages
- packageProxy: the ip of the proxy the packages should be fetched from
- packagePort: the proxy port

5.60.9 HS_fetchm23HSAdminAndm23hwscannerByOS

Description: Fetches the m23HSAdmin tool and m23hwscanner by detection of the OS.

5.60.10 HS_fetchm23HSAdminAndm23hwscanner

Description: Fetches the m23HSAdmin tool and m23hwscanner matching the given distribution.
Parameter:

- release: halfSister distribution to get the m23HSAdmin tool for.
- pathPrefix : Extra path to put before the store path (e.g. to store the m23HSAdmin on a mounted filesystem).

5.60.11 HS_netConfig

Description: Sets IP, gateway, netmask, DNS and hostname.
Parameter:

- clientParams: Associated array with the parameters of the client.

5.60.12 HS_setPackageProxy

Description: Sets the proxy for the package management tool.
Parameter:

- clientOptions: Associated array with the options of the client.

5.60.13 HS_setSourcesList

Description: Writes the package sources list for the client's package manager.
Parameter:

- clientName: name of the client

5.60.14 HS_normalUpdate

Description: Performs a normal update of the installed packages.

5.60.15 HS_pkgFullUpdate

Description: Performs a full update of the installed packages.

5.60.16 HS_sysSetm23ClientID

Description: Sets the m23 client ID.
Parameter:

- clientParams: Associated array with the parameters of the client.

5.60.17 HS_netEnableNTP

Description: Enable getting the system time by NTP.

5.60.18 HS_netDisableNTP

Description: Disable getting the system time by NTP.

5.60.19 HS_hookBeginAfterChroot

Description: Scripts that should be run at the beginning of the afterChroot.
Parameter:

- rootDevice: Device with partition where the os will be installed (e.g. /dev/hda1).
- bootDevice: Device (with partition) where the bootmanager will be installed (e.g. /dev/hda).
- bootDeviceFS: Filesystem of the root device.

5.60.20 HS_hookEndAfterChroot

Description: Scripts that should be run at the end of the afterChroot.
Parameter:

- rootDevice: Device with partition where the os will be installed (e.g. /dev/hda1).
- bootDevice: Device (with partition) where the bootmanager will be installed (e.g. /dev/hda).
- bootDeviceFS: Filesystem of the root device.

5.60.21 HS_pkgInstallBasePackages

Description: Installs basic packages

5.60.22 HS_netSetm23SSLCertificate

Description: Downloads and stores the SSL public key of the m23 server into the correct directory.
Parameter:

- pathPrefix : Extra path to put before the SSL store path (e.g. to store the SSL key on a mounted filesystem).

5.60.23 HS_sysSetLanguage

Description: Sets the system language.
Parameter:

- lang: Two-character code of the language to set (de, en, fr).

5.60.24 HS_sysSetRootPW

Description: Sets the root password.
Parameter:

- Password: The crypted password of the user root.

5.60.25 HS_sysSetTimeZone

Description: Sets the time zone.
Parameter:

- timezone: Timezone in POSIX notation (e.g. Europe/Berlin)

5.60.26 HS_sysHWsetup

Description: Detects and configures new hardware

5.60.27 HS_sysAddUser

Description: Creates a new user with home directoy and sets password.
Parameter:

- username: Name of the user to add.
- password: The password of the user to add.
- uid: User ID to give the new user.
- gid: Group ID to give the new user.

5.60.28 HS_sysChangeUser

Description: changes the settings of an useraccount on a client
Parameter:

- userName: the (old) username for the account
- password: the new unecrypted password for the account
- newUserName: the new username

5.60.29 HS_netEnableSSHdAndImportKey

Description: Enables the SSH daemon and adds a SSH key to let the m23 server log into the machine.

5.60.30 HS_writeHosts

Description: Writes the /etc/hosts file for the client

5.60.31 HS_sysWriteM23fetchjob

Description: Generates the m23fetchjob script and adds it to the init levels.

5.60.32 HS_sysWriteCrontabm23fetchjobEvery5Minutes

Description: Adds entries to crontab to check every 5 minutes for new jobs.
Parameter:

- clientParams: Associated array with the parameters of the client.

5.60.33 HS_sysInstallKernel

Description: Installs a matching kernel.
Parameter:

- kernelPkg: Name of the kernel package to install.

5.60.34 HS_netEnableNFSHome

Description: Enables storing of home directories on a NFS server
Parameter:

- nfsURL: URL to the NFS storage (e.g. 192.168.1.42/up/home)

5.60.35 HS_netEnableLDAP

Description: Enables LDAP login for a client.
Parameter:

- clientOptions: the client's options array

5.60.36 HS_sysAddFstabEntries

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

- fstab: All fstab lines to add as assiciative array.

5.60.37 HS_sysMakeBootable

Description: Makes the system bootable.
Parameter:

- rootDevice: Device with partition where the os will be installed (e.g. /dev/hda1).
- bootDevice: Device (with partition) where the bootmanager will be installed (e.g. /dev/hda).
- bootDeviceFS: Filesystem of the root device.

5.60.38 HS_sysConfigurePrinter

Description: Detects and configures printer(s).

5.60.39 HS_sysInstallPrinter

Description: Installs the printer software.

5.60.40 HS_pkgInstallKDE

Description: Installs KDE

5.60.41 HS_pkgInstallX

Description: Installs XOrg or another shipped X11 server.

5.60.42 HS_pkgInstalledList

Description: Lists the installed packages or writes the list to a file.
Parameter:

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

5.60.43 HS_pkgInstall

Description: Installs one or more packages
Parameter:

- packages: List of packages to install seperated by spaces.

5.60.44 HS_runClientPackageConfDB

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

- clientName: Name of the client.

5.60.45 HS_pkgSearch

Description: Searches for available packages matching all keywords.
Parameter:

- searchTerms: Search terms seperated spaces.

5.60.46 HS_pkgInstallPreview

Description: Generates commands for getting a installation preview on the client.
Parameter:

- packages: List of packages to install seperated by spaces

5.60.47 HS_pkgDeinstallPreview

Description: Generates commands for getting a deinstallation preview on the client.
Parameter:

- packages: List of packages to deinstall seperated by spaces

5.60.48 HS_statusFileCommand

Description: Generates the commands to send the package infos to the server (This has the same functionality as MSR_statusFileCommand).

5.60.49 HS_pkgUpdateCache

Description: Updates the list of available packages.

5.60.50 HS_pkgInstallGnome

Description: Installs Gnome.

5.60.51 HS_pkgInstallLXDE

Description: Installs LXDE.

5.60.52 HS_pkgInstallXFce

Description: Installs XFce.

5.60.53 HS_pkgDeinstall

Description: Deinstalls one or more packages.
Parameter:

- packages: List of packages to deinstall seperated by spaces.

5.60.54 HS_wrapperHS_wrapperReturn

Description: Creates a m23HSAdmin action with parameters and returns the result.
Parameter:

- Multiple parameters with the desired action as first parameter followed by the parameters of the action.

5.60.55 HS_wrapper

Description: Creates a m23HSAdmin action with parameters and shows the result.
Parameter:

- Multiple parameters with the desired action as first parameter followed by the parameters of the action.

5.60.56 HS_sendCommandExecutionStatus

Description: Sends a status message for the finished HS job and if there were errors, the log file too.
Parameter:

- cmd: Name of the HS command

5.61 ./inc/helper.php

Author: Hauke Goos-Habermann (HHabermann@pc-kiel.de)
Description: Helper functions that did not fit into another include file.

5.61.1 HELPER_arrayString2AssociativeArray

Description: Parses a string holding array information to an associative array.
Parameter:

- in: String holding array information (eg. "a => 'a', 0 => 'zero'")

Returns:

- Associative array

5.61.2 HELPER_logOnClientBASH

Description: Generates BASH code to show and log a status message with timestamp.
Parameter:

- logFile: Name of the log file on the client.
- message: Message to show.

5.61.3 HELPER_each

Description: Replacement for the deprecaded each function: Returns the next element from an array on each call.
Parameter:

- arr: Input array.

Returns:

- : Array with (0 => $key, 1 => $val, 'key' => $key, 'value' => $val) or NULL, if there are no more elements in the array.

5.61.4 HELPER_getClientNameAndID

Description: Gets client ID and name from the $_POST or $_GET array.
Parameter:

- client: Variable to write the client name to.
- id: Variable to write the client ID to.

5.61.5 HELPER_indentLines

Description: Intents all lines by putting an intent string in front of them.
Parameter:

- in: Input text.
- intent: Text tu put in front of all lines.

Returns:

- Text with intented lines.

5.61.6 HELPER_filterOutUnwantedSSHOutputs

Description: Filters out unwanted lines from SSH client.
Parameter:

- text: Text to filter.

Returns:

- Text with all lines removed that are containing unwanted lines from SSH client.

5.61.7 dieWithExitCode

Description: Let the script die with a message on stderr and exit code.
Parameter:

- exitMessage: Message to show on exit.
- exitCode: The exit code of the script.

5.61.8 HELPER_str_equal_UTF8ISO

Description: Checks, if two strings are equal independent from their encodings (UTF-8 or ISO).
Parameter:

- a: First string to compare.
- b: Second string to compare.

Returns:

- true, if both strings "mean" the same, otherwise false.

5.61.9 HELPER_logToFile

Description: Adds text to an exclusively opened log file.
Parameter:

- logFile: Name of the log file with full path.
- text: Text to add.
- htmlH: If nummeric, the text will shown with the given HTML heading level.

5.61.10 HELPER_splitDayHourMinuteString

Description: Splits a combined numeric day and hour/minute string into day and hour/minute.
Parameter:

- in: Combined numeric day and hour/minute string
- day: The day variable to write the day number to.
- hour: The hour variable to write the hour number to.
- minute: The minute variable to write the minute number to.

Returns:

- true, if the input string is long enough.

5.61.11 HELPER_showScriptHeader

Description: Shows a header for own scripts, that creates a log file with the package name and the start time. Sends the log file to the m23 server and shows an installation dialog on the client's screen.
Parameter:

- id: Job ID of the script
- packName: Name of the package

5.61.12 HELPER_showScriptFooter

Description: Shows a header for own scripts, that saves the script end time to a log file and sends it to the m23 server. Sets status bar to 100**Parameter:**

- id: Job ID of the script.
- packName: Name of the package.

5.61.13 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

5.61.14 HELPER_isUpper

Description: Checks, if a character is upper case
Parameter:

- char: Character to check.

Returns:

- true, when upper case otherwise false

5.61.15 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.

5.61.16 HELPER_isExecutedInCLI

Description: Checks, if it is run in CLI.
Returns:

- True, when run in CLI otherwise false.

5.61.17 HELPER_pageFetchedByBrowser

Description: Detects, if the current page or BASH script was fetched by a webbrowser (and not wget).
Returns:

- True, if the current page or BASH script was fetched by a webbrowser, otherwise false.

5.61.18 HELPER_isExecutedOnUCS

Description: Checks, if it is run on UCS.
Returns:

- True, when run on UCS otherwise false.

5.61.19 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)
- useProxy: If set to true, the system wide proxy will be used, otherwise a direct connection is attempted.

Returns:

- The downloaded site code or false in case of an error.

5.61.20 HELPER_trimValue

Description: Runs trim on the input parameter and writes the result back.
Parameter:

- value: Value to trim.

5.61.21 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.

5.61.22 HELPER_resetNewLogLines

Description: Resets the line number of the last read line.
Parameter:

- sessionPrefix: Prefix for storing the last read line number in the session.

5.61.23 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.
- filterFunction: Function to filter the lines before adding them to the output. The function gets the unfiltered string as input and returns the filtedred version.

Returns:

- UTF8-encoded new lines of the log file.

5.61.24 HELPER_rmRecursive

Description: Removes a directory with sub-directories and contained files.
Parameter:

- : dir: Full path to the directory.

5.61.25 HELPER_showBAfH

Description: Shows the German BAfH excuse of the day.

5.61.26 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.

5.61.27 HELPER_md5x5

Description: Hashes an input value 5 times with MD5.
Parameter:

- : in: Input to hash.

Returns:

- Hashed value.

5.61.28 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.

5.61.29 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

5.61.30 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.

5.61.31 HELPER_importAllIntoPOST

Description: Imports all values into the $_POST array in case that there are too much array keys for the normal processing.

5.61.32 HELPER_randomUsername

Description: Generates a random username with a given length.
Parameter:

- length: Length of the username to create.

Returns:

- The random username.

5.61.33 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.

5.61.34 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.

5.61.35 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.

5.61.36 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.

5.61.37 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))

5.61.38 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, ...)

5.61.39 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.

5.61.40 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.

5.61.41 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.
- noProxy: Set to true, if the system proxy should not be used.

Returns:

- The contents of the files from chache or from download or false if no file could be found.

5.61.42 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.

5.61.43 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.

5.61.44 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).

5.61.45 HELPER_generateSalt

Description: Generates a random salt string.
Parameter:

- length: Length of the salt.

Returns:

- Random salt of given length.

5.61.46 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.

5.61.47 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.

5.61.48 HELPER_getBootLoaders

Description: Returns a list of available bootloaders.
Returns:

- Array with available bootloaders.

5.61.49 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.

5.61.50 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

5.61.51 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.

5.61.52 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.

5.61.53 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.

5.61.54 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.

5.61.55 HELPER_getApacheUser

Description: returnes the name of the Apache user

5.61.56 HELPER_getApacheGroup

Description: returnes the group of the Apache user

5.61.57 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.

5.61.58 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

5.61.59 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

5.61.60 HELPER_maxPhpUploadSize

Description: Returns the maximum file upload size allowed by php.ini.

5.61.61 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

5.61.62 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

5.61.63 HELPER_hostname2IP

Description: Gets the IP of a known host.
Parameter:

- host: Name of the host, IP or FQDN.

Returns:

- IP of the host or false, if no IP could be found.

5.62 ./inc/help.php

Author: Hauke Goos-Habermann (HHabermann@pc-kiel.de)
Description: function to show the help box in the correct language

5.62.1 HELP_showHelpTex2

Description: shows the help file converted to LaTeX code
Parameter:

- fileName: name of the help file

5.62.2 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,...)

5.62.3 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

5.62.4 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,...)

5.62.5 HELP_showHelpTex

Description: shows the help file converted to LaTeX code
Parameter:

- fileName: name of the help file

5.63 ./inc/html.php

Author: Hauke Goos-Habermann (HHabermann@pc-kiel.de)
Description: functions for generating often used HTML code

5.63.1 HTML_showJSsanitizeASCII

Description: Shows the JavaScript function sanitizeASCII, that will convert all characters to lower case, replace all umlauts, delete all characters that are not 0-9, a-z or '.' from an input element.
Parameter:

- htmlName: Name of the HTML element to sanitize.

5.63.2 HTML_showJSSanitizeMAC

Description: Shows the JavaScript function sanitizeMAC, that will delete all non hex values from an input element.
Parameter:

- htmlName: Name of the HTML element to sanitize.

5.63.3 HTML_rowColor

Description: Toggles the row color between oddrow and evenrow class.
Parameter:

- lastRow: External variable to store the last row state.

Returns:

- Newly set row color.

5.63.4 HTML_getInvisiblePasswordsIfFeatureEnabled

Description: Makes a password invisible by setting text color to "transparent", if the "makePasswordsInvisibleEnabled" feature is active, otherwise it will be shown normally.
Parameter:

- pass: The password to make (in)visble.

Returns:

- Visible or invisible password.

5.63.5 HTML_waitAnimation

Description: Defines HTML code for showing a waiting animation and onClick code for insering into buttons. The JavaScript showing function is given out where the PHP function is called.
Parameter:

- htmlName: Base name for the defines and naming of the JavaScript function and DIVs.
- waitText: Text to show while the animation runs.
- width: Width for the animation image or unscaled if not set.

5.63.6 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.

5.63.7 HTML_getOriginalUploadFilename

Description: Get the original file name of an uploaded file.
Parameter:

- htmlName: Name of the HTML element

Returns:

- Original file name of an uploaded file or NULL, if none is given.

5.63.8 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.

5.63.9 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.

5.63.10 HTML_weekdayTimeChooser

Description: Creates a picker for weekday and hour/minute (with 15 minute steps).
Parameter:

- htmlNames: Name of the weekday and hour/minute picker.

Returns:

- Choosen weekday and hour/minute as DHHMM string.

5.63.11 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).

5.63.12 HTML_getOutputBuffer

Description: Gets the complete (previously rendered) HTML output buffer.
Returns:

- The complete contents of the (previously rendered) HTML output buffer.

5.63.13 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.

5.63.14 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.

5.63.15 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.

5.63.16 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.

5.63.17 HTML_showTitle

Description: Shows a title.
Parameter:

- title: Text of the title.

5.63.18 HTML_showSmallTitle

Description: Shows a title of the second tier.
Parameter:

- title: Text of the small title.

5.63.19 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.
- storeIntoPreferenceSpace: Set to true, if the value should be stored under the prefKey in the preferenceSpace.

5.63.20 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).

5.63.21 HTML_checkboxChangerButtons

Description: Defines buttons for changing all check boxes.
Parameter:

- htmlNames: Name of the HTML elements.

5.63.22 HTML_jsCheckboxChanger

Description: Generates JavaScript code for changing all check boxes.
Parameter:

- jsBlockName: Name of the JS block constant.

5.63.23 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.

5.63.24 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.

5.63.25 HTML_questionnaire

Description: Shows the questionnairem window.
Parameter:

- disable: Set to true to disable showing of the window again.

5.63.26 HTML_esel

Description: Shows an dog-ear that can be opened to show "goos-habermann.de/m23ad".

5.63.27 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.

5.63.28 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.

5.63.29 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.

5.63.30 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.

5.63.31 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.

5.63.32 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.

5.63.33 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.

5.63.34 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.

5.63.35 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.

5.63.36 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.

5.63.37 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.

5.63.38 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.

5.63.39 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.

5.63.40 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.

5.63.41 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.

5.63.42 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.

5.63.43 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.

5.63.44 HTML_setPage

Description: Sets the m23 page as hidden value.
Parameter:

- page: Name of the page.

5.63.45 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)
- extraHTML: Extra HTML/JavaScript code

Returns:

- Returns the entered value, the default value or false.

5.63.46 HTML_storable2xPassword

Description: HTML 2x password edit lines for entering and re-entering a the same password.
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: unused, but there to be parameter compatible with HTML_storableInput.
- extraHTML: Extra HTML/JavaScript code

Returns:

- Returns the password, if it's equal in both password edit lines, or false.

5.63.47 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.

5.63.48 HTML_storableMultiSelection

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.
- multipleSize: Number of elements to display.
- 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.

5.63.49 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.

5.63.50 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.

5.63.51 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

5.63.52 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.

5.63.53 HTML_showTableEnd

Description: prints the end of a shadowed table
Parameter:

- centerTable: set to true if the table should be centered vertically

5.63.54 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

5.63.55 HTML_showFormEnd

Description: Shows the end of a formular
Parameter:

- addSessionAsHidden: Set to true, to add $_SESSION as hidden variable.

5.63.56 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.
- allowDoubleDefinition: If set to true, HTML element constants will be defined even if there is a previously defined constant with the same name. This will run into an error and helps debugging.

Returns:

- True if it was clicked otherwise false.

5.63.57 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.

5.63.58 HTML_button

Description: Defines a (graphical) button
Parameter:

- htmlName: Name of the HTML element.
- label: Label of the button
- value: value of the button (not label)
- extra: Extra options for the HTML input tag.

5.63.59 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)
- forceDefaultSelection: Don't query $_POST, use what is supplied by htmlValue
- extraHTML: Extra HTML/JavaScript code
- Returns the entered value, the default value or false.

5.63.60 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

5.63.61 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 of 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.

5.63.62 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.
- forceDefaultSelection: Don't query $_POST or prefKey, use what is supplied by defaultSelection

Returns:

- The value of the selected element or false if nothing was selected.

5.63.63 HTML_datalist

Description: Shows a datalist with auto-completion. Caution: Not supported by Safari browser 12.0 (or earlier).
Parameter:

- htmlName: Name of the HTML element.
- array: An array that hold the returned values (array keys) the naming for the elements (array values).
- 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.
- forceDefaultSelection: Don't query $_POST, use what is supplied by defaultSelection

Returns:

- The value of the selected element or false if nothing was selected.

5.63.64 HTML_imgSelection

Description: Shows a jQuery enhanced SELECT
Parameter:

- htmlName: Name of the HTML element.
- array: An array that hold the returned values (array keys) the naming for the elements (array values).
- imgArray: An array containing the images for our options.
- 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.
- forceDefaultSelection: Don't query $_POST, use what is supplied by defaultSelection

Returns:

- The value of the selected element or false if nothing was selected.

5.63.65 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.
- extraHTML: Extra HTML/JavaScript code

Returns:

- True if the check box is checked.

5.63.66 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.

5.63.67 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.
- allowDoubleDefinition: If set to true, HTML element constants will be defined even if there is a previously defined constant with the same name. This will run into an error and helps debugging.

5.63.68 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.

5.63.69 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.

5.63.70 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.

5.63.71 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.

5.63.72 HTML_showPagePrintButton

Description: Shows a print button that allows easy printing of the current m23 administration interface.

5.64 ./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.

5.64.1 HWINFO_getParam

Description: get hardware infos
Parameter:

- paramName: parameter to read from the hardware info (e.g. cpu, mem, ...)
- clientName: name of the client

5.64.2 HWINFO_getMemory

Description: returns the size of memory
Parameter:

- clientName: name of the client

5.64.3 HWINFO_getHDSize

Description: Returnes the sizes of all harddisks in a string, sperated by html breaks
Parameter:

- clientName: name of the client

5.64.4 HWINFO_printPartitions

Description: prints the partition information
Parameter:

- clientName: name of the client

5.64.5 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.

5.64.6 DMI_getParam

Description: get dmi info for a special parameter
Parameter:

- paramName: name of dmi setting
- clientName: name of the client

5.64.7 DMI_getBoard

Description: get the dmi board informations
Parameter:

- clientName: name of the client

5.64.8 DMI_getMemory

Description: get the dmi memory informations
Parameter:

- clientName: name of the client

5.64.9 DMI_getCPU

Description: get the dmi cpu informations
Parameter:

- clientName: name of the client

5.64.10 DMI_getChassis

Description: get the dmi chassis information
Parameter:

- clientName: name of the client

5.64.11 DMI_getCache

Description: get the dmi cache information
Parameter:

- clientName: name of the client

5.64.12 DMI_getSlot

Description: get the dmi information about slots
Parameter:

- clientName: name of the client

5.65 ./inc/i18n.php

Author: Hauke Goos-Habermann (HHabermann@pc-kiel.de)
Description: functions for multi language handling in m23.

5.65.1 I18N_safeUTF8_encode

Description: Converts a string to UTF-8 and makes sure that it is not converted again, if it was UTF-8.
Parameter:

- in: String to convert.

Returns:

- String converted to UTF-8.

5.65.2 I18N_isUTF8

Description: Checks, if a string is UTF-8 encoded.
Parameter:

- in: String to check.

Returns:

- true, if input string is UTF-8 encoded, otherwise false.

5.65.3 I18N_replaceUmlaute

Description: Replaces German "Umlaute" and language specific charaters of other languages.
Parameter:

- in: String to convert.

Returns:

- String with only ASCII characters.

5.65.4 I18N_getTimeDateAndElapsedMinutes

Description: Takes a timestamp, calculates its date and time sting in the given language and the minutes from the timestamp to the current time.
Parameter:

- time: The timestamp.

Returns:

- Time information string in the given language.

5.65.5 I18N_getAllTranslationsForAllVariables

Description: Translates all I18N variables (in all languages) to their language dependent strings.
Parameter:

- in: Input text.

Returns:

- Associative array with all languages as keys and the I18N variables translated to their language dependent strings.

5.65.6 I18N_getAllTranslationsForVariable

Description: Returns all translations for a given variable.
Parameter:

- var: Name of the variable (without "$")

Returns:

- All translations for a given variable.

5.65.7 I18N_getHumanReadableDayHourMinute

Description: Converts a combined numeric day and hour/minute string into a human readable day and hour/minute string.
Parameter:

- in: Combined numeric day and hour/minute string

Returns:

- Human readable day and hour/minute string or false in case of conversion error.

5.65.8 I18N_getWeekDayArray

Description: Generates an array with all week days as value and week day numbers as keys (Monday = 1, Sunday = 7).
Returns:

- Array with all week days.

5.65.9 I18N_number_format

Description: Converts numbers to the language specific number formating.
Parameter:

- in: Input number.

Returns:

- Language specific number formated number.

5.65.10 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.

5.65.11 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.

5.65.12 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)

5.65.13 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.

5.65.14 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.

5.65.15 I18N_cacheWebinterfaceLanguages

Description: Caches the language information from the language.info files to the DB.

5.65.16 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.

5.65.17 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.

5.65.18 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.

5.65.19 I18N_cacheClientLanguages

Description: Caches the client languages in the DB.

5.65.20 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))

5.66 ./inc/ldap.php

Author: Hauke Goos-Habermann (HHabermann@pc-kiel.de)
Description: Functions for using a LDAP server

5.66.1 LDAP_getPasswordHashTypes

Description: Returns an array with the known password hash types.
Returns:

- Array with the known password hash types.

5.66.2 LDAP_getSSHA

Description: Hashes a password in SSHA as LDAP/PAM expects it.
Parameter:

- pwd: Password to hash.

Returns:

- Hashed password.

5.66.3 LDAP_getMD5

Description: Hashes a password in MD5 as LDAP/PAM expects it.
Parameter:

- pwd: Password to hash.

Returns:

- Hashed password.

5.66.4 LDAP_needsBase64UTF8

Description: Checks, if a string must be converted to UTF-8 and Base64 because it contains non-ASCII characters.
Parameter:

- in: String to check.

Returns:

- true, of converting is needed, otherwise flase.

5.66.5 LDAP_getBase64UTF8

Description: Converts a string to UTF-8 and Base64.
Parameter:

- in: String to convert.
- forceConvert: Force conversion.

Returns:

- String in UTF-8 and Base64

5.66.6 LDAP_getTypes

Description: Returns an array with the known LDAP types.
Returns:

- Array with the known LDAP types.

5.66.7 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

5.66.8 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

5.66.9 LDAP_getValueFromConfigLine

Description: Returns the value from a phpLDAPadmin configuration line.
Returns:

- Value from a phpLDAPadmin configuration line.

5.66.10 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.

5.66.11 LDAP_loadServer

Description: Loads the variables from a LDAP server.
Parameter:

- name: server name

5.66.12 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

Returns:

- true or error message string in case of an error.

5.66.13 LDAP_fqdn2dn

Description: Returns the DN converted from a FQDN
Parameter:

- domain: a full qualified domain name (e.g. test.m23.de)

5.66.14 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
- port: Port number, the LDAP server runs on.
- minUidNumber: Minimum user ID.
- minGidNumber: Minimum group ID.
- passwordHashAlgorithm: Algorithm for hashing the passwords.
- additionalA: Associative array with additional variables and values (eg. array('realm' => 'REALM', 'krbpass' => 'test'))

5.66.15 LDAP_delServerFromphpLdapAdmin

Description: Deletes a LDAP server from the phpLDAPadmin configuration file.
Parameter:

- name: the name of the LDAP server that should be deleted

5.66.16 LDAP_checkphpLdapAdminConfiguration

Description: Checks if the phpLDAPadmin configuration file is existing and creates it if it's missing

5.66.17 LDAP_showServerManagementDialog

Description: Shows a dialog for adding, removing and changing LDAP servers.

5.66.18 LDAP_I18NLdapType

Description: Returns the human readable description of the LDAP usage type
Parameter:

- status: status string

5.66.19 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)

5.66.20 LDAP_getNextID

Description: Returns an array with free (posix user or group) IDs or a single free ID from the LDAP database.
Parameter:

- serverName: Name of the LDAP server
- attribute: Name ID attribute (eg. uidnumber)
- container: Container (subtree) in the DIT to search for the IDs. If empty the whole database will be searched.
- filter: Filter for only handling entries that match.
- serverMinimumSettingsVar: The name of the variable that holds the minimum allowed ID value.
- amount: Amount of IDs to generate.
- minimumID: If set, this ID will be used as start ID instead of the ID set by server information.

Returns:

- Array with free IDs in the given amount or a single free ID as int.

5.66.21 LDAP_getFreeUserIDs

Description: Returns an array with free posix group IDs or a single free ID from the LDAP database.
Parameter:

- amount: Amount of IDs to generate.
- minimumID: If set, this ID will be used as start ID instead of the ID set by server information.

Returns:

- Array with free user IDs in the given amount or a single free user ID as int.

5.66.22 LDAP_getFreeGroupIDs

Description: Returns an array with free posix group IDs or a single free ID from the LDAP database.
Parameter:

- amount: Amount of IDs to generate.
- minimumID: If set, this ID will be used as start ID instead of the ID set by server information.

Returns:

- Array with free group IDs in the given amount or a single free group ID as int.

5.67 ./inc/m23shared/m23shared-raus.php

Author: Hauke Goos-Habermann (HHabermann@goos-habermann.de)
Description: database functions, open, close the database, get ip of the calling client

5.67.1 m23SHARED_deleteBills

Description: Deletes all bills.

5.67.2 m23SHARED_deleteUserdataFromDB

Description: Deletes user data not needed for archiving purpose.

5.67.3 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.

5.67.4 m23SHARED_markForDeletion

Description: Marks the current user's data for deletion and disable his login.

5.67.5 m23SHARED_DBname

Description: Generates a database name for m23 shared by input and salt.
Parameter:

- adminName: Name of the m23shared admin

Returns:

- Database name.

5.67.6 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.

5.67.7 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.

5.67.8 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.

5.67.9 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.

5.67.10 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.

5.67.11 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)

5.67.12 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.

5.67.13 m23SHARED_paidClients

Description: Calculates and returns the amount of clients the customer has paid for.
Returns:

- : Amount of clients the customer has paid for.

5.67.14 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)

5.67.15 m23SHARED_getServerIP

Description: Returns the FQDN of the m23shared server.
Returns:

- : FQDN of the m23shared server.

5.67.16 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.

5.67.17 m23SHARED_getSalutationsArray

Description: Returns an array with salutation forms.
Returns:

- : Array with gender character as key and localised salutation.

5.67.18 m23SHARED_getSalutationHeadline

Description: Returns a saluation headline that can be used as beginning of an email.
Returns:

- : Salutation headline in current language.

5.67.19 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.

5.67.20 m23SHARED_getMonthlyFee

Description: Calculates the monthly fee with the given amount af paid clients.
Returns:

- : Monthly price.

5.67.21 m23SHARED_getCustomerEmail

Description: Returns the eMail address of customer.
Returns:

- : eMail address of customer.

5.67.22 m23SHARED_setCustomerEmail

Description: Sets the eMail address of customer.

5.67.23 m23SHARED_generateActivationKey

Description: Calculates a random activation key and stores it into the DB.
Returns:

- : Random activation key.

5.67.24 m23SHARED_getActivationKey

Description: Returns the activation key for this customer.
Returns:

- : Activation key.

5.67.25 m23SHARED_generateCustomerNumber

Description: Calculates the customer number by current time and random value and stores it into the DB.
Returns:

- : Customer number.

5.67.26 m23SHARED_getCustomerNr

Description: Returns the customer number for this customer.
Returns:

- : Customer number.

5.67.27 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.

5.67.28 m23SHARED_setCustomerLanguage

Description: Sets the language for this customer.
Parameter:

- lang: Language for the customer.

5.67.29 m23SHARED_getCustomerLanguage

Description: Returns the language setting for this customer.
Returns:

- : Language for this customer.

5.67.30 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.

5.67.31 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.

5.67.32 m23SHARED_setRealName

Description: Sets the real name of the customer.
Parameter:

- name: Real name of the customer.

5.67.33 m23SHARED_getRealName

Description: Returns the real name of the customer.
Returns:

- : Real name of the customer.

5.67.34 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.

5.67.35 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.

5.67.36 m23SHARED_billDateSelection

Description: Creates an HTML selection with valid billing months.
Parameter:

- htmlName: Name of the HTML selection.

Returns:

- The selected month and year.

5.67.37 m23SHARED_priceFormater

Description: Formats a price with two digit decimal place.
Parameter:

- price: The price to format.

Returns:

- The formatted price.

5.67.38 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.

5.67.39 m23SHARED_getLicenseType

Description: Returns the m23shared license of the current m23shared user.
Returns:

- License type as constant.

5.67.40 m23SHARED_sendAdminMail

Description: Sends an GPG encrypted eMail to the admin.
Parameter:

- subject: The subject of the eMail.
- text: The eMail message.

5.67.41 m23SHARED_getCurrentUser

Description: Returns the current m23shared user.
Returns:

- Current m23shared user.

5.67.42 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.

5.67.43 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.

5.67.44 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.

5.67.45 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.

5.67.46 m23SHARED_getCustomerAddress

Description: Returns HTML formatted address information for the current customer.
Returns:

- Address information for the current customer.

5.67.47 m23SHARED_getCustomerBankHTML

Description: Returns HTML formatted bank account information for the current customer.
Returns:

- Bank account information for the current customer.

5.67.48 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.

5.67.49 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.

5.67.50 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.

5.67.51 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.

5.67.52 m23SHARED_showLicenseDialog

Description: Shows a dialog for viewing and changing the license and paid client amount.

5.67.53 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.

5.67.54 m23SHARED_switchUser

Description: Changes the current m23shared user.

5.67.55 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.

5.67.56 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.

5.67.57 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.

5.67.58 m23SHARED_addExtraBill

Description: Adds an extra entry to the bill.
Parameter:

- amount: Amount of good.
- description: Description of the good.
- unitprice: Price per unit.

5.67.59 m23SHARED_showDonationDialog

Description: Shows a dialog where the user can donate to the m23 project.

5.67.60 m23SHARED_showBootMediaDownloadDialog

Description: Shows a dialog with download icons for the different boot media.

5.67.61 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.

5.67.62 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.

5.67.63 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.

5.67.64 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.

5.67.65 m23SHARED_showAdminDialog

Description: Shows a dialog for the admin to search users for and change values.

5.67.66 m23SHARED_blockAccount

Description: Blocks or unblocks an account.
Parameter:

- block: Set to true blocks, false unblocks.

5.67.67 m23SHARED_isAccountBlocked

Description: Checks if an account is blocked.
Returns:

- : True if the account is blocked, false if not.

5.67.68 m23SHARED_showAddExtraBillDialog

Description: Shows a dialog for adding extra entries to the bill.

5.67.69 m23SHARED_showBillDialog

Description: Shows a dialog for viewing and choosing the bill.

5.67.70 m23SHARED_showPriceListTable

Description: Shows a table with the price list.

5.67.71 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.

5.68 ./inc/mail.php

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

5.68.1 MAIL_getKeyFromeMailAddress

Description: Generates an MD5 key from an eMail address.
Parameter:

- eMail: eMail to use as input for the hashing.

Returns:

- Key generated from the eMail.

5.68.2 MAIL_AESencode

Description: AES encryptes a message with a key.
Parameter:

- key: The passphrase to encode the message with.
- text: The message to encode.

Returns:

- The encrypted message.

5.68.3 MAIL_AESDecode

Description: AES decryptes a crypted message with a key.
Parameter:

- key: The passphrase to encode the message with.
- cText: The crypted message to decode.

Returns:

- The decrypted message.

5.68.4 MAIL_sendAESMail

Description: Sends an AES encrypted eMail to a crypt mail gateway.
Parameter:

- eMail: eMail address of the recipient.
- subject: The subject of the eMail.
- text: The eMail message.

5.68.5 MAIL_attach

Description: Attaches a file to the message body of the mail and changes the mail header.
Parameter:

- file: The file with full path to attact.
- header: The original header that will be modified.
- message: The text message.
- fileName: Alternate file name for the attacement as it should be seen by the eMail client.

5.68.6 MAIL_getHeader

Description: Generates a mail header with sender and reply-to field, mail software and a BCC to send a copy to the admin.
Parameter:

- from: The sender's eMail address

Returns:

- : Complete mail header.

5.68.7 MAIL_gpgSignDetached

Description: Creates a detached signature file for a given private GPG key ID and input file.
Parameter:

- gpgID: ID of the private GPG key.
- inFile: The file to create a signature for.
- outFile: The file with the detached signature.

Returns:

- : true, if the signature file was created and the input file exists, otherwise false.

5.68.8 MAIL_gpgSignClear

Description: Creates a clear text signature file for a given private GPG key ID and input file.
Parameter:

- gpgID: ID of the private GPG key.
- inFile: The file to create a signature for.
- outFile: The file with the detached signature.

Returns:

- : true, if the signature file was created and the input file exists, otherwise false.

5.68.9 MAIL_gpgSign

Description: Creates a signature file for a given private GPG key ID and input file with GPG parameters.
Parameter:

- gpgID: ID of the private GPG key.
- inFile: The file to create a signature for.
- outFile: The file with the detached signature.
- gpgParams: GPG parameters to specify the type of the signature.
- user: The owner of the output file.
- mode: The file mode of the output file.

Returns:

- : true, if the signature file was created and the input file exists, otherwise false.

5.68.10 MAIL_gpgGettKey

Description: Gets the public GPG key as ASCII.
Parameter:

- gpgID: ID of the public GPG key.

Returns:

- : false, if there is no public GPG key for the given GPG ID.

5.68.11 MAIL_gpgCheckKey

Description: Checks, if a GPG key ID has a valid private or public key.
Parameter:

- gpgID: ID of the private GPG key.
- privateKey: If set to true, the validity as private key is checked.

Returns:

- : true, if the GPG key is valid.

5.68.12 MAIL_gpgMail

Description: Encrypts a message with GPG for a given eMail address.
Parameter:

- message: The message text to encrypt.
- eMail: The eMail address to search the GPG key for and to encrypt to.

Returns:

- : Encrypted message or false if there were errors.

5.68.13 MAIL_getGpgKeyList

Description: Gets the list of known GPG keys/identities.
Parameter:

- listSecretKeys: If set to true, available secret keys are listed.

Returns:

- : Associative array with key ID as key and the identity with the key information as value.

5.68.14 MAIL_importGPGKey

Description: Imports an GPG key into the bunch of GPG keys.
Returns:

- : True on sucess otherwise false.

5.68.15 MAIL_deleteGPGKey

Description: Deletes an GPG key from the bunch of GPG keys.
Parameter:

- keyID: The ID of the GPG key to delete.

Returns:

- : True on sucess otherwise false.

5.68.16 MAIL_manageGPGKeysDialog

Description: Shows a dialog for importing and deleting GPG keys.

5.68.17 MAIL_sendMail

Description: Sends a mail, that may be GPG encrypted and contain an attachement via the cryptmail gateway.
Parameter:

- eMail: eMail address of the recipient.
- message: The message text.
- subject: The subject of the eMail.
- file: The file name (with full path) to attach or empty if no file should be attached.
- from: The sender's email address.

5.68.18 MAIL_cryptMailServer

Description: Server part for sending AES mails.

5.69 ./inc/massTools.php

Author: Hauke Goos-Habermann (HHabermann@pc-kiel.de)
Description: routines for mass installations

5.69.1 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.

5.69.2 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.

5.69.3 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

5.69.4 MASS_propertyKeys

Description: returns the keys for all properties

5.69.5 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

5.69.6 MASS_keyToI18N

Description: converts the property names to I18N names
Parameter:

- key: property name

5.69.7 MASS_I18NTokey

Description: converts the I18N names to property names
Parameter:

- key: property name

5.69.8 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

5.69.9 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

5.69.10 MASS_openDBFile

Description: opens a DB file
Parameter:

- fileName: name of the DB file

5.69.11 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.

5.69.12 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

5.69.13 MASS_closeDBFile

Description: closes the DB file.
Parameter:

- file: file pointer

5.69.14 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

5.69.15 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

5.69.16 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.

5.69.17 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

5.69.18 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

5.69.19 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)

5.69.20 MASS_generateNetmask

Description: generate netmasks from ip addresses via network class definitions.
Parameter:

- ip: the ip that shoulb be used to calculate the netmask

5.69.21 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

5.69.22 MASS_saveGeneratorOptions

Description: saves all geneator options to EGKparams
Parameter:

- EGKparams: enter generate keep parameters, that hold information about handling of the properties

5.69.23 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

5.69.24 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

5.69.25 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

5.69.26 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

5.70 ./inc/menu.php

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

5.70.1 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

5.70.2 MENU_startGroup

Description: shows the start of a menu group
Parameter:

- name: name of the menu group

5.70.3 MENU_endGroup

Description: shows the end of a menu group

5.71 ./inc/message.php

Author: Hauke Goos-Habermann (HHabermann@pc-kiel.de)
Description: function to show an info box

5.71.1 MSG_showMessageBoxPlaceholder

Description: Shows a placeholder for message boxes at the position of execution.

5.71.2 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.

5.71.3 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.

5.71.4 MSG_getm23DevelopmentBlog

Description: Shows the m23 development blog.
Parameter:

- width: Width of the box
- refreshTime: The time in minutes the file is downloaded again.

5.71.5 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.

5.71.6 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

5.71.7 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

5.71.8 MSG_showEmergencyError

Description: Shows the ermergency error message with a JavaScript "back" button and stops the script afterwards.
Parameter:

- message: the text for the info message
- language: two character language description (e.g. de, en, fr,...)
- width: width of the box

5.71.9 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

5.71.10 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)

5.71.11 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.

5.71.12 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.

5.71.13 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.

5.71.14 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,...)

5.71.15 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,...)

5.71.16 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

5.71.17 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.

5.72 ./inc/messageReceive.php

Author: Hauke Goos-Habermann (HHabermann@pc-kiel.de)
Description: Functions for handling messages sent by m23 clients.

5.72.1 MSR_decodeMessage

Description: checks the type of the message and cals the right procedure

5.72.2 MSR_curDynIPCommand

Description: Generates the commands to transfer the current dynamic IP of the m23 client to the server.
Parameter:

- return: true, if the commands should be returned, false when shown.

5.72.3 MSR_buildPoolFromClientDebsCMD

Description: Generates the commands to start the process for building a CPoolFromClientDebsGUI object and to start the download of the packages from the client to the server and the creation of the pool.

5.72.4 MSR_buildPoolFromClientDebs

Description: Builds a CPoolFromClientDebsGUI object and starts the download of the packages from the client to the server and the creation of the pool.

5.72.5 MSR_statusBarInc

Description: Increments the status bar percent by a given amount for the current client and for the "installStatus" status bar.
Parameter:

- percent: Percent value of the current job.

5.72.6 MSR_statusBarIncCommand

Description: Command to increment the status bar percent by a given amount for the current client and for the "installStatus" status bar.
Parameter:

- percent: Percent value of the current job.

5.72.7 MSR_statusBarCommand

Description: Command to set a new percent value and/or new status text for the current client and for the "installStatus" status bar.
Parameter:

- 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).

5.72.8 MSR_genericSendCommand

Description: Generates a generic command for sending information from the client to the server.
Parameter:

- type: Type of the message understood by MSR_decodeMessage
- params: Parameters to send with POST to the server in the form of "var1=val1&var2=val2&var3=val3..."
- wgetParams: Extra parameters for wget.

5.72.9 MSR_statusBar

Description: Sets new percent value and/or new status text.
Parameter:

- percent: Percent value to write into the DB (may be empty, 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 empty, if it should not be changed).

5.72.10 MSR_curDynIP

Description: Sets the current IP of a client with dynamic IP.
Parameter:

- curIP: The current IP.

5.72.11 MSR_copyClientPackageStatus

Description: Writes a sent full or difference package status file to the correct directory.
Parameter:

- diffType: "full" for a complete copy of the file and "diff" for a difference copy.

5.72.12 MSR_importDiffFile

Description: Writes a sent full or difference file to the destination.
Parameter:

- outFile: The file name (with full path) on the server to store the file.
- diffType: "full" for a complete copy of the file and "diff" for a difference copy.

5.72.13 MSR_CommandCopyClientPackageStatus

Description: Generates commands to transfer the package status file from the client to the server.

5.72.14 MSR_copyDiffFileFromClient

Description: Sends a file from the client to the server and tries to send only the changes towards an existing file on the server.
Parameter:

- clientFile: Name of the file on client side (with full path)
- serverFile: Name of the file on server side (with full path)
- type: Type of the message understood by MSR_decodeMessage (extended by the type of transfer XXXfull or XXXdiff)

5.72.15 MSR_markm23normalAsDone

Description: Marks comming m23normal install jobs as done if the packages they would install are already installed. This may happen when a package with dependencies is installed.
Parameter:

- $_POST['data']: List of new installed packages.

5.72.16 MSR_decodeClientSideBase64

Description: Decodes the slightly modificated base64 input stream created from MSR_clientSideBase64Encode.
Parameter:

- in: Base64 encoded input string.
- md5: md5 sum of the original input file.

Returns:

- Decoded (maybe binary) string.

5.72.17 MSR_clientSideBase64Encode

Description: Encodes a given file to (a slighly different, + is converted to - for sending it as post variable via wget) base64 format and appends the output to statusdata.post. There are three methods for generating the base64 output. First and second the native uuencode and base64 tools that are very fast and third a plattform idependent implementation of base64 encode in AWK taken from the HylaFAX package.
Parameter:

- fileName: name of the file

Returns:

- Commands for encoding the file.

5.72.18 MSR_genSendBinayFileCommand

Description: Sends a binary file to postMessage.php. In contrast to MSR_genSendCommand MSR_genSendBinayFileCommand is binary safe.
Parameter:

- fileName: Name of the file
- type: Type of the message understood by MSR_decodeMessage

5.72.19 MSR_m23ImagerMBR

Description: Saves the received master boot record to a file after cecking the MD5.

5.72.20 MSR_m23ImagerSize

Description: Sets the size of the decompressed image in its file name.

5.72.21 MSR_getm23clientIDCMD

Description: returns the BASH code for storing the m23clientID in the BASH variable "varid"
Parameter:

- addChar: is the

5.72.22 MSR_importPackageStatus

Description: imports the data from a package status message to the database. pNr=package, vNr=version, sNr=status

5.72.23 MSR_importLog

Description: reads log data from the post data and store it in the DB

5.72.24 MSR_logCommand

Description: generates BASH code to send a log file to the server
Parameter:

- logFile: name of the logfile
- show: If set to true, the output is shown directly, if set to false, it is returned.

5.72.25 MSR_statusFileCommand

Description: generates the commands to send the package infos and package status file

5.72.26 MSR_importStatusFile

Description: reads status file data from the post data and store it under /m23/var/cache/clients/clientName/packageStatus

5.72.27 MSR_genSendCommand

Description: sends file to postMessage.php
Parameter:

- fileName: name of the file
- type: type of the message understood by MSR_decodeMessage
- show: If set to true, the output is shown directly, if set to false, it is returned.

5.72.28 MSR_clientChangeCommand

Description: sends the ID of a "m23changeClient" job to the server.
Parameter:

- id: the ID of the client job

5.72.29 MSR_clientChange

Description: executes changes of a "m23changeClient" job on the server.

5.72.30 MSR_partHwDataCommand

Description: generates the commands to send partition and hardware info

5.72.31 MSR_importPartHwData

Description: imports partition and hardware information

5.72.32 MSR_getClientSettingsCommand

Description: Generates a script to gather network, release, distribution, login, LDAP, NFS and kernel informations for client import on the client

5.72.33 MSR_clientSettings

Description: Imports the clients settings

5.72.34 MSR_setOnline

Description: Sets the online-state of a client.
Parameter:

- online: The new online-state.

5.72.35 MSR_sshHttpsStatus

Description: Sets the ping and advanced SSH/HTTPs available status.

5.72.36 MSR_setTimeStampForRebootClientAfterJobsIsNecessaryCMD

Description: Generates the commands to inform the m23 server that a reboot of the client is necessary because Debian packages, that were installed, required it.
Returns:

- BASH commands.

5.72.37 MSR_setTimeStampForRebootClientAfterJobsIsNecessary

Description: Updates the timestamp of the reboot request of the client, that became necessary because of newly installed Debian packages.

5.72.38 MSR_unsetTimeStampForRebootingClientIfNOTNecessaryCMD

Description: Generates the commands to inform the m23 server that a reboot of the client has been done, after Debian packages required it.
Returns:

- BASH commands.

5.72.39 MSR_unsetTimeStampForRebootingClientIfNOTNecessary

Description: Removes the timestamp of the reboot request of the client, that became necessary because of newly installed Debian packages.

5.73 ./inc/packageBuilder.php

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

5.73.1 PKGBUILDER_showDialog

Description: Shows a dialog for creating Debian packages from archives.

5.73.2 PKGBUILDER_showKeySelectionDialog

Description: Shows a dialog for choosing the GPG signing key for the extra packages.

5.73.3 PKGBUILDER_showUploadDialog

Description: Shows a dialog for uploading the tar files with checking of the file extension.

5.73.4 PKGBUILDER_signExtraDebsRelease

Description: Signs the Release file in the extraDebs directory as Release.gpg and InRelease.

5.73.5 PKGBUILDER_listFiles

Description: Shows a dialog of all files in EXTRA_DEBS_DIRECTORY with possibillity to create Debian packages from tar files and to delete files.

5.73.6 array_keysSearch

Description: Searches for a key in an associative array that matches a regular expression.
Parameter:

- array: Array to search in.

- expr: Regular expression for searching the keys (e.g. "/$^{B}UT\_action/")$. **Returns:**

- : The first found key that matches the expression or false if no matching key was found.

5.73.7 PKGBUILDER_tar2deb

Description: Converts a tar file (with bzip2 or gzip compression) into a Debian package.
Parameter:

- tarFile: Name of the tar file to convert or "false" if only the package index should be updated.

5.74 ./inc/packages.php

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

5.74.1 PKG_getLastUpgradeTime

Description: Gets the timestamp of the latest finished update job.
Parameter:

- clientName: Name of the client.

5.74.2 PKG_updateStartTime

Description: Updates the starting time of a job with the current timestamp.
Parameter:

- id: id for the selected job.

5.74.3 PKG_getDelayedJobsSQL

Description: Prepares an SQL statement to get information about delayed jobs of a client or its amount.
Parameter:

- clientName: Name of the client or false, if all clients that are delayed, should be found.
- maxAllowedDelay: The maximum tolerable amount of time since adding the job. Only jobs that exceed the limit are added to the array.
- select: If true, an SQL statement to get information about delayed jobs will be created, otherwise it's an SQL statement for getting the amount of deleyed jobs.
- status: Status of the packages to check for delays. If unset, waiting jobs, if false all, will be checked.
- package: Name of the packages to check or false to check all.

5.74.4 PKG_getDelayedJobs

Description: Gets an associative array with information about delayed jobs of a client.
Parameter:

- clientName: Name of the client.
- maxAllowedDelay: The maximum tolerable amount of time since adding the job. Only jobs that exceed the limit are added to the array.
- status: Status of the packages to check for delays. If unset, waiting jobs, if false all, will be checked.
- package: Name of the packages to check or false to check all.

5.74.5 PKG_getDelayedJobsAmount

Description: Gets the amount of delayed jobs of a client.
Parameter:

- clientName: Name of the client.
- maxAllowedDelay: The maximum tolerable amount of time since adding the job. Only jobs that exceed the limit are added to the array.

5.74.6 PKG_getClientsWithDelayedUpdateJobs

Description: Gets all clients that have delayed jobs.
Parameter:

- maxAllowedDelay: The maximum tolerable amount of time since adding the job. Only jobs that exceed the limit are added to the array.

5.74.7 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.

5.74.8 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.

5.74.9 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.

5.74.10 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.
- debianAndUbuntu: If set to true, groups for Debian and Ubuntu systems are added and the job chooses the groups when it is executed on the system.

5.74.11 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.

5.74.12 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.

5.74.13 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.

5.74.14 PKG_exportSelectedPackages

Description: Exports the wait4acc/selected packages of a client.
Parameter:

- client: Name of the client or empty.

5.74.15 PKG_getPackageStatusCSV

Description: Exports the packages of a client in tabulator separated CSV.
Parameter:

- client: Name of the client.

5.74.16 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).

5.74.17 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.

5.74.18 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.

5.74.19 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.

5.74.20 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.

5.74.21 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.

5.74.22 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.

5.74.23 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.

5.74.24 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.

5.74.25 PKG_OptionPageHeader2

Description: Starts the option page for debconf settings with all necessary options.
Parameter:

- title: the window title of the OptionPage

5.74.26 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.

5.74.27 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.

5.74.28 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.

5.74.29 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

5.74.30 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

5.74.31 PKG_closeSearch

Description: closes a search request
Parameter:

- file: file handle

5.74.32 PKG_getNextPackage

Description: fetches the next package
Parameter:

- file: file handle

5.74.33 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

5.74.34 PKG_listRecommendSubPackages

Description: returnes subpackages of a package
Parameter:

- cut: cuts the packages by $cut
- params: variable to write package names to

5.74.35 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.

5.74.36 PKG_addPackageSelection

Description: Adds one or multiple package selection(s) to the list of packages to install.
Parameter:

- client: name of client to install packages on
- packageSelection: Name of a single package selection or an array with multiple package selections 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.

5.74.37 PKG_addPackageSelectionSingle

Description: Adds a single package selection 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.

5.74.38 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

5.74.39 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.

5.74.40 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.

5.74.41 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

5.74.42 PKG_addNormalPackages

Description: adds normal packages to db
Parameter:

- amount: amount of selected packages
- client: name of client to install packages on

5.74.43 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.

Returns:

- Amount of jobs with changed priority.

5.74.44 PKG_changeInstallReasonSelectedPackages

Description: Changes the priority of selected wait4acc packages.
Parameter:

- amount: amount of selected packages
- client: name of client to install packages on
- newReason: The new reason to set.

5.74.45 PKG_rmSelectedPackages

Description: removes normal packages from db
Parameter:

- amount: amount of selected packages
- client: name of client to install packages on

Returns:

- Amount of removed jobs.

5.74.46 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

5.74.47 PKG_countSelectedpackages

Description: counts the preselected packages
Parameter:

- client: name of client to install packages on

5.74.48 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

5.74.49 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.

5.74.50 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.

5.74.51 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.

5.74.52 PKG_listSpecialpackages

Description: lists special packages matching a key
Parameter:

- key: search key

5.74.53 PKG_addSpecialPackages

Description: adds normal packages to db
Parameter:

- amount: amount of selected packages
- client: name of client to install packages on

5.74.54 PKG_getSpecialPackagePriority

Description: gets the priority of a special package
Parameter:

- package: name of package
- distr: name of the distribution

5.74.55 PKG_getSpecialPackageDescription

Description: gets the description of a special package
Parameter:

- package: name of package

5.74.56 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.

5.74.57 PKG_getPackageID

Description: get the id for a wait4acc job
Parameter:

- client: name of the client
- package: name of the package

5.74.58 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.

5.74.59 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

5.74.60 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

5.74.61 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

5.74.62 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

5.74.63 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

5.74.64 PKG_discardJobs

Description: discards all preselected jobs
Parameter:

- client: name of the client

5.74.65 PKG_discardJob

Description: discards a selected job
Parameter:

- client: name of the client
- package: name of package you want to discard

5.74.66 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

5.74.67 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

5.74.68 PKG_setClientPackageInstalledOK

Description: changes the status of a package to "installed ok"
Parameter:

- client: name of the client
- package: name of package

5.74.69 PKG_addShutdownPackage

Description: adds a shutdown package, but only if the client is NOT running. returns true, if a shutdown package is added
Parameter:

- client: name of the client

5.74.70 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

5.74.71 PKG_getAllParams

Description: gets all parameters of the parameters column of a clientjob
Parameter:

- packageID: the ID of the package

5.74.72 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

5.74.73 PKG_OptionPageHeader

Description: starts the option page with all necessary options
Parameter:

- title: the window title of the OptionPage

5.74.74 PKG_OptionPageRender

Description: renderes the layout of a OptionPage
Parameter:

- layout: for the definition of the layout array see the development guide

5.74.75 PKG_OptionPageTail

Description: generates the bottom of the OptionPage
Parameter:

- layout: for the definition of the layout array see the development guide

5.74.76 PKG_OptionPageSaveAlsParameters

Description: saves the entered values in the packagejobs params
Parameter:

- layout: for the definition of the layout array see the development guide

5.74.77 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

5.74.78 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)

5.74.79 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

5.74.80 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

5.74.81 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

5.74.82 PKG_updateSourcesListAtAllClients

Description: updates the sources.list at all clients using it
Parameter:

- sourcename: name of the sources.list that should be updated

5.74.83 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

5.74.84 PKG_removeFromJobList

Description: removes all jobs identified by the IDs in packageIDList
Parameter:

- packageIDList: the list of IDs of jobs to be deleted

5.74.85 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

5.74.86 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.

5.74.87 PKG_getHigestIDOfSpecialPackage

Description: Gets the highest ID of a special package.
Parameter:

- clientName: Name of the client.
- package: Name of the special package.

5.74.88 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

5.74.89 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

5.74.90 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

5.74.91 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).

5.74.92 PKG_getClientsWithWaitingJobs

Description: Gets all clients that have waiting jobs.
Returns:

- Array with all clients that have waiting jobs.

5.74.93 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).

5.74.94 PKG_countPackages

Description: counts all packages on a client
Parameter:

- clientName: name of the client

5.74.95 PKG_copyWait4accPackagesToClient

Description: copies the waiting jobs from one client to another
Parameter:

- from: the source client
- to: the destination client

5.74.96 PKG_copyPackagesToClient

Description: copies all packages 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

5.74.97 PKG_remNormalPackages

Description: adds normal deinstallation jobs to db
Parameter:

- amount: amount of selected packages
- client: name of client to deinstall packages on

5.74.98 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

5.74.99 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

5.74.100 PKG_deletePackageselection

Description: delete all packages from package selection
Parameter:

- selectionName: name for the package selection

5.74.101 PKG_getAllPackageSelections

Description: returns all package selection names
Parameter:

- addEmpty: set to true to add an empty entry at the beginning.

5.74.102 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.

5.74.103 PKG_storablePackageSelectionsSelection

Description: Generates a storble multi selection with all package selections.
Parameter:

- selName: Name of the selection
- prefKey: Variable name of the preference the dialog element stands for.
- storePointer: Additional pointer to the variable where to store the entered value.

5.74.104 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.

5.74.105 PKG_getPackageParams

Description: gets the parameters for a selected package
Parameter:

- id: package ID

5.74.106 PKG_getClientbyPackageID

Description: gets the clientname that owns a selected package ID
Parameter:

- id: package ID

5.74.107 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)

5.74.108 PKG_getClientIDbyPackageID

Description: returns the ID of a client that owns a selected package ID
Parameter:

- id: package ID

5.74.109 PKG_getPackageParamsVar

Description: fetch the device for installation
Parameter:

- id: package ID
- var: name of variable you want to get the value of

5.74.110 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

5.74.111 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

5.74.112 PKG_getPackagesListMarker

Description: returns the string to mark client names to store packages

5.74.113 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

5.74.114 PKG_getDuplicatedIdenticalJobs

Description: Returns an array with a list of all jobs that are identical by package, priority, status and params.
Parameter:

- clientName: Name of the client.
- omitFirstID: Set to true if the first ID of each duplicate jobs block should be omitted.

Returns:

- Array with a list of all jobs that are identical by package, priority, status and params.

5.74.115 PKG_removeDuplicatedIdenticalJobs

Description: Removes all jobs that are identical by package, priority, status and params omitting the first job.
Parameter:

- clientName: Name of the client.

5.74.116 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

5.74.117 PKG_deletePackagesList

Description: deletes a packages list
Parameter:

- listName: name of the list to delete

5.74.118 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.

5.74.119 PKG_recountAllClientPackages

Description: Recounts all packages of a client and updates corresponding fields in the database.

5.75 ./inc/pdf.php

Author: Hauke Goos-Habermann (HHabermann@pc-kiel.de)
Description: Functions for generating PDF files

5.75.1 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.

5.75.2 PDF_showTableHeader

Description: Inits some values for starting a new PDF table.

5.75.3 PDF_showTableEnd

Description: Prints the PDF table.

5.75.4 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

5.75.5 PDF_output

Description: Shows the created PDF.

5.76 ./inc/plugin.php

Author: Hauke Goos-Habermann (HHabermann@pc-kiel.de)
Description: functions dealing with (de)installation of plugins and status information.

5.76.1 PLG_listMenuPlugins

Description: generate the menu entries for the plugins
Parameter:

- path: the path you want to scan for plugins

5.76.2 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

5.76.3 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

5.76.4 PLG_getPLGName

Description: gets the name of the plugin
Parameter:

- fileName: file name with whole path to the plugin file

5.76.5 PLG_getPLGPage

Description: gets the page of the plugin
Parameter:

- fileName: file name with whole path to the plugin file

5.76.6 PLG_getPLGAuthor

Description: gets the author of the plugin
Parameter:

- fileName: file name with whole path to the plugin file

5.76.7 PLG_getPLGUpdateURL

Description: gets the update address of the plugin
Parameter:

- fileName: file name with whole path to the plugin file

5.76.8 PLG_getPLGClientRequires

Description: gets the "client requires" packages of the plugin
Parameter:

- fileName: file name with whole path to the plugin file

5.76.9 PLG_getPLGVersion

Description: gets the version of the plugin
Parameter:

- fileName: file name with whole path to the plugin file

5.76.10 PLG_showDownloadStatus

Description: shows the status of the plugin download
Parameter:

- fileName: file name with whole path to the plugin file

5.76.11 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

5.76.12 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

5.76.13 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

5.76.14 PLG_realInstall

Description: does the real installation
Parameter:

- tempDir: where to find the extracted files of the plugin

5.76.15 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.

5.76.16 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.

5.76.17 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.

5.76.18 PLG_showPluginOverview

Description: shows a overview of all plugins

5.76.19 PLG_install

Description: deletes a plugin
Parameter:

- name: name of the plugin

5.76.20 PLG_getUpdateFile

Description: gets the update info file
Parameter:

- name: name of the plugin

5.76.21 PLG_update

Description: initalizes the update, shows information about the plugin update
Parameter:

- name: name of the plugin

5.76.22 PLG_listInfofile

Description: lists information of a plugin update file
Parameter:

- name: name of the plugin

5.76.23 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.

5.77 ./inc/pool.php

Author: Hauke Goos-Habermann (HHabermann@pc-kiel.de)
Description: functions for administrating package pools

5.77.1 POOL_selectPoolType

Description: shows buttons for selecting the type of pool and returns the pressed button

5.77.2 POOL_getPools

Description: returns an array with all pool names

5.77.3 POOL_showLoadDeleteCreate

Description: shows a dialog for loading, deleting and creating a pool
Parameter:

- poolName: name of the pool

5.77.4 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

5.77.5 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

5.77.6 POOL_getProperty

Description: returns the contents of a property file
Parameter:

- poolName: name of the pool
- property: name of the pool property

5.77.7 POOL_delete

Description: deletes a pool
Parameter:

- poolName: name of the pool

5.77.8 POOL_showReadCD

Description: shows a dialog for copying the CD contents to the pool
Parameter:

- poolName: name of the pool

5.77.9 POOL_readCD

Description: copys the CD contents to the pool
Parameter:

- poolName: name of the pool
- mountPoint: the mount point of the CD drive

5.77.10 POOL_createExtendedPackageIndex

Description: creates the Packages* index files for the pool
Parameter:

- poolName: name of the pool

5.77.11 POOL_showCreatePackageIndex

Description: shows information (status of the Packages* generation, sources.list) about the currently generated pool

5.77.12 POOL_getSize

Description: returns the size of a pool in MB
Parameter:

- poolName: name of the pool

5.77.13 POOL_getDir

Description: returns the directory of the pool
Parameter:

- poolName: name of the pool

5.77.14 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

5.77.15 POOL_showDownloadStatus

Description: shows the package download status of a pool
Parameter:

- poolName: name of the pool

5.77.16 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

5.77.17 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
-

5.77.18 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

5.77.19 POOL_showSourcesList

Description: Shows the sources list of a selected package source.
Parameter:

- poolName: name of the pool

5.78 ./inc/preferences.php

Author: Hauke Goos-Habermann (HHabermann@pc-kiel.de)
Description: functions to save and load preferences for client setup.

5.78.1 PREF_preferenceLoadManagerHandler

Description: Executes loading and deletion of preferences after pressing the according buttons and defines the buttons for PREF_showPreferenceManager();

5.78.2 PREF_preferenceSaveManagerHandler

Description: Executes the saving of preferences.

5.78.3 PREF_showPreferenceManager

Description: Shows a dialog to load and delete existing preferences and to create new preferences.

5.78.4 PREF_saveAllPreferenceValues

Description: Saves all values of a session into the preference.

5.78.5 PREF_loadAllPreferenceValues

Description: Loads all values of a preference into the session.

5.78.6 PREF_getClientPreferences

Description: List all preferences, as HTML selections or associative array.
Parameter:

- default: The name of the preference to list first
- generateHTML: If enabled the preference names will be generated as a HTML option list. If disabled an array with the preference names as key and value will be returned.
- showHTML: If enabled the generated as a HTML option list will be shown, if disabled, it will be returned.

Returns:

- Array with the preference names or nothing on enabled showHTML.

5.78.7 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

5.78.8 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

5.78.9 PREF_delete

Description: deletes a preference
Parameter:

- name: the name of the preference

5.78.10 PREF_exists

Description: checks if a preference with the selected name exists
Parameter:

- name: the name of the preference

5.78.11 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

5.78.12 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

5.79 ./inc/remotevar.php

Author: Hauke Goos-Habermann (HHabermann@pc-kiel.de)
Description: functions for saving and loading serverside variables.

5.79.1 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

5.79.2 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

5.79.3 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

5.79.4 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

5.79.5 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

5.79.6 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

5.79.7 RMV_rm

Description: removes a variable for the ip of the calling client
Parameter:

- var: name of the variable to get the value from

5.79.8 RMV_rm_old

Description: removes all vars older than $time seconds
Parameter:

- time: time in seconds

5.80 ./inc/scredit.php

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

5.80.1 SCREDIT_correctScriptFilename

Description: Corrects the given filename to have it a valid prefix and suffix.
Parameter:

- filename: The filename to check and correct.

5.80.2 SCREDIT_showEditor

Description: Shows a script editor with syntax highlighting if JavaScript is enabled or a normal textarea input dialog.

5.80.3 SCREDIT_newScriptTemplate

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

- Text of the script template.

5.81 ./inc/serverBackup.php

Author: Hauke Goos-Habermann (HHabermann@pc-kiel.de)
Description: Functions for handling m23 server backups.

5.81.1 SERVERBACKUP_getBackupConfiguration

Description: Gets the server backup configuration.
Returns:

- Associative array with the variable name as key and its value as value.

5.81.2 SERVERBACKUP_storeBackupConfiguration

Description: Stores the server backup configuration.
Parameter:

- conf: Associative array with the configuration values with the variable name as key and its value as value.

5.81.3 SERVERBACKUP_showConfigurationDialog

Description: Shows a dialog for configuring the server backup.

5.81.4 SERVERBACKUP_runBackupNowDialog

Description: Shows a dialog for starting the server backup manually at once.

5.81.5 SERVERBACKUP_getBackupList

Description: Generates a list of existing server backups.
Returns:

- Associative array with information about sizes and dates of the backups.

5.81.6 SERVERBACKUP_rmBackup

Description: Removes server backup.
Parameter:

- name: Name of the backup (that is a in form of YYYY-MM-DD-HH-MM)

5.81.7 SERVERBACKUP_backupOverviewDialog

Description: Shows a dialog with overview of all existing server backups with possibility for deletion.

5.82 ./inc/server.php

Author: Hauke Goos-Habermann (HHabermann@pc-kiel.de)
Description: functions for getting information from the server

5.82.1 SERVER_getPublicSSHKey

Description: Gets the public SSH (root) key of the m23 server.
Returns:

- Public SSH (root) key of the m23 server.

5.82.2 SERVER_setFusionDirectoryPassword

Description: Sets the FusionDirectory password for the admin account.
Parameter:

- fdAdminPassword: FusionDirectory admin password

5.82.3 SERVER_getFusionDirectoryPassword

Description: Gets the FusionDirectory password for the admin account.
Returns:

- FusionDirectory admin password

5.82.4 SERVER_getShowClientSerialColumn

Description: Determines, if the client's serial should be shown in the client overview.
Returns:

- : true, if the serial should be invisible, false otherwise.

5.82.5 SERVER_setShowClientSerialColumn

Description: En/Disables showing client's serial in the client overview.
Parameter:

- enabled: true, if the serial should be invisible, false otherwise.

5.82.6 SERVER_getShowClientMACColumn

Description: Determines, if the client's MAC should be shown in the client overview.
Returns:

- : true, if the MAC should be invisible, false otherwise.

5.82.7 SERVER_setShowClientMACColumn

Description: En/Disables showing client MACs in the client overview.
Parameter:

- enabled: true, if the MAC should be invisible, false otherwise.

5.82.8 SERVER_getShowClientIPColumn

Description: Determines, if the client's IP should be shown in the client overview.
Returns:

- : true, if the IP should be invisible, false otherwise.

5.82.9 SERVER_setShowClientIPColumn

Description: En/Disables showing client IPs in the client overview.
Parameter:

- enabled: true, if the IP should be invisible, false otherwise.

5.82.10 SERVER_getMakePasswordsInvisibleEnabled

Description: Determines, if passwords should be made invisible (eg. in the Control center).
Returns:

- : true, if passwords should be invisible, false otherwise.

5.82.11 SERVER_setMakePasswordsInvisibleEnabled

Description: Enables making passwords invisible (eg. in the Control center).
Parameter:

- enabled: true, if passwords should be invisible, false otherwise.

5.82.12 SERVER_set2xPasswordDialogEnabled

Description: Enables using two passwords edit lines with obscured passwords instead of showing the password in plaintext.
Parameter:

- enabled: true, if two passwords edit lines with obscured passwords should be used, false otherwise

5.82.13 SERVER_get2xPasswordDialogEnabled

Description: Determines, if two passwords edit lines with obscured passwords should be used.
Returns:

- : true, if two passwords edit lines with obscured passwords should be used, false otherwise

5.82.14 SERVER_getExportIntoClientreporting

Description: Determines, if the information should be exported to the ‘clientreporting‘ table.
Returns:

- : true, if the information should be exported to the ‘clientreporting‘ table, otherwise false.

5.82.15 SERVER_setExportIntoClientreporting

Description: Sets, if the information should be exported to the ‘clientreporting‘ table.
Parameter:

- active: true, if the information should be exported to the ‘clientreporting‘ table, otherwise false.

5.82.16 SERVER_getShowClientLastUpgradeColumn

Description: Determines, if the time of the last finished update job should be shown in the client's job overview.
Returns:

- : true, if the time of the last finished update job should be shown, otherwise false.

5.82.17 SERVER_setShowClientLastUpgradeColumn

Description: Sets, if the time of the last finished update job should be shown.
Parameter:

- active: true, if the time of the last finished update job should be shown, otherwise false.

5.82.18 SERVER_setWarnWhenClientRebootsRequestedByPackagesAreDelayed

Description: Sets the tolerable amount of time between a reboot requested by a Debian package and the actual reboot.
Parameter:

- maxAllowedDelay: Tolerable amount of time (in minutes) between a reboot requested by a Debian package and the actual reboot. or 0, to disable the warning.

5.82.19 SERVER_getWarnWhenClientRebootsRequestedByPackagesAreDelayed

Description: Gets the tolerable amount of time between a reboot requested by a Debian package and the actual reboot.
Returns:

- : Tolerable amount of time (in minutes) between a reboot requested by a Debian package and the actual reboot. or 0, to disable the warning.

5.82.20 SERVER_getShowTimeInformationOnJobs

Description: Determines, if time information should be shown in the client's job overview.
Returns:

- : true, if the information should be shown, otherwise false.

5.82.21 SERVER_setShowTimeInformationOnJobs

Description: Sets, if time information should be shown in the client's job overview.
Parameter:

- active: true, if the information should be shown, otherwise false.

5.82.22 SERVER_importGPGPackageSignKey

Description: Imports the m23 GPG package sign key.

5.82.23 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.

5.82.24 SERVER_getWarnWhenUpdateJobsAreDelayed

Description: Gets the tolerable amount of running/waiting time to finish an update job.
Returns:

- : Tolerable amount of running/waiting time (in minutes) to finish a job or 0, to disable the warning.

5.82.25 SERVER_setWarnWhenUpdateJobsAreDelayed

Description: Sets the tolerable amount of running/waiting time to finish an update job.
Parameter:

- maxAllowedDelay: The maximum tolerable amount of running/waiting time (in minutes) to finish since adding it or 0, to disable.

5.82.26 SERVER_getWarnWhenJobsAreDelayed

Description: Gets the tolerable amount of running/waiting time to finish a job.
Returns:

- : Tolerable amount of running/waiting time (in minutes) to finish a job or 0, to disable the warning.

5.82.27 SERVER_setWarnWhenJobsAreDelayed

Description: Sets the tolerable amount of running/waiting time to finish a job.
Parameter:

- maxAllowedDelay: The maximum tolerable amount of running/waiting time (in minutes) to finish since adding it or 0, to disable.

5.82.28 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.

5.82.29 SERVER_isSSLCertCheckDisabled

Description: Determines, if the SSL certificate check is disabled globally for all clients.
Returns:

- : true, if the check is disabled otherwise false.

5.82.30 SERVER_isLiveLogDisabled

Description: Determines, if live logging should be disabled globally for all clients.
Returns:

- : true, if live logging should be disabled globally for all clients otherwise false.

5.82.31 SERVER_setLiveLogDisabled

Description: Sets/Unsets live disabling of logging globally for all clients.
Parameter:

- disabled: true, if live logging should be disabled globally for all clients otherwise false.

5.82.32 SERVER_isInstallReasonEnabled

Description: Determines, if the install reason feature is enabled
Returns:

- : true, if the install reason feature is enabled, false otherwise

5.82.33 SERVER_setRebootClientAfterJobsIfNecessary

Description: Sets/Unsets rebooting clients, if a reboot is necessary after jobs are finished.
Parameter:

- disabled: true, if clients should be rebooted after the jobs are finished and a reboot is necessary, otherwise false.

5.82.34 SERVER_isRebootClientAfterJobsIfNecessary

Description: Determines, if clients should be rebooted after the jobs are finished and a reboot is necessary, otherwise false.
Returns:

- : true, if clients should be rebooted after the jobs are finished and a reboot is necessary, otherwise false.

5.82.35 SERVER_setHandleAllClientsAsDynamic

Description: Sets/Unsets handling of all m23 clients as dynamic clients.
Parameter:

- enabled: true, if all m23 clients should be handled as dynamic clients, otherwise false.

5.82.36 SERVER_isHandleAllClientsAsDynamicEnabled

Description: Determines, if handling of all m23 clients as dynamic clients is enabled.
Returns:

- : true, if handling of all m23 clients as dynamic clients is enabled, otherwise false.

5.82.37 SERVER_setm23ServerIncudedInSourcesListDisabled

Description: Sets/Unsets disabling of including the m23 server into all client's sources.lists.
Parameter:

- disabled: true, if the m23 server should not be included into all client's sources.lists otherwise false.

5.82.38 SERVER_ism23ServerIncudedInSourcesListDisabled

Description: Determines, if including the m23 server into all client's sources.lists should be disabled.
Returns:

- : true, if including the m23 server into all client's sources.lists should be disabled, otherwise false.

5.82.39 SERVER_setInstallReasonEnabled

Description: Enables the install reason feature
Parameter:

- enabled: true, if the install reason feature shall be enabled, false otherwise

5.82.40 SERVER_isUpdatePackageInfosDisabled

Description: Determines, if the updatePackageInfos job should NOT be added to any clients.
Returns:

- : true, if the updatePackageInfos job should NOT be added to any clients otherwise false.

5.82.41 SERVER_setUpdatePackageInfosDisabled

Description: Sets/Unsets if the updatePackageInfos job should NOT be added to any clients.
Parameter:

- disabled: true, if the updatePackageInfos job should NOT be added to any clients otherwise false.

5.82.42 SERVER_isClientSshHttpsStatusEnabled

Description: Determines, if the SSH (server -> client) and the HTTPs (client -> server) status of all clients should be detected and shown.
Returns:

- : true, if the SSH and the HTTPs status of all clients should be detected and shown.

5.82.43 SERVER_setClientSshHttpsStatusEnabled

Description: Sets/Unsets if the SSH (server -> client) and the HTTPs (client -> server) status of all clients should be detected and shown.
Parameter:

- active: true, if the SSH and the HTTPs status of all clients should be detected and shown.

5.82.44 SERVER_isClientOnlineStatusEnabled

Description: Determines, if the online status of all clients should be detected and shown.
Returns:

- : true, if the online status of all clients should be detected and shown.

5.82.45 SERVER_startUpdateClientOnlineInDBBackgroundJob

Description: Starts a background job to update the online status of the clients cyclically.

5.82.46 SERVER_setClientOnlineStatusEnabled

Description: Sets/Unsets if the online status of all clients should be detected and shown.
Parameter:

- active: true, if the online status of all clients should be detected and shown.

5.82.47 SERVER_setServerBoolSetting

Description: Sets a bool value of a server setting.
Parameter:

- : var: Name of the setting.
- : val: true or false.

5.82.48 SERVER_getServerBoolSetting

Description: Gets the boolean value of a server setting.
Parameter:

- : var: Name of the setting.

Returns:

- : true or false.

5.82.49 SERVER_setServerSetting

Description: Sets the value of a server setting.
Parameter:

- : var: Name of the setting.
- : val: Value of the setting.

5.82.50 SERVER_getServerSetting

Description: Gets the value of a server setting.
Parameter:

- : var: Name of the setting.

Returns:

- : Value the setting.

5.82.51 SERVER_getServerIntSetting

Description: Gets the integer value of a server setting.
Parameter:

- : var: Name of the setting.
- : default: Default value, if the server side variable is not set or the tored value is too low
- : minimum: Minumum allowed value.

Returns:

- : Value the setting or default, if no value is stored or it's invalid.

5.82.52 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.

5.82.53 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.

5.82.54 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.

5.82.55 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.

5.82.56 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.

5.82.57 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.

5.82.58 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.

5.82.59 SERVER_getPublicSSHKeyOfm23Server

Description: Returns the public SSH key of the m23 server.
Returns:

- : Public SSH key of the m23 server.

5.82.60 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.

5.82.61 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.

5.82.62 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.

5.82.63 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.

5.82.64 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.

5.82.65 SERVER_dynamicIPWarn

Description: Shows an error message if the m23 server has a dynmic IP address.
Returns:

- : false, if the IP address is static.

5.82.66 SERVER_tmpNotWritable

Description: Shows an error message if /tmp is not writable.
Returns:

- : false, if /tmp is writable.

5.82.67 SERVER_rootFreeSpace

Description: Shows an error message if the free space of the root partition is low.
Returns:

- : false, if there is enough space.

5.82.68 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)

5.82.69 SERVER_checkPackageInstalled

Description: checks if a certain package is installed
Parameter:

- pkgName: the name of the package

5.82.70 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.

5.82.71 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.

5.82.72 SERVER_installTool

Description: installs a tool on the server
Parameter:

- pkgName: name of the software package

Returns:

- : true on successfully execution otherwise false.

5.82.73 SERVER_installToolInBackground

Description: Installs a tool on the server in background.
Parameter:

- pkgName: name of the software package

5.82.74 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

5.82.75 SERVER_apacheInfo

Description: returnes an information string for the Apache server

5.82.76 SERVER_mysqlInfo

Description: returnes an information string for the MySQL server

5.82.77 SERVER_dhcpInfo

Description: returnes an information string for the DHCP server

5.82.78 SERVER_LDAPInfo

Description: Returnes an information string for the LDAP server.

5.82.79 SERVER_programmStatusTableHeader

Description: shows the header of the table needed for the programm status lines

5.82.80 SERVER_waitForLock

Description: Cyclically and randomly waits until a lock file isn't existing and creates it at once.
Parameter:

- lockFile: name of the lock file.

Returns:

- true, if the lock file could be created, otherwise false.

5.82.81 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.
- redirectStdErr: Set to true if stderr should be redirected and not shown on in the console.
- bashLock: Set to false to use the PHP locking with randomly waiting instead of the locking inside the BASH script.

5.82.82 SERVER_runningInBackground

Description: Returns "true" if a lock file for a given job name is existing.
Parameter:

- jobName: name of the job

5.82.83 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.

5.82.84 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.

5.82.85 SERVER_deleteFile

Description: Deletes a file from the server.
Parameter:

- fileName: Name of the file to delete.

Returns:

- : true on successfully execution otherwise false.

5.82.86 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

5.82.87 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.

5.82.88 SERVER_overrideServerIP

Description: Overrides the (autodetected) IP of the m23 server.
Parameter:

- ip: New server IP to show to the m23 clients.

Returns:

- true on successfully setting and false otherwise.

5.82.89 SERVER_unoverrideServerIP

Description: Removes the file for overriding the IP of the m23 server.
Parameter:

- ip: New server IP to show to the m23 clients.

Returns:

- true on successfully deleting and false otherwise.

5.82.90 SERVER_isOverrideServerIPFilePresent

Description: Checks, if the file for overriding the IP of the m23 server is present.
Returns:

- true, if the file for overriding the IP of the m23 server is present and false otherwise.

5.82.91 SERVER_Dialog_overrideServerIP

Description: Dialog for (un)overriding the IP of the m23 server.
Returns:

- HTML code with the dialog elements.

5.82.92 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.

5.82.93 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

5.82.94 SERVER_delEtcHosts

Description: Deletes a host entry from /etc/hosts and /etc/backuppc/hosts
Parameter:

- hostname: name of the host to delete

5.82.95 SERVER_getInstallationMedium

Description: Tries to figure out how the m23 server was installed
Returns:

- : CD, Internet or Unknown source.

5.82.96 SERVER_getOS

Description: Returns the version string of the distribution.
Returns:

- : version string.

5.82.97 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.

5.82.98 SERVER_checkDiskFree

Description: Reports the free space of all mounted media.
Returns:

- : Output of "df".

5.82.99 SERVER_checkRunInVM

Description: Checks if the m23 server is executed in a virtual machine or on native hardware.
Returns:

- : VMWare, VirtualBox or native.

5.82.100 SERVER_checkKernel

Description: Returns the kernel information string of "uname -a".
Returns:

- : Kernel information string.

5.82.101 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)
- user: User name to create the directory for. If no name is given, the directory will be created for the Apache user.

5.82.102 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.

5.83 ./inc/sourceslist.php

Author: Hauke Goos-Habermann (HHabermann@pc-kiel.de)
Description: function to generate the sources.list for the client

5.83.1 SRCLST_getExportedListNames

Description: Generates an array with all sources lists that are exported by /mdk/bin/exportDBsourceslist.php.
Returns:

- : Array with all sources lists that are exported by /mdk/bin/exportDBsourceslist.php.

5.83.2 SRCLST_getImportGPGKeyBASH

Description: Returns BASH code to import GPG APT keys from the URLs that are given in the importGPGKey paramters in a sources list.
Parameter:

- sourceName: The name of the package source list

Returns:

- : BASH code to import GPG APT keys.

5.83.3 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), ...)

5.83.4 SRCLST_getAppendToFile

Description: Returns appendToFile 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), ...)

5.83.5 SRCLST_getXToFile

Description: Returns paramters of a given parameter type 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), ...)

5.83.6 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.

5.83.7 SRCLST_genList

Description: generates the sources.list file for the client
Parameter:

- clientName: the name of the client

5.83.8 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.

5.83.9 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

5.83.10 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

5.83.11 SRCLST_genSelection

Description: generates a HTML selection with the names of all 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

5.83.12 SRCLST_storableSelection

Description: Generates a storable HTML selection with the names of all package sources.
Parameter:

- htmlName: The name of the selection
- distr: The distribution the sources list is for or "*" for all distributions
- prefKey: Variable name of the preference the dialog element stands for.
- storePointer: Additional pointer to the variable where to store the entered value.

Returns:

- Name of selected sources list.

5.83.13 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

5.83.14 SRCLST_loadSourceListFromDB

Description: loads and returnes the the package source list from the DB.
Parameter:

- name: the name of the package source list

5.83.15 SRCLST_sourceListExists

Description: Checks, if a named sources list exists.
Parameter:

- name: the name of the package source list

Returns:

- true, if the package source list exists, otherwise false.

5.83.16 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

5.83.17 SRCLST_getDescription

Description: returnes the the package source description
Parameter:

- name: the name of the package source list

5.83.18 SRCLST_delete

Description: deletes package source
Parameter:

- name: the name of the package source list

5.83.19 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

5.83.20 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.

5.83.21 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.

5.83.22 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.

5.83.23 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

5.83.24 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.

5.83.25 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.

5.83.26 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.

5.83.27 SRCLST_getMirror

Description: returns the mirror from the sources list
Parameter:

- sourceName: the name of the package source list

Returns:

- URL to the mirror

5.83.28 SRCLST_getDesktopList

Description: returnes an array with all supported desktops
Parameter:

- sourceName: the name of the package source list

5.83.29 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

5.83.30 SRCLST_storableDesktopsSelection

Description: Generates a storable HTML selection with the names of all desktops.
Parameter:

- htmlName: The name of the selection
- sourceName: the name of the package source list
- prefKey: Variable name of the preference the dialog element stands for.
- storePointer: Additional pointer to the variable where to store the entered value.

Returns:

- Name of selected desktop.

5.83.31 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.

5.83.32 SRCLST_getListnamesWithEfiSupport

Description: Gets a list with all sources lists that support EFI.
Returns:

- Array with all sources lists that support EFI.

5.83.33 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.

5.83.34 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.

5.83.35 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.

5.83.36 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.

5.83.37 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.

5.83.38 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
- showSupportedUserInterfacesList: if it is set, the list with the supported GUIs will be shown.

5.83.39 SRCLST_getListnames

Description: Returns an array that contains all sourceslist names
Parameter:

- distr: the distribution the sources list is for or "*" for all distributions

5.83.40 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

5.83.41 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

5.83.42 SRCLST_possiblem23debsMirrors

Description: Returns an array with mirrors for m23 debs.
Returns:

- Array with mirrors for m23 debs.

5.83.43 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.

5.83.44 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.

5.84 ./inc/ucs.php

Author: Hauke Goos-Habermann (HHabermann@pc-kiel.de)
Description: Univention corporate server specific functions.

5.84.1 UCS_getPrefnameByClient

Description: Generates the preference name for an UCS client.
Parameter:

- client: Client name.

Returns:

- Preference name for an UCS client

5.84.2 UCS_addUCSClientTom23ClientPreferences

Description: Adds an UCS client to the client preferences in m23.
Parameter:

- client: Client name.
- mac: The MAC of the client.
- ip: The client's IP address.
- netmask: The client's netmask.
- gateway: The client's gateway.
- dns1: The client's first DNS server.

5.84.3 UCS_delUCSClientFromm23ClientPreferences

Description: Removes an UCS client from the client preferences in m23.
Parameter:

- client: Client name.

5.84.4 UCS_createNetworkObject

Description: Creates a new network object in the UCS LDAP:
Parameter:

- networkName: Name of the network object to create.
- netmaskBits: Amount of set bits in the netmask.
- networkIP: Network IP (ends with .0)

5.84.5 UCS_getGenericNetworkName

Description: Generates a generic name for a network object in UCS.
Parameter:

- netmaskBits: Amount of set bits in the netmask.
- networkIP: Network IP (ends with .0)

Returns:

- : Generic name for a network object in UCS.

5.84.6 UCS_ensureNetworkObjectExists

Description: Makes sure that the network object exists in the UCS.
Parameter:

- netmaskBits: Amount of set bits in the netmask.
- networkIP: Network IP (ends with .0)

Returns:

- : Generic name for a network object in UCS.

5.84.7 UCS_modifyClientIP

Description: Modifies the IP of a client in the UCS LDAP.
Parameter:

- client: Client name or CClient object.
- ip: The new client's IP address.
- netmask: Amount of set bits in the netmask or normal netmask.
- networkName: UCS network name.

5.84.8 UCS_getAllClientNamesLDAP

Description: Gets all client names stored in the LDAP of the UCS.
Returns:

- : Array with all client names stored in the LDAP of the UCS.

5.84.9 UCS_getUDMCompleteInfo

Description: Get complete information from the UCS's LDAP by udm tool for a given module.
Parameter:

- multiple: true, if there are multiple values with identical key names.
- DNcn: If set, blocks starting with "DN: cn=$DNcn," will be handled only.

Returns:

- : Array with associative arrays containing the information of all blocks. If multiple is active, the values are stored under [key][...] = $val otherwise under [key] = $val.

5.84.10 UCS_getUCSschoolRoles

Description: Get the UCSschoolRoles (eg. groups) of a client.
Parameter:

- client: Name of the client.

Returns:

- : Array with the UCSschoolRoles.

5.84.11 UCS_getOrganisationUnits

Description: Get information about all organisation units (eg. schools).
Returns:

- : Associative array with all organisation units. Name as key, name and display name as value.

5.84.12 UCS_getUDMInfo

Description: Get information from the UCS's LDAP by udm tool.
Parameter:

- udmModule: Name of the udm module (eg. networks/network or computers/linux).
- afterLines: Amount of lines with information after the line containing the keyword.
- keyWord: Keyword matching the first line to find the block with the wanted information.
- multiple: true, if there are multiple data sets with identical key names.

Returns:

- : Associative array with the information. Can be nested into a normal array, if $multiple = true.

5.84.13 UCS_getAllNetworkInformation

Description: Get information about all UCS networks.
Returns:

- : Array with nested associative arrays that are containing the information about each network.

5.84.14 UCS_getAllNetworkNames

Description: Get the names of all UCS networks.
Returns:

- : Associative arrays the names of all UCS networks as key and value.

5.84.15 UCS_networkObjectExists

Description: Checks, if a network object exists in the UCS.
Parameter:

- networkName: Network name.

Returns:

- : true, if a network object exists in the UCS, otherwise false.

5.84.16 UCS_getNetworkLDAPInfo

Description: Get information about a network stored in the UCS LDAP.
Parameter:

- networkName: Network name.

Returns:

- : Associative array with the information about the network.

5.84.17 UCS_getFirstElementFromDN

Description: Gets the first value of a distinguished name.
Parameter:

- dn: distinguished name.

Returns:

- : First value of a distinguished name.

5.84.18 UCS_getClientLDAPInfo

Description: Get information about a client stored in the UCS LDAP.
Parameter:

- client: Client name.

Returns:

- : Associative array with the information about the client.

5.84.19 UCS_addClient

Description: Adds a client to the UCS LDAP.
Parameter:

- client: Client name.
- mac: The MAC of the client.
- ip: The client's IP address.
- networkname: UCS network name.

5.84.20 UCS_delClient

Description: Removes a client from the UCS LDAP.
Parameter:

- client: Client name.

5.84.21 UCS_enableClientPXEBoot

Description: Activates PXE booting of a client that is in the UCS LDAP via the univention-dhcp.
Parameter:

- client: Client name.
- bootFilename: Image to boot over network (e.g. pxelinux.0).

5.84.22 UCS_disableClientPXEBoot

Description: Deactivates PXE booting of a client that is in the UCS LDAP via the univention-dhcp.
Parameter:

- client: Client name.

5.84.23 UCS_setClientDistrAndRelease

Description: Sets the distribution and the release of a client to the UCS LDAP.
Parameter:

- client: Client name.
- distr: Client's distribution.
- release: Client's distribution release.

5.84.24 UCS_openFirewallPort

Description: Opens a port on the UCS firewall.
Parameter:

- port: Port number to open.
- type: tcp or udp.

5.84.25 UCS_addLDAPUser

Description: Adds a posix account to the UCS LDAP server.
Parameter:

- 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

Returns:

- The output from udm after executing the parameters.

5.84.26 UCS_getEtc_ucr_master

Description: Gets the contents for /etc/univention/ucr_master on the client.

5.84.27 UCS_enableClientLDAP

Description: Enables LDAP authentification on the client on the UCS.

5.84.28 UCS_udmSuccessOrErrorMessage

Description: Takes the output from an udm command and checks for the return code (must be a single number in the last line). If the return code is non-zero, all lines above the last line are treated as error message and written to $errorMessage.
Parameter:

- ret: The complete message block with return code (must be a single number in the last line)
- errorMessage: The variable, the error message may be written to.

Returns:

- true, if the udm command returned a success return code (0), otherwise false.

5.84.29 UCS_getUsedIPs

Description: Gets a list with all used IPs managed or known by UCS.
Returns:

- Array with all used IPs managed or known by UCS.

5.85 ./inc/update.php

Author: Hauke Goos-Habermann (HHabermann@pc-kiel.de)
Description: functions for updating the server

5.85.1 UPDATE_doUpdate

Description: Makes sure the package repository is included and upgrades the m23 server.

5.85.2 UPDATE_running

Description: checks, if an update is running (returns true otherwise false)

5.85.3 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

5.85.4 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.

5.86 ./inc/vm.php

Author: Hauke Goos-Habermann (HHabermann@pc-kiel.de)
Description: Functions for managing virtual clients.

5.86.1 VM_isHost

Description: Check, if an m23 client is an VM host.
Parameter:

- clientName: Name of the m23 client.

Returns:

- false, if no VM host software is installed on the client otherwise the number of VM software.

5.86.2 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.

5.86.3 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)

5.86.4 VM_isCloudStackClient

Description: Checks, if the client is run in CloudStack
Parameter:

- clientname: CloudStack name of the instance / name of the m23 client.

Returns:

- true, when the client is run in CloudStack otherwise false.

5.86.5 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.

5.86.6 VM_CloudStackConfigGUI

Description: Shows a dialog for editing the CloudStack config file and uploading the m23 client ISO.

5.86.7 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.

5.86.8 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

5.86.9 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

5.86.10 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

5.86.11 VM_CloudStackSendSetVisualURL

Description: Sends the visual URL (current client ip:22) to the m23 server, if run under CloudStack.

5.86.12 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

5.86.13 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

5.86.14 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)

5.86.15 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

5.86.16 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

5.86.17 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

5.86.18 VM_CloudStack_getServerIP

Description: Gets the external m23 server IP if the m23 server is run as CloudStack VM.
Returns:

- External m23 server IP.

5.86.19 VM_GUIstepCreateCloudStackVM

Description: Shows a dialog to create a new VM in CloudStack.

5.86.20 VM_CloudStack_available

Description: Checks, if the CloudStack configuration file is included and contains the needed constants.
Returns:

- true, if the CloudStack are present.

5.86.21 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.

5.86.22 VM_CloudStack_getVersion

Description: Gets the version of CloudStack.
Returns:

- CloudStack version.

5.86.23 VM_shutdownAndDisableNetbootAfterInstall

Description: Reboots an VM and disables network booting.
Parameter:

- vmname: Name of the VM.

5.86.24 VM_rebootAndActivateNetboot

Description: Reboots an VM and activates network booting.
Parameter:

- vmname: Name of the VM.

5.86.25 VM_shutdownAndDisableNetbootAfterInstall

Description: Shuts down an VM and disables network booting.
Parameter:

- vmName: Name of the VM.

5.86.26 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.

5.86.27 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.

5.86.28 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.

5.86.29 VM_setVBoxAddonAsDefault

Description: Sets a choosen VirtualBox addition package version as default.
Parameter:

- version: Version number of the VirtualBox addition to set as default.

5.86.30 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.

5.86.31 VM_getVBoxAddonDefaultVersion

Description: Gets the version number of the VirtualBox addition package.
Returns:

- The default version of the VirtualBox addition package.

5.86.32 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.

5.86.33 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.

5.86.34 VM_VBOXaddonDownloadDialog

Description: Shows a dialog for downloading the VirtualBox additions to the m23 server.

5.86.35 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.

5.86.36 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.

5.86.37 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.

5.86.38 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.

5.86.39 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.

5.86.40 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.
- visual: Set to true, if the VM should be started in visual mode always.

Returns:

- True if the command can be executed otherwise false.

5.86.41 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.

5.86.42 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.

5.86.43 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.

5.86.44 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.

5.86.45 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.

5.86.46 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.

5.86.47 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.

5.86.48 VM_GUIstepCreateGuest

Description: Shows a dialog to create a new VM on the chosen host.

5.86.49 VM_GUIstepCheckHost

Description: Shows a dialog part with information about the chose VM host.

5.86.50 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.

5.86.51 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.

5.86.52 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.

5.86.53 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.

5.86.54 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.

5.86.55 VM_statusIcons

Description: Returns HTML codes that include the VM status icons of the client.
Parameter:

- clientName: Name of the m23 client
- id: ID of the m23 client
- vmRole: Role of the m23 client (host, guest, no virtualisation)
- vmSoftware: Number of the used virtualisation software.

Returns:

- HTML codes with included status icons.

5.86.56 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.

5.86.57 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.

5.86.58 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 (de)activate network booting of a VM.

5.86.59 VM_restoreSnapshot

Description: Restores a snapshot.
Parameter:

- type: VM_SW_VBOX for VirtualBox OSE
- vmName: Name of the VM.
- snapshotName: Name of the snapshot to restore.

Returns:

- BASH code to restore a snapshot.

5.86.60 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.
- imageDir: Base directory where to create the directory structure for storing the VM files.
- arch: Architecture of the VM (amd64 or i386)

Returns:

- BASH code to create a virtual machine.

5.86.61 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.

5.86.62 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.

5.86.63 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.

5.86.64 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.

5.86.65 VM_stopVMCommandFile

Description: Removes automatical staring of a VM by removing the command file.
Parameter:

- vmName: Name of the VM.
- cmd: Bash code to stop the VM.

5.86.66 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.

5.86.67 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.

5.86.68 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.

5.86.69 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.

5.86.70 VM_parseStatus

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

- Array with the current state of the VM.