



# ***m23 software distribution***

m23 Development Guide  
for m23 rock 18.2

Hauke Goos-Habermann

May 8, 2018



# Contents

<b>1</b>	<b>introduction</b>	<b>5</b>
1.1	What you can expect from this document: . . . . .	5
1.2	What you can't expect from this document: . . . . .	5
<b>2</b>	<b>m23 license: The GNU General Public License</b>	<b>7</b>
<b>3</b>	<b>MDK menu system</b>	<b>13</b>
3.1	Introduction . . . . .	13
3.2	Starting the MDK menu system . . . . .	13
<b>4</b>	<b>database</b>	<b>15</b>
4.1	overview . . . . .	15
4.2	the tables . . . . .	15
4.2.1	clientjobs . . . . .	15
4.2.2	clientlogs . . . . .	16
4.2.3	clientpackages . . . . .	16
4.2.4	clientpreferences . . . . .	16
4.2.5	clients . . . . .	16
4.2.6	ftpusers . . . . .	17
4.2.7	groups . . . . .	18
4.2.8	plugins . . . . .	18
4.2.9	recommendpackages . . . . .	18
4.2.10	remotevar . . . . .	18
<b>5</b>	<b>HowTos</b>	<b>19</b>
5.1	Using new Debian/Ubuntu releases with m23 . . . . .	19
5.1.1	Test and development . . . . .	19
5.1.2	Building the packages . . . . .	20
5.2	Releasing a new m23 version . . . . .	20
5.2.1	Switch to relase . . . . .	20
5.2.2	Documentation . . . . .	20
5.2.3	Building the files . . . . .	20
5.2.4	Update the m23admin-offline-copy . . . . .	21
5.2.5	CMS . . . . .	21
5.2.6	Announcement . . . . .	22
5.3	How to translate m23? . . . . .	22
5.3.1	make directories . . . . .	22
5.3.2	generate the language file . . . . .	22
5.3.3	translating the messages . . . . .	22
5.3.4	translating the help texts . . . . .	23
5.4	How to generate a new base system? . . . . .	23
5.4.1	install necessary system files . . . . .	23
5.4.2	editing files . . . . .	23
5.4.3	step in to your new system . . . . .	24

5.4.4	source update and cleanup . . . . .	24
5.4.5	installing a kernel . . . . .	24
5.4.6	system update and installation of additional packages . . . . .	24
5.4.7	booting your new system . . . . .	24
5.5	How to create a bootable server installation CD? . . . . .	25
5.5.1	Introduction . . . . .	25
5.5.2	what does the makeCD script? . . . . .	25
5.6	How to make network bootimages? . . . . .	25
5.6.1	Introduction . . . . .	25
5.6.2	creating a new bootimage . . . . .	26
5.6.3	What can you do with this SDK? . . . . .	27
5.7	How to generate an Etherboot bootimage for booting from harddisk? . . . . .	27
5.7.1	exchange epic100.c with the special boot code . . . . .	27
5.7.2	Making the bootimage . . . . .	31
5.8	How to create the server update files? . . . . .	31
5.8.1	The files . . . . .	31
5.9	debugging m23 scripts . . . . .	32
5.9.1	Activate PHP debugging . . . . .	32
5.9.2	Turn on m23 script debugging . . . . .	32
5.9.3	Turn off m23 script debugging . . . . .	33
5.9.4	Set script status . . . . .	33
5.9.5	get the output of your script . . . . .	33
5.10	Additional installation parameters for normal packages . . . . .	33
5.10.1	The *OptionPage.php . . . . .	33
5.10.2	Filling the *OptionPage.php . . . . .	34
5.10.3	Layoutoptions . . . . .	34
<b>6</b>	<b>m23 helpers</b>	<b>37</b>
6.1	mdoc . . . . .	37
6.1.1	what is mdoc? . . . . .	37
6.1.2	how to make your source code mdoc compatible? . . . . .	37
6.1.3	mdoc info block . . . . .	37
6.1.4	example for a mdoc comment . . . . .	38
6.1.5	using mdoc . . . . .	38
6.1.6	example . . . . .	38
<b>7</b>	<b>m23customPatch</b>	<b>39</b>
7.1	Indicating patchable areas in the source code . . . . .	39
7.1.1	Start position of a patchable area . . . . .	39
7.1.2	End position of a patchable area . . . . .	39
7.1.3	Example (/m23/data+scripts/m23admin/head/head.php) . . . . .	40
7.2	m23customPatch file format . . . . .	40
7.2.1	Example (logo.php.m23customPatch) . . . . .	40
7.3	/m23/bin/m23customPatch . . . . .	40
7.3.1	Return/error codes . . . . .	40
7.4	Applying patches on m23 update . . . . .	40
<b>8</b>	<b>m23 API reference</b>	<b>41</b>
8.1	./data+scripts/packages/m23CommonInstallRoutines.php . . . . .	42
8.1.1	CIR_setDateAndTimeTemporarily . . . . .	42
8.1.2	CIR_WorkaroundForMissingModulesDep . . . . .	42
8.1.3	CIR_transferClientIP . . . . .	42
8.1.4	CIR_writeClientID . . . . .	42
8.1.5	CIR_detectSCSI . . . . .	42
8.1.6	CIR_waitForNextJob . . . . .	42
8.1.7	CIR_enableDropbear . . . . .	42

8.2	./inc/assimilate.php . . . . .	43
8.2.1	ASSI_showClientAddDialog . . . . .	43
8.2.2	ASSI_addClient . . . . .	43
8.2.3	ASSI_addUbuntuRoot . . . . .	43
8.2.4	ASSI_prepareClient . . . . .	43
8.3	./inc/autoTest.php . . . . .	44
8.3.1	AUTOTEST_getKey . . . . .	44
8.3.2	AUTOTEST_calcScancodes . . . . .	44
8.3.3	AUTOTEST_keyAndRelease . . . . .	44
8.3.4	AUTOTEST_VM_create . . . . .	44
8.3.5	AUTOTEST_VM_enableCapture . . . . .	45
8.3.6	AUTOTEST_VM_delete . . . . .	45
8.3.7	AUTOTEST_VM_start . . . . .	45
8.3.8	AUTOTEST_VM_insertBootISO . . . . .	45
8.3.9	AUTOTEST_VM_rebootFromHD . . . . .	45
8.3.10	AUTOTEST_executePHPFunction . . . . .	45
8.3.11	AUTOTEST_VM_keyboardWrite . . . . .	45
8.3.12	AUTOTEST_VM_ocrScreen . . . . .	46
8.3.13	AUTOTEST_VM_getStatus . . . . .	46
8.3.14	AUTOTEST_VM_isRunning . . . . .	46
8.4	./inc/backup.php . . . . .	47
8.4.1	BACKUP_showClientSettings . . . . .	47
8.4.2	BACKUP_getBackupDirs . . . . .	47
8.4.3	BACKUP_saveBackupDirs . . . . .	47
8.4.4	BACKUP_getAdmins . . . . .	47
8.4.5	BACKUP_addAdmin . . . . .	47
8.4.6	BACKUP_delAdmin . . . . .	47
8.5	./inc/bittorrent.php . . . . .	48
8.5.1	BT_dlFile . . . . .	48
8.5.2	BT_status . . . . .	48
8.5.3	BT_checkSoftware . . . . .	48
8.5.4	BT_stopService . . . . .	48
8.5.5	BT_startService . . . . .	48
8.5.6	BT_stopTracker . . . . .	49
8.5.7	BT_startTracker . . . . .	49
8.5.8	BT_restartTracker . . . . .	49
8.5.9	BT_autostart . . . . .	49
8.5.10	BT_updateWhitelist . . . . .	49
8.5.11	BT_createTorrent . . . . .	49
8.5.12	BT_startClient . . . . .	49
8.5.13	BT_stopClient . . . . .	49
8.5.14	BT_restartClient . . . . .	49
8.6	./inc/burn.php . . . . .	50
8.6.1	BURN_listBurners . . . . .	50
8.6.2	BURN_getDevice . . . . .	50
8.6.3	BURN_blank . . . . .	50
8.6.4	BURN_burn . . . . .	50
8.6.5	BURN_getStatus . . . . .	50
8.6.6	BURN_showLog . . . . .	50
8.6.7	BURN_checkISO . . . . .	50
8.6.8	BURN_getISOSize . . . . .	51
8.7	./inc/capture.php . . . . .	52
8.7.1	CAPTURE_getKeys . . . . .	52
8.7.2	CAPTURE_captureAll . . . . .	52
8.7.3	CAPTURE_load . . . . .	52
8.7.4	CAPTURE_deActivate . . . . .	52

8.7.5	CAPTURE_isActive . . . . .	52
8.7.6	CAPTURE_captureImg . . . . .	52
8.7.7	CAPTURE_toggle . . . . .	52
8.7.8	CAPTURE_showMessageBox . . . . .	53
8.7.9	CAPTURE_showEntries . . . . .	53
8.7.10	CAPTURE_deleteById . . . . .	53
8.7.11	CAPTURE_showMarker . . . . .	53
8.7.12	CAPTURE_showTableWith . . . . .	53
8.8	/inc/CAutoTest.php . . . . .	54
8.8.1	CClient::__construct . . . . .	54
8.8.2	CAutoTest::readAndDefineOrDieXML . . . . .	54
8.8.3	CAutoTest::readSettings . . . . .	54
8.8.4	CAutoTest::VMCreate . . . . .	54
8.8.5	CAutoTest::VMStart . . . . .	54
8.8.6	CAutoTest::setISO . . . . .	54
8.8.7	CAutoTest::getISO . . . . .	54
8.8.8	CAutoTest::isVM . . . . .	55
8.8.9	CAutoTest::getVM_hdsize . . . . .	55
8.8.10	CAutoTest::getVMRam . . . . .	55
8.8.11	CAutoTest::getMachine . . . . .	55
8.8.12	CAutoTest::getMovieFileName . . . . .	55
8.8.13	CAutoTest::getLogFileName . . . . .	55
8.8.14	CAutoTest::addToLogFile . . . . .	55
8.8.15	CAutoTest::getTimestampString . . . . .	55
8.8.16	CAutoTest::setTriggered . . . . .	56
8.8.17	CAutoTest::isTriggered . . . . .	56
8.8.18	CAutoTest::setTimeout . . . . .	56
8.8.19	CAutoTest::decTimeout . . . . .	56
8.8.20	CAutoTest::addToSequence . . . . .	56
8.8.21	CAutoTest::matchArray . . . . .	56
8.8.22	CAutoTest::getAnswersA . . . . .	57
8.8.23	CAutoTest::getTriggerTypes . . . . .	57
8.8.24	CAutoTest::getTriggerParams . . . . .	57
8.8.25	CAutoTest::getExecTypes . . . . .	57
8.8.26	CAutoTest::getExecParams . . . . .	57
8.8.27	CAutoTest::checkTriggerResult . . . . .	57
8.8.28	CAutoTest::executeTriggerAction . . . . .	57
8.8.29	CAutoTest::waitForTrigger . . . . .	57
8.8.30	CAutoTest::getCurElement . . . . .	57
8.8.31	CAutoTest::nextCurElement . . . . .	58
8.8.32	CAutoTest::showAndLogMessage . . . . .	58
8.8.33	CAutoTest::elemOk . . . . .	58
8.8.34	CAutoTest::elemWarn . . . . .	58
8.8.35	CAutoTest::elemBad . . . . .	58
8.8.36	CAutoTest::setVariableFromXML . . . . .	58
8.8.37	CAutoTest::triggerTypeToConstant . . . . .	58
8.8.38	CAutoTest::parseTriggerFromXML . . . . .	59
8.8.39	CAutoTest::parseActionFromXML . . . . .	59
8.8.40	CAutoTest::parseAnswersFromXML . . . . .	59
8.8.41	CAutoTest::parseXML . . . . .	59
8.9	/inc/CAutoUpdate.php . . . . .	60
8.9.1	CAutoUpdate::getActiveSetID . . . . .	60
8.9.2	CAutoUpdate::getActiveSet . . . . .	60
8.9.3	CAutoUpdate::addSet . . . . .	60
8.9.4	CAutoUpdate::setParallelUpdates . . . . .	60
8.9.5	CAutoUpdate::getParallelUpdates . . . . .	60

8.9.6	CAutoUpdate::setType . . . . .	61
8.9.7	CAutoUpdate::getType . . . . .	61
8.9.8	CAutoUpdate::getUpdateTypeTranslation . . . . .	61
8.9.9	CAutoUpdate::delSet . . . . .	61
8.9.10	CAutoUpdate::getAllSets . . . . .	61
8.9.11	CAutoUpdate::loadSets . . . . .	62
8.9.12	CAutoUpdate::saveSets . . . . .	62
8.9.13	CAutoUpdate::getSetAmount . . . . .	62
8.9.14	CAutoUpdate::configureCron . . . . .	62
8.9.15	CAutoUpdate::isASetActive . . . . .	62
8.9.16	CAutoUpdate::activateSet . . . . .	62
8.9.17	CAutoUpdate::activateNextSet . . . . .	62
8.9.18	CAutoUpdate::activateNextTimlySet . . . . .	62
8.9.19	CAutoUpdate::addStartEnd . . . . .	63
8.9.20	CAutoUpdate::delStartEnd . . . . .	63
8.9.21	CAutoUpdate::getAllStartsEndsHumanReadable . . . . .	63
8.9.22	CAutoUpdate::addGroup . . . . .	63
8.9.23	CAutoUpdate::delGroup . . . . .	63
8.9.24	CAutoUpdate::getAllGroupsHumanReadable . . . . .	64
8.9.25	CAutoUpdate::getImgButtonParams . . . . .	64
8.9.26	CAutoUpdate::isTimeToRunNow . . . . .	64
8.9.27	CAutoUpdate::getActiveGroups . . . . .	64
8.9.28	CAutoUpdate::getAllClientnames . . . . .	64
8.9.29	CAutoUpdate::startNewClients . . . . .	65
8.9.30	CAutoUpdate::run . . . . .	65
8.9.31	CAutoUpdate::showAutoUpdateManager . . . . .	65
8.10	/inc/CChecks.php . . . . .	66
8.10.1	CChecks::checkSizeInMB . . . . .	66
8.10.2	CChecks::checkDiskDefinedSize . . . . .	66
8.10.3	CChecks::checkFdiskAdjustmentUpperToleranceIdentical . . . . .	66
8.10.4	CChecks::checkFdiskAdjustmentLowerToleranceIdentical . . . . .	66
8.10.5	CChecks::checkSwapPart . . . . .	66
8.10.6	CChecks::checkEFIPart . . . . .	67
8.10.7	CChecks::checkInstPart . . . . .	67
8.10.8	CChecks::checkMountDev . . . . .	67
8.10.9	CChecks::checkMountPoint . . . . .	67
8.10.10	CChecks::checkFdiskAdjustmentSpecifiedDev . . . . .	67
8.10.11	CChecks::checkRelease . . . . .	67
8.10.12	CChecks::checkDistribution . . . . .	68
8.10.13	CChecks::checkNfshomeserver . . . . .	68
8.10.14	CChecks::checkGroupname . . . . .	68
8.10.15	CChecks::checkUserGroupIDs . . . . .	68
8.10.16	CChecks::checkLanguage . . . . .	68
8.10.17	CChecks::checkRootpassword . . . . .	68
8.10.18	CChecks::firstpw . . . . .	69
8.10.19	CChecks::checkLogin . . . . .	69
8.10.20	CChecks::checkProxy . . . . .	69
8.10.21	CChecks::checkOffice . . . . .	69
8.10.22	CChecks::checkEmail . . . . .	69
8.10.23	CChecks::checkFamilyname . . . . .	70
8.10.24	CChecks::checkForename . . . . .	70
8.10.25	CChecks::checkIPGeneric . . . . .	70
8.10.26	CChecks::checkIP . . . . .	70
8.10.27	CChecks::checkNonusedIP . . . . .	70
8.10.28	CChecks::checkDNS1 . . . . .	71
8.10.29	CChecks::checkDNS2 . . . . .	71

8.10.30 CChecks::checkGateway . . . . .	71
8.10.31 CChecks::checkClientname . . . . .	71
8.10.32 CChecks::checkNonusedClientname . . . . .	71
8.10.33 CChecks::checkNetmask . . . . .	72
8.10.34 CChecks::checkMAC . . . . .	72
8.10.35 CChecks::checkNonusedMAC . . . . .	72
8.10.36 CChecks::checkIPRange . . . . .	72
8.10.37 CChecks::checkStatus . . . . .	72
8.10.38 CChecks::checkBootType . . . . .	73
8.10.39 CChecks::genericCHECK_FW . . . . .	73
8.10.40 CChecks::checkPoolName . . . . .	73
8.11 /inc/CClientLister.php . . . . .	74
8.11.1 CClientLister::showClientTable . . . . .	74
8.11.2 CClientLister::generateHTMLClientNameIdCheckbox . . . . .	74
8.11.3 CClientLister::importCheckedClientNamesIds . . . . .	74
8.11.4 CClientLister::getCheckedClientNames . . . . .	74
8.11.5 CClientLister::isVisibleByPingableFilter . . . . .	74
8.11.6 CClientLister::setVisibleByPingableFilter . . . . .	74
8.11.7 CClientLister::getClientNames . . . . .	74
8.11.8 CClientLister::showClientList . . . . .	75
8.11.9 CClientLister::showClientListExtraLine . . . . .	75
8.11.10 CClientLister::setClientListExtraLine . . . . .	75
8.11.11 CClientLister::getColumnNrToSort . . . . .	75
8.11.12 CClientLister::getColumnNrWithContinousNumber . . . . .	75
8.11.13 CClientLister::cmpArrayElements . . . . .	75
8.11.14 CClientLister::sortLines . . . . .	75
8.11.15 CClientLister::getAscDescHeader . . . . .	75
8.11.16 CClientLister::getStatusHeader . . . . .	75
8.11.17 CClientLister::getClientNameHeader . . . . .	76
8.11.18 CClientLister::getInstallDateHeader . . . . .	76
8.11.19 CClientLister::getLastModifyHeader . . . . .	76
8.11.20 CClientLister::getIPHeader . . . . .	76
8.11.21 CClientLister::getMACHeader . . . . .	76
8.11.22 CClientLister::getJobsHeader . . . . .	76
8.11.23 CClientLister::getPackagesHeader . . . . .	76
8.11.24 CClientLister::getGroupHeader . . . . .	76
8.11.25 CClientLister::showClientListHeader . . . . .	76
8.11.26 CClientLister::setOutputColumns . . . . .	76
8.11.27 CClientLister::setActionString . . . . .	76
8.11.28 CClientLister::getActionString . . . . .	77
8.11.29 CClientLister::setOrderBy . . . . .	77
8.11.30 CClientLister::addStatusFilter . . . . .	77
8.11.31 CClientLister::addKeyValueStoreFilter . . . . .	77
8.11.32 CClientLister::addGroupFilter . . . . .	77
8.11.33 CClientLister::addSearchFilter . . . . .	77
8.11.34 CClientLister::setAscending . . . . .	77
8.11.35 CClientLister::isAscending . . . . .	78
8.11.36 CClientLister::vmRunOnHostFilter . . . . .	78
8.11.37 CClientLister::resetGetting . . . . .	78
8.11.38 CClientLister::getClient . . . . .	78
8.11.39 CClientLister::getAllMatchingClients . . . . .	78
8.11.40 CClientLister::isMatchingClientPresent . . . . .	78
8.11.41 CClientLister::getAllAutoUpdateClientNames . . . . .	78
8.11.42 CClientLister::IPexists . . . . .	78
8.11.43 CClientLister::MACexists . . . . .	79
8.11.44 CClientLister::ClientExists . . . . .	79



8.12	/inc/CClient.php	80
8.12.1	CClient::__construct	80
8.12.2	CClient::__destruct	80
8.12.3	CClient::setKeyValueStore	80
8.12.4	CClient::getKeyValueStore	80
8.12.5	CClient::copyImagingParameters	80
8.12.6	CClient::copyMassOptions	80
8.12.7	CClient::setInstallationStatusBar	81
8.12.8	CClient::delSpecialJob	81
8.12.9	CClient::addJob	81
8.12.10	CClient::includeDistributionSpecificPackagesPHP	81
8.12.11	CClient::addNormalJob	81
8.12.12	CClient::addSpecialJob	81
8.12.13	CClient::addUpdateSourcesListJob	81
8.12.14	CClient::addUpdatePackageInfosJob	82
8.12.15	CClient::addNormalUpdateJob	82
8.12.16	CClient::addCompleteUpdateJob	82
8.12.17	CClient::addUpdateJob	82
8.12.18	CClient::startAutoUpdate	82
8.12.19	CClient::stopAutoUpdate	82
8.12.20	CClient::getAutoUpdate_lastAttempt	82
8.12.21	CClient::hasAutoUpdateJob	82
8.12.22	CClient::checkAutoUpdate	82
8.12.23	CClient::unsetInstPartDev	83
8.12.24	CClient::setInstPartDev	83
8.12.25	CClient::setUEFI	83
8.12.26	CClient::isUEFIActive	83
8.12.27	CClient::isHalfSisterClient	83
8.12.28	CClient::getEFIBootPartDev	83
8.12.29	CClient::setEFIBootPartDev	83
8.12.30	CClient::unsetEFIBootPartDev	83
8.12.31	CClient::getInstPartDev	84
8.12.32	CClient::unsetSwapPartDev	84
8.12.33	CClient::setSwapPartDev	84
8.12.34	CClient::getSwapPartDev	84
8.12.35	CClient::isDerivedClient	84
8.12.36	CClient::setLanguage	84
8.12.37	CClient::getLanguage	84
8.12.38	CClient::setTimeZone	84
8.12.39	CClient::getTimeZone	85
8.12.40	CClient::setBootloader	85
8.12.41	CClient::getBootloader	85
8.12.42	CClient::getClientGroup	85
8.12.43	CClient::setNetRootPwd	85
8.12.44	CClient::getNetRootPwd	85
8.12.45	CClient::setNfshomeserver	85
8.12.46	CClient::getNfshomeserver	85
8.12.47	CClient::setBoolProperty	86
8.12.48	CClient::setInstallPrinter	86
8.12.49	CClient::getInstallPrinter	86
8.12.50	CClient::setAddNewLocalLogin	86
8.12.51	CClient::getAddNewLocalLogin	86
8.12.52	CClient::setGetSystemtimeByNTP	86
8.12.53	CClient::getGetSystemtimeByNTP	87
8.12.54	CClient::setRootPassword	87
8.12.55	CClient::addToClientGroup	87

8.12.56 CClient::setUserGroupIDs . . . . .	87
8.12.57 CClient::getGroupID . . . . .	87
8.12.58 CClient::getUserID . . . . .	87
8.12.59 CClient::setUserDetails . . . . .	88
8.12.60 CClient::getFirstpw . . . . .	88
8.12.61 CClient::getFamilyname . . . . .	88
8.12.62 CClient::getForename . . . . .	88
8.12.63 CClient::getLogin . . . . .	88
8.12.64 CClient::setLDAPType . . . . .	88
8.12.65 CClient::getLDAPType . . . . .	88
8.12.66 CClient::setLDAPServer . . . . .	89
8.12.67 CClient::getLDAPServer . . . . .	89
8.12.68 CClient::addToCredentialsToLDAPServer . . . . .	89
8.12.69 CClient::getBackToDetailsLink . . . . .	89
8.12.70 CClient::getClientWorkPHPURL . . . . .	89
8.12.71 CClient::getClientCurrentWorkPHP . . . . .	89
8.12.72 CClient::getClientPackages . . . . .	89
8.12.73 CClient::save . . . . .	90
8.12.74 CClient::getProperty . . . . .	90
8.12.75 CClient::getSourcesList . . . . .	90
8.12.76 CClient::setArch . . . . .	90
8.12.77 CClient::getArch . . . . .	90
8.12.78 CClient::setDistribution . . . . .	90
8.12.79 CClient::getDistribution . . . . .	90
8.12.80 CClient::setRelease . . . . .	91
8.12.81 CClient::getRelease . . . . .	91
8.12.82 CClient::updateModifyDate . . . . .	91
8.12.83 CClient::getModifyDate . . . . .	91
8.12.84 CClient::getModifyDateHumanReadable . . . . .	91
8.12.85 CClient::updateInstallDate . . . . .	91
8.12.86 CClient::getInstallDate . . . . .	91
8.12.87 CClient::getInstallDateHumanReadable . . . . .	91
8.12.88 CClient::export . . . . .	91
8.12.89 CClient::setClientName . . . . .	92
8.12.90 CClient::getClientName . . . . .	92
8.12.91 CClient::getID . . . . .	92
8.12.92 CClient::setDNS . . . . .	92
8.12.93 CClient::getDNS1 . . . . .	92
8.12.94 CClient::getDNS2 . . . . .	92
8.12.95 CClient::setPackageProxy . . . . .	92
8.12.96 CClient::setIP . . . . .	93
8.12.97 CClient::getIP . . . . .	93
8.12.98 CClient::setNetmask . . . . .	93
8.12.99 CClient::getNetmask . . . . .	93
8.12.100 CClient::setMAC . . . . .	93
8.12.101 CClient::getMAC . . . . .	93
8.12.102 CClient::setGateway . . . . .	94
8.12.103 CClient::getGateway . . . . .	94
8.12.104 CClient::setGateway . . . . .	94
8.12.105 CClient::getStatus . . . . .	94
8.12.106 CClient::isDefinedClient . . . . .	94
8.12.107 CClient::generateHTMLStatusBar . . . . .	94
8.12.108 CClient::getBootType . . . . .	94
8.12.109 CClient::usesDynamicIP . . . . .	94
8.12.110 CClient::setBootType . . . . .	95
8.12.111 CClient::getNetworkBootTypesArrayForSelection . . . . .	95

8.12.11	CCClient::startInstall . . . . .	95
8.12.11	CCClient::addShutdownPackage . . . . .	95
8.12.11	CCClient::activateNetboot . . . . .	95
8.12.11	CCClient::deactivateNetboot . . . . .	95
8.12.11	CCClient::wol . . . . .	95
8.12.11	CCClient::isNetbootActive . . . . .	95
8.12.11	CCClient::isPingable . . . . .	96
8.12.11	CCClient::sshFetchJob . . . . .	96
8.12.12	CCClient::executeBySSH . . . . .	96
8.12.12	CCClient::generateHTMLClientNameBar . . . . .	96
8.12.12	CCClient::generateHTMLPackagesBar . . . . .	96
8.12.12	CCClient::generateHTMLWaitingAllJobsBar . . . . .	96
8.12.12	CCClient::generateHTMLWaitingAllJobsBar . . . . .	96
8.12.12	CCClient::getActionString . . . . .	97
8.12.12	CCClient::destroy . . . . .	97
8.12.12	CCClient::getVMRole . . . . .	97
8.12.12	CCClient::getVmSoftware . . . . .	97
8.13	./inc/CFDiskBasic.php . . . . .	98
8.13.1	CFDiskBasic::__construct . . . . .	98
8.13.2	CFDiskBasic::__destruct . . . . .	98
8.13.3	CFDiskBasic::deletePartitionJob . . . . .	98
8.13.4	CFDiskBasic::createPartitionJob . . . . .	98
8.13.5	CFDiskBasic::bootflagJob . . . . .	98
8.13.6	CFDiskBasic::EFItypeAndGUIDJob . . . . .	99
8.13.7	CFDiskBasic::formatJob . . . . .	99
8.13.8	CFDiskBasic::createRaidJob . . . . .	99
8.13.9	CFDiskBasic::deleteRaidJob . . . . .	99
8.13.10	CFDiskBasic::makeInstOrEFIPartBootable . . . . .	99
8.13.11	CFDiskBasic::rereadPartTable . . . . .	99
8.13.12	CFDiskBasic::getMknodCommandsForDeviceArray . . . . .	99
8.13.13	CFDiskBasic::genPartedCommands . . . . .	100
8.13.14	CFDiskBasic::createPartition . . . . .	100
8.13.15	CFDiskBasic::createUEFIPartition . . . . .	100
8.13.16	CFDiskBasic::deletePartition . . . . .	100
8.13.17	CFDiskBasic::formatPartition . . . . .	100
8.13.18	CFDiskBasic::deleteAllPartitions . . . . .	101
8.13.19	CFDiskBasic::createInstallPartition . . . . .	101
8.13.20	CFDiskBasic::createSwapPartition . . . . .	101
8.13.21	CFDiskBasic::autoPartitionDisk . . . . .	101
8.13.22	CFDiskBasic::PM_auto2Disk1SysSwap2Home . . . . .	101
8.13.23	CFDiskBasic::createAllRaidJobs . . . . .	101
8.13.24	CFDiskBasic::createRaidJobForRaid . . . . .	101
8.13.25	CFDiskBasic::createRaid . . . . .	102
8.13.26	CFDiskBasic::deleteRaid . . . . .	102
8.13.27	CFDiskBasic::assignDeviceToRaid . . . . .	102
8.13.28	CFDiskBasic::deleteDeviceFromRaid . . . . .	102
8.13.29	CFDiskBasic::deleteAllPartitionsOnDisk . . . . .	102
8.13.30	CFDiskBasic::fdiskAdjustPartitioningLinearScale . . . . .	102
8.13.31	CFDiskBasic::fdiskAdjustPartitioning . . . . .	102
8.14	./inc/CFDiskGUI.php . . . . .	103
8.14.1	CFDiskBasic::__construct . . . . .	103
8.14.2	CFDiskGUI::__destruct . . . . .	103
8.14.3	CFDiskGUI::fdiskSessionSetter . . . . .	103
8.14.4	CFDiskGUI::fdiskSessionPage . . . . .	103
8.14.5	FDISK_fdiskSessionReset . . . . .	103
8.14.6	CFDiskGUI::fdiskSessionHelpPage . . . . .	103

8.14.7	CFDiskGUI::getDiskLayoutEntryStart . . . . .	104
8.14.8	CFDiskGUI::getDiskLayoutEntryEnd . . . . .	104
8.14.9	CFDiskGUI::getDiskLayoutEntrySize . . . . .	104
8.14.10	CFDiskGUI::getDiskLayoutEntryType . . . . .	104
8.14.11	CFDiskGUI::getDiskLayoutEntryFileSystem . . . . .	104
8.14.12	CFDiskGUI::getFileSystemTranslator . . . . .	105
8.14.13	CFDiskGUI::getPartitionTypeTranslator . . . . .	105
8.14.14	CFDiskGUI::getDiskLayoutEntryDev . . . . .	105
8.14.15	CFDiskGUI::getDiskLayoutEntriesAmount . . . . .	105
8.14.16	CFDiskGUI::getFreeSpaceHTMLColor . . . . .	105
8.14.17	CFDiskGUI::showFreeSpaceBarBlock . . . . .	105
8.14.18	CFDiskGUI::showEmptyDiskTable . . . . .	106
8.14.19	CFDiskGUI::getPartitionExtraIconHTML . . . . .	106
8.14.20	CFDiskGUI::getExtraCSSForExtended . . . . .	106
8.14.21	CFDiskGUI::showPartitionBarBlock . . . . .	106
8.14.22	CFDiskGUI::printBars . . . . .	107
8.14.23	CFDiskGUI::showPartTable . . . . .	107
8.14.24	CFDiskGUI::addDiskLayoutEntry . . . . .	107
8.14.25	CFDiskGUI::newDiskLayout . . . . .	107
8.14.26	CFDiskGUI::updateDiskLayout . . . . .	107
8.14.27	CFDiskGUI::showCombinedFdiskGUIDialog . . . . .	107
8.14.28	CFDiskGUI::getCurrentDiskDev . . . . .	108
8.14.29	CFDiskGUI::getCurrentDiskvDev . . . . .	108
8.14.30	CFDiskGUI::getCurrentPartDev . . . . .	108
8.14.31	CFDiskGUI::finalChecksAndRealPartitionAndFormatStart . . . . .	108
8.14.32	CFDiskGUI::showFdiskCombinedGUIFunctions . . . . .	108
8.14.33	CFDiskGUI::getPartitionPercent . . . . .	108
8.14.34	CFDiskGUI::getHTMLColorForFilesystemOrType . . . . .	108
8.14.35	CFDiskGUI::getPartTableEntryNr . . . . .	109
8.14.36	CFDiskGUI::getJSSelectPartTypeForNewPartitions . . . . .	109
8.14.37	CFDiskGUI::printAllBars . . . . .	109
8.14.38	CFDiskGUI::getPartInfoIcon . . . . .	109
8.14.39	CFDiskGUI::getPartInfoString . . . . .	109
8.14.40	CFDiskGUI::getDiskInfoString . . . . .	110
8.14.41	CFDiskGUI::getDriveInfoIcon . . . . .	110
8.14.42	CFDiskGUI::listPartJobs . . . . .	110
8.14.43	CFDiskGUI::showAllPartTables . . . . .	110
8.14.44	CFDiskGUI::getRaidTable . . . . .	110
8.14.45	CFDiskGUI::showColorDefinitions . . . . .	110
8.14.46	CFDiskGUI::fstabAddDialog . . . . .	110
8.14.47	CFDiskGUI::getFstabTable . . . . .	110
8.14.48	CFDiskGUI::getHDSizes . . . . .	111
8.15	./inc/CFDiskIO.php . . . . .	112
8.15.1	CFDiskIO::__construct . . . . .	112
8.15.2	CFDiskIO::__destruct . . . . .	112
8.15.3	CFDiskIO::isTypeFreeSpace . . . . .	112
8.15.4	CFDiskIO::fdiskGetProperty . . . . .	112
8.15.5	CFDiskIO::getCurrentPartitioning . . . . .	112
8.15.6	CFDiskIO::setCurrentPartitioning . . . . .	112
8.15.7	CFDiskIO::fdiskLoadFromDB . . . . .	113
8.15.8	CFDiskIO::setCFDiskTemp . . . . .	113
8.15.9	CFDiskIO::getDiskDevs . . . . .	113
8.15.10	CFDiskIO::getDiskAmount . . . . .	113
8.15.11	CFDiskIO::getPartAmount . . . . .	113
8.15.12	CFDiskIO::getpDiskAndpPartFromDev . . . . .	113
8.15.13	CFDiskIO::isDevValidDiskPartitionOrRaid . . . . .	113

8.15.14 CFDiskIO::dev2VDiskVPart . . . . .	114
8.15.15 CFDiskIO::getPreviousPartitionStep . . . . .	114
8.15.16 CFDiskIO::backToPreviousPartitionStep . . . . .	114
8.15.17 CFDiskIO::findAndSetEFIBootPartDev . . . . .	114
8.15.18 CFDiskIO::discardUndo . . . . .	114
8.15.19 CFDiskIO::updateUndoMd5 . . . . .	114
8.15.20 CFDiskIO::getUndoMd5 . . . . .	114
8.15.21 CFDiskIO::addUndo . . . . .	114
8.15.22 CFDiskIO::addPartitionStepAtTheBeginning . . . . .	115
8.15.23 CFDiskIO::addPartitionStepBeforeFormat . . . . .	115
8.15.24 CFDiskIO::addPartitionStepBeforeFormatArray . . . . .	115
8.15.25 CFDiskIO::addPartitionStep . . . . .	115
8.15.26 CFDiskIO::shiftPartitionStep . . . . .	115
8.15.27 CFDiskIO::resetWantedPartitioningAndSteps . . . . .	115
8.15.28 CFDiskIO::resetWantedPartitioning . . . . .	115
8.15.29 CFDiskIO::fdiskSaveToDB . . . . .	115
8.15.30 CFDiskIO::getCurrentDiskSizesForDefinedDiskSizes . . . . .	116
8.15.31 CFDiskIO::saveDefinedDiskSizesToDB . . . . .	116
8.15.32 CFDiskIO::getDefinedDiskSizes . . . . .	116
8.15.33 CFDiskIO::convertPartitioning2Array . . . . .	116
8.15.34 CFDiskIO::isDiskRaid . . . . .	116
8.15.35 CFDiskIO::isDevRaid . . . . .	116
8.15.36 CFDiskIO::isNVMe . . . . .	116
8.15.37 CFDiskIO::setDiskPartLockedByRaid . . . . .	117
8.15.38 CFDiskIO::isDiskOrPartLockedByRaid . . . . .	117
8.15.39 CFDiskIO::isPartitionLockedByRaid . . . . .	117
8.15.40 CFDiskIO::isDiskLockedByRaid . . . . .	117
8.15.41 CFDiskIO::isDiskNotLockedByRaidAgainstCreationOfNewPartition . . . . .	117
8.15.42 CFDiskIO::getPartitionAmountOfType . . . . .	118
8.15.43 CFDiskIO::getExtendedVPart . . . . .	118
8.15.44 CFDiskIO::getDiskSize . . . . .	118
8.15.45 CFDiskIO::getDiskDev . . . . .	118
8.15.46 CFDiskIO::getPartitionStart . . . . .	118
8.15.47 CFDiskIO::getPartitionNumber . . . . .	119
8.15.48 CFDiskIO::getPartitionSize . . . . .	119
8.15.49 CFDiskIO::getPartitionDev . . . . .	119
8.15.50 CFDiskIO::getDevBypDiskpPart . . . . .	119
8.15.51 CFDiskIO::getPartitionDevs . . . . .	120
8.15.52 CFDiskIO::getPartitionEnd . . . . .	120
8.15.53 CFDiskIO::getPartitionType . . . . .	120
8.15.54 CFDiskIO::getPartitionFileSystem . . . . .	120
8.15.55 CFDiskIO::setPartitionFileSystem . . . . .	121
8.15.56 CFDiskIO::virtualDelPartition . . . . .	121
8.15.57 CFDiskIO::virtualDeleteDisk . . . . .	121
8.15.58 CFDiskIO::getvPartBypPart . . . . .	121
8.15.59 CFDiskIO::getLogicalpParts . . . . .	121
8.15.60 CFDiskIO::correctLogical . . . . .	121
8.15.61 CFDiskIO::devNrExists . . . . .	122
8.15.62 CFDiskIO::getBiggestLowestValueOf . . . . .	122
8.15.63 CFDiskIO::getBiggestValueOf . . . . .	122
8.15.64 CFDiskIO::getLowestValueOf . . . . .	122
8.15.65 CFDiskIO::virtualAddPartition . . . . .	123
8.15.66 CFDiskIO::virtualAddDisk . . . . .	123
8.15.67 CFDiskIO::getBelongingRaidDev . . . . .	123
8.15.68 CFDiskIO::FDISK_dev2VDiskVPart_Legacy . . . . .	123
8.15.69 CFDiskIO::collidesWithPartitionOfType . . . . .	124

8.15.70	CFDiskIO::isFreeSpaceBetween . . . . .	124
8.15.71	CFDiskIO::getPossiblePartitionTypesBetween . . . . .	124
8.15.72	CFDiskIO::isNewPartitionOfTypeCreatable . . . . .	124
8.15.73	CFDiskIO::getCreatablePartitionTypes . . . . .	125
8.15.74	CFDiskIO::mayLogicalPartitionsBeChanged . . . . .	125
8.15.75	CFDiskIO::mayPartitioningBeChanged . . . . .	125
8.15.76	CFDiskIO::getNextFreePhysicalVirtualPartitionNumber . . . . .	125
8.15.77	CFDiskIO::nextLogicalDevNr . . . . .	125
8.15.78	CFDiskIO::nextPrimaryDevNr . . . . .	126
8.15.79	CFDiskIO::nextPrimaryDevNr . . . . .	126
8.15.80	CFDiskIO::getWantedPartitioning . . . . .	126
8.15.81	CFDiskIO::sortDiskKeyByDev . . . . .	126
8.15.82	CFDiskIO::sortPartitionKeyByStart . . . . .	126
8.15.83	CFDiskIO::getRaidLevelNumbers . . . . .	126
8.15.84	CFDiskIO::isRaidLevelNumberValid . . . . .	127
8.15.85	CFDiskIO::getRaidDevsBuildingDisk . . . . .	127
8.15.86	CFDiskIO::getvrDevNrByrDev . . . . .	127
8.15.87	CFDiskIO::doesDiskHavePartitionsLockedByRaid . . . . .	127
8.15.88	CFDiskIO::addDevToRaid . . . . .	127
8.15.89	CFDiskIO::delDevFromRaid . . . . .	128
8.15.90	CFDiskIO::updateRAIDSize . . . . .	128
8.15.91	CFDiskIO::getRaidCompleteParameters . . . . .	128
8.15.92	CFDiskIO::isRaidComplete . . . . .	128
8.15.93	CFDiskIO::maximumDevicesForRaidReached . . . . .	128
8.15.94	CFDiskIO::areAllRaidsComplete . . . . .	129
8.15.95	CFDiskIO::getRaidDevsBuildingDiskAmount . . . . .	129
8.15.96	CFDiskIO::setRaidLevel . . . . .	129
8.15.97	CFDiskIO::getDiskSize . . . . .	129
8.15.98	CFDiskIO::virtualCreateRaidDisk . . . . .	129
8.15.99	CFDiskIO::virtualDeleteRaidDisk . . . . .	129
8.15.100	CFDiskIO::getDrivesAndPartitions . . . . .	130
8.15.101	CFDiskIO::getUnusedMDs . . . . .	130
8.15.102	CFDiskIO::getUnusedDiskDev . . . . .	130
8.15.103	CFDiskIO::getUsedMDs . . . . .	130
8.15.104	CFDiskIO::getPartDevs . . . . .	130
8.15.105	CFDiskIO::getNotFS . . . . .	130
8.15.106	CFDiskIO::getSupportedFS . . . . .	131
8.15.107	CFDiskIO::swapFilesystems . . . . .	131
8.15.108	CFDiskIO::installFilesystems . . . . .	131
8.15.109	CFDiskIO::fdiskAddFstab . . . . .	131
8.15.110	CFDiskIO::fdiskGetEntry . . . . .	131
8.15.111	CFDiskIO::fdiskDelFstabEntry . . . . .	131
8.15.112	CFDiskIO::adjustFstabParam . . . . .	132
8.15.113	CFDiskIO::genManualFstab . . . . .	132
8.15.114	CFDiskIO::FDISK_getOldStyleFstabArrayForHalfSister . . . . .	132
8.15.115	CFDiskIO::FDISK_genManualFstab . . . . .	132
8.15.116	CFDiskIO::fdiskGetFstabArray . . . . .	132
8.15.117	CFDiskIO::getPartitionStepsArray . . . . .	132
8.15.118	CFDiskIO::findFstabMountPointByDev . . . . .	133
8.15.119	CFDiskIO::getPrimaryOrEfiPartitionType . . . . .	133
8.15.120	CFDiskIO::changeToEfiPartitionTypeIfInUefiMode . . . . .	133
8.15.121	CFDiskIO::getNVMeMknodCommand . . . . .	133
8.15.122	CFDiskIO::getMknodCommand . . . . .	133
8.16	/inc/CFDiskTest.php . . . . .	134
8.16.1	CFDiskTest::getTestDiskDev . . . . .	134
8.16.2	CFDiskTest::nextTurn . . . . .	134

8.16.3	CFDiskTest::addLineToReplay . . . . .	134
8.16.4	CFDiskTest::rand . . . . .	134
8.16.5	CFDiskTest::getTestDiskvDisk . . . . .	134
8.16.6	CFDiskTest::getRandomPartitionDev . . . . .	134
8.16.7	CFDiskTest::getRadomStartEnd . . . . .	135
8.16.8	CFDiskTest::checkBothSetKey . . . . .	135
8.16.9	CFDiskTest::checkConsistencyDiskKey . . . . .	135
8.16.10	CFDiskTest::checkConsistencyvDiskvPartKey . . . . .	135
8.16.11	CFDiskTest::checkPartitionConsistency . . . . .	135
8.16.12	CFDiskTest::getDiskArrayFromClient . . . . .	135
8.16.13	CFDiskTest::getDiskDevsFromClient . . . . .	136
8.16.14	CFDiskTest::getPartInfoFromClient . . . . .	136
8.16.15	CFDiskTest::getRaidsFromClient . . . . .	136
8.16.16	CFDiskTest::executePartedCommands . . . . .	136
8.16.17	CFDiskTest::createPartition . . . . .	136
8.16.18	CFDiskTest::randomCreatePartition . . . . .	136
8.16.19	CFDiskTest::deletePartition . . . . .	136
8.16.20	CFDiskTest::randomDeletePartition . . . . .	137
8.16.21	CFDiskTest::formatPartition . . . . .	137
8.16.22	CFDiskTest::randomFormatPartition . . . . .	137
8.16.23	CFDiskTest::getRandomFilesystem . . . . .	137
8.16.24	CFDiskTest::getRandomPartitionType . . . . .	137
8.16.25	CFDiskTest::createPartitionsForRaid . . . . .	137
8.16.26	CFDiskTest::showDebugConsistencyClientVirtualArray . . . . .	137
8.16.27	CFDiskTest::randomPartTest . . . . .	137
8.16.28	CFDiskTest::getRandomRaidLevel . . . . .	137
8.16.29	CFDiskTest::getRandomUnusedMD . . . . .	138
8.16.30	CFDiskTest::getRandomUsedMD . . . . .	138
8.16.31	CFDiskTest::deleteRaid . . . . .	138
8.16.32	CFDiskTest::areThereEnoughFreePartitionsToBuildTheRaid . . . . .	138
8.16.33	CFDiskTest::randomCreateRaid . . . . .	138
8.16.34	CFDiskTest::randomDeleteRaid . . . . .	138
8.16.35	CFDiskTest::randomFormatRaid . . . . .	138
8.16.36	CFDiskTest::randomRaidTest . . . . .	138
8.16.37	CFDiskTest::randomPartTest . . . . .	138
8.17	./inc/CFirewall.php . . . . .	139
8.17.1	CFirewall::__construct . . . . .	139
8.17.2	CFirewall::clearIPTablesSettings . . . . .	139
8.17.3	CFirewall::scriptFileExists . . . . .	139
8.17.4	CFirewall::rulesFileExists . . . . .	139
8.17.5	CFirewall::writeDefaultScript . . . . .	139
8.17.6	CFirewall::getScript . . . . .	139
8.17.7	CFirewall::putScript . . . . .	139
8.17.8	CFirewall::executeScript . . . . .	139
8.17.9	CFirewall::exportRules . . . . .	140
8.17.10	CFirewall::activateInInterfaces . . . . .	140
8.17.11	CFirewall::isActive . . . . .	140
8.17.12	CFirewall::activate . . . . .	140
8.17.13	CFirewall::deactivate . . . . .	140
8.17.14	CFirewall::getSatusIconHTML . . . . .	140
8.17.15	CFirewall::show . . . . .	140
8.18	./inc/CGPGSign.php . . . . .	141
8.18.1	CGPGSign::__construct . . . . .	141
8.18.2	CGPGSign::getKeySelectionDialog . . . . .	141
8.18.3	CGPGSign::showWarningAndDie . . . . .	141
8.18.4	CGPGSign::exportPublicSignKey . . . . .	141

8.18.5	CGPGSign::checkKey	141
8.18.6	CGPGSign::getKeyInfo	141
8.18.7	CGPGSign::gpgSignDetached	142
8.18.8	CGPGSign::gpgSignClear	142
8.18.9	CGPGSign::hasConfigFile	142
8.18.10	CGPGSign::loadConfigFile	142
8.18.11	CGPGSign::writeConfigFile	142
8.18.12	CGPGSign::setGPGID	142
8.18.13	CGPGSign::getGPGID	142
8.18.14	CGPGSign::setStoreMode	143
8.18.15	CGPGSign::getStoreMode	143
8.18.16	CClient::__destruct	143
8.19	/inc/checks.php	144
8.19.1	CHECK_ip	144
8.19.2	CHECK_safeFilename	144
8.19.3	CHECK_text2db	144
8.19.4	CHECK_db2text	144
8.19.5	CHECK_FW	144
8.19.6	CHECK_deviceName	145
8.19.7	CHECK_mointPoint	145
8.19.8	CHECK_int	145
8.19.9	CHECK_float	145
8.19.10	CHECK_strAlpha	146
8.19.11	CHECK_strAlphaNum	146
8.19.12	CHECK_letFWDie	146
8.19.13	CHECK_str	146
8.19.14	countLinesInFile	147
8.19.15	checkIP	147
8.19.16	checkMAC	147
8.19.17	checkNetmask	147
8.19.18	checkEmail	147
8.19.19	checkFQDN	147
8.19.20	checkNormalKeys	147
8.20	/inc/CIPRanges.php	148
8.21	/inc/client_details.php	149
8.21.1	CLIENT_DETAILS_beginCategory	149
8.21.2	CLIENT_DETAILS_endCategory	149
8.21.3	CLIENT_DETAILS_addIcon	149
8.21.4	CLIENT_DETAILS_addIcon2	149
8.22	/inc/client.php	150
8.22.1	CLIENT_insertIntoClientlogs	150
8.22.2	CLIENT_addClientlogsOk	150
8.22.3	CLIENT_addClientlogsFailure	150
8.22.4	CLIENT_addClientlogsUnknown	150
8.22.5	CLIENT_removeServerCache	150
8.22.6	CLIENT_touchLiveLogFile	150
8.22.7	CLIENT_getNextFreeIp	151
8.22.8	CLIENT_getAllAsRes	151
8.22.9	CLIENT_touchLogFile	151
8.22.10	CLIENT_liveLogJobName	151
8.22.11	CLIENT_stopLiveScreenRecording	151
8.22.12	CLIENT_startLiveScreenRecording	151
8.22.13	CLIENT_filterLinesFromLiveScreenRecording	152
8.22.14	CLIENT_getOverviewSearchLine	152
8.22.15	CLIENT_showOverviewSearchDialog	152
8.22.16	CLIENT_extraWebAction	152



8.22.17 CLIENT_getOption . . . . .	152
8.22.18 CLIENT_getDistribution . . . . .	152
8.22.19 CLIENT_runDebconf . . . . .	153
8.22.20 CLIENT_copyDebconfDB . . . . .	153
8.22.21 CLIENT_setDebconfDB . . . . .	153
8.22.22 CLIENT_getDebconfDB . . . . .	153
8.22.23 CLIENT_getDebconfDBValue . . . . .	153
8.22.24 CLIENT_getAllClientNames . . . . .	153
8.22.25 CLIENT_getClientAmount . . . . .	154
8.22.26 CLIENT_getCurrentMemoryUsage . . . . .	154
8.22.27 CLIENT_getCurrentFreeSpaceInDir . . . . .	154
8.22.28 CLIENT_getClientID . . . . .	154
8.22.29 CLIENT_getActiveNetDevices . . . . .	154
8.22.30 CLIENT_executeOnClientOrIP . . . . .	154
8.22.31 CLIENT_isBasesystemInstalledFromImage . . . . .	155
8.22.32 CLIENT_addClient . . . . .	155
8.22.33 CLIENT_IPexists . . . . .	156
8.22.34 CLIENT_MACexists . . . . .	156
8.22.35 CLIENT_exists . . . . .	156
8.22.36 CLIENT_getAskingParams . . . . .	156
8.22.37 CLIENT_getParams . . . . .	156
8.22.38 CLIENT_getClientStatus . . . . .	157
8.22.39 CLIENT_getProperty . . . . .	157
8.22.40 CLIENT_listPackages . . . . .	157
8.22.41 CLIENT_getPossibleActions . . . . .	157
8.22.42 CLIENT_acceptChanges . . . . .	157
8.22.43 CLIENT_showHardwareInfo . . . . .	157
8.22.44 CLIENT_showGeneralInfo . . . . .	157
8.22.45 CLIENT_showWaitingJobs . . . . .	158
8.22.46 CLIENT_setLastmodify . . . . .	158
8.22.47 CLIENT_getNetmaskBits . . . . .	158
8.22.48 CLIENT_getNetmaskFromBitAmount . . . . .	158
8.22.49 CLIENT_getSubnet . . . . .	158
8.22.50 CLIENT_getBroadcast . . . . .	158
8.22.51 CLIENT_convertMac . . . . .	158
8.22.52 CLIENT_getIPbyName . . . . .	159
8.22.53 CLIENT_getNamebyIP . . . . .	159
8.22.54 CLIENT_getMACbyName . . . . .	159
8.22.55 CLIENT_sshFetchJob . . . . .	159
8.22.56 CLIENT_backToRed . . . . .	159
8.22.57 CLIENT_desasterRecovery . . . . .	159
8.22.58 CLIENT_recoverClientCombineJobs . . . . .	159
8.22.59 CLIENT_wol . . . . .	160
8.22.60 CLIENT_recalculateStatusBar . . . . .	160
8.22.61 CLIENT_resetStatusBar . . . . .	160
8.22.62 CLIENT_startInstall . . . . .	160
8.22.63 CLIENT_resetAndInstall . . . . .	160
8.22.64 CLIENT_getBootType . . . . .	160
8.22.65 CLIENT_isrunning . . . . .	160
8.22.66 CLIENT_reset . . . . .	160
8.22.67 CLIENT_showLastLogError . . . . .	160
8.22.68 CLIENT_showLog . . . . .	161
8.22.69 CLIENT_getClientName . . . . .	161
8.22.70 CLIENT_getAllOptions . . . . .	161
8.22.71 CLIENT_setAllOptions . . . . .	161
8.22.72 CLIENT_getAllAskingOptions . . . . .	161

8.22.73	CLIENT_getSetOption . . . . .	161
8.22.74	CLIENT_options2HiddenForm . . . . .	161
8.22.75	CLIENT_hiddenForm2options . . . . .	161
8.22.76	CLIENT_getStatusimage . . . . .	161
8.22.77	CLIENT_showStatusSelection . . . . .	162
8.22.78	CLIENT_listCriticalClients . . . . .	162
8.22.79	CLIENT_isInDebugMode . . . . .	162
8.22.80	CLIENT_toggleDebugMode . . . . .	162
8.22.81	CLIENT_getStatusimage . . . . .	162
8.22.82	CLIENT_generateHTMLStatusBar . . . . .	162
8.22.83	CLIENT_showDebugSelection . . . . .	162
8.22.84	CLIENT_isInRescueMode . . . . .	162
8.22.85	CLIENT_showDirectConnectionHelp . . . . .	163
8.22.86	CLIENT_isInDebugMode . . . . .	163
8.22.87	CLIENT_getToDetailsURL . . . . .	163
8.22.88	CLIENT_HTMLBackToDetails . . . . .	163
8.22.89	CLIENT_getId . . . . .	163
8.22.90	CLIENT_query . . . . .	163
8.22.91	CLIENT_addChangeElement . . . . .	164
8.22.92	CLIENT_showDelDialog . . . . .	164
8.22.93	CLIENT_showAddDialog . . . . .	164
8.22.94	CLIENT_deleteClient . . . . .	164
8.22.95	CLIENT_getNames . . . . .	164
8.22.96	CLIENT_getNamesWithPackages . . . . .	164
8.22.97	CLIENT_changeClient . . . . .	164
8.22.98	CLIENT_setAllParams . . . . .	165
8.22.99	CLIENT_plinkFetchJob . . . . .	165
8.22.100	CLIENT_isAssimilated . . . . .	165
8.22.101	CLIENT_isDedicatedAndReachable . . . . .	165
8.22.102	CLIENT_generateHTMLDedicatedAndReachableStatus . . . . .	165
8.22.103	CLIENT_sendPlymouthMessage . . . . .	165
8.22.104	CLIENT_sendNotifySendMessage . . . . .	166
8.23	./inc/Cm23AdminLister.php . . . . .	167
8.23.1	Cm23AdminLister::AdminExistsDB . . . . .	167
8.23.2	Cm23AdminLister::AdminExistsHt . . . . .	167
8.23.3	Cm23AdminLister::CountAdmins . . . . .	167
8.23.4	Cm23AdminLister::ListAdmins . . . . .	167
8.24	./inc/Cm23Admin.php . . . . .	168
8.24.1	Cm23Admin::__construct . . . . .	168
8.24.2	Cm23Admin::__destruct . . . . .	168
8.24.3	Cm23Admin::delete . . . . .	168
8.24.4	Cm23Admin::changePw . . . . .	168
8.24.5	Cm23Admin::setCSS . . . . .	168
8.24.6	Cm23Admin::getCSS . . . . .	168
8.24.7	Cm23Admin::setLanguage . . . . .	169
8.24.8	Cm23Admin::getLanguage . . . . .	169
8.24.9	Cm23Admin::addToDB . . . . .	169
8.24.10	Cm23Admin::addToPwFiles . . . . .	169
8.24.11	Cm23Admin::delFromPwFiles . . . . .	169
8.25	./inc/CMessageManager.php . . . . .	170
8.25.1	CMessageManager::addMessage . . . . .	170
8.25.2	CMessageManager::addInfoMessage . . . . .	170
8.25.3	CMessageManager::addWarningMessage . . . . .	170
8.25.4	CMessageManager::addErrorMessage . . . . .	170
8.25.5	CMessageManager::popInfoMessagesHTML . . . . .	170
8.25.6	CMessageManager::popWarningMessagesHTML . . . . .	170

8.25.7	CMessageManager::popErrorMessagesHTML	170
8.25.8	CMessageManager::deleteAllMessages	171
8.25.9	CMessageManager::hasInfos	171
8.25.10	CMessageManager::hasWarnings	171
8.25.11	CMessageManager::hasErrors	171
8.25.12	CMessageManager::showInfo	171
8.25.13	CMessageManager::showWarning	171
8.25.14	CMessageManager::showError	171
8.25.15	CMessageManager::showMessages	171
8.26	./inc/CObjectStorageManager.php	172
8.26.1	CObjectStorageManager::saveObject	172
8.26.2	CObjectStorageManager::getAllObjectsByRes	172
8.26.3	CObjectStorageManager::getByIdent	172
8.26.4	CObjectStorageManager::getClass	172
8.26.5	CObjectStorageManager::getByIdentClass	172
8.26.6	CObjectStorageManager::deleteObject	173
8.26.7	CObjectStorageManager::getObjectMethodReturnValue	173
8.26.8	CObjectStorageManager::getCOSStatus	173
8.26.9	CObjectStorageManager::getCOSStatusHumanReadable	173
8.26.10	CObjectStorageManager::getCOSI18NVariable	173
8.26.11	CObjectStorageManager::getCOSDescription	174
8.26.12	CObjectStorageManager::getCOSName	174
8.26.13	CObjectStorageManager::showList	174
8.27	./inc/CObjectStorage.php	175
8.27.1	CObjectStorage::statusCodeToHumanReadable	175
8.27.2	CObjectStorage::__construct	175
8.27.3	CObjectStorage::__destruct	175
8.27.4	CObjectStorage::getObject	175
8.27.5	CObjectStorage::destroy	175
8.28	./inc/CPackageStatusCompare.php	176
8.28.1	CPackageStatusCompare::__construct	176
8.28.2	CPackageStatusCompare::__destruct	176
8.28.3	CPackageStatusCompare::save	176
8.28.4	CPackageStatusCompare::setFile	176
8.28.5	CPackageStatusCompare::setFile1	176
8.28.6	CPackageStatusCompare::setFile2	176
8.28.7	CPackageStatusCompare::setClient	176
8.28.8	CPackageStatusCompare::setClient1	177
8.28.9	CPackageStatusCompare::setClient2	177
8.28.10	CPackageStatusCompare::isClientSet	177
8.28.11	CPackageStatusCompare::isClient1Set	177
8.28.12	CPackageStatusCompare::isClient2Set	177
8.28.13	CPackageStatusCompare::isFile1Set	177
8.28.14	CPackageStatusCompare::isFile2Set	177
8.28.15	CPackageStatusCompare::getVersionStatus	177
8.28.16	CPackageStatusCompare::getStatusFile	178
8.28.17	CPackageStatusCompare::showStatusRow	178
8.28.18	CPackageStatusCompare::diff	178
8.28.19	CPackageStatusCompare::show	178
8.29	./inc/CPoolFromClientDebsGUI.php	179
8.29.1	CPoolFromClientDebsGUI::__construct	179
8.29.2	CPoolFromClientDebsGUI::addm23BuildPoolFromClientDebsJob	179
8.29.3	CPoolFromClientDebsGUI::DEFINE_checkboxForAddingm23BuildPoolFromClientDebsPackage	179
8.30	./inc/CPoolFromClientGUI.php	180
8.30.1	CPoolFromClientGUI::DEFINE_checkboxForAddingm23BuildPoolFromClientPackage	180
8.30.2	CPoolFromClientGUI::setImportPoolName	180

8.30.3	CPoolFromClientGUI::getImportPoolName . . . . .	180
8.30.4	CPoolFromClientGUI::__construct . . . . .	180
8.30.5	CPoolFromClientGUI::getCOSStatusHumanReadable . . . . .	180
8.30.6	CPoolFromClientGUI::getCOSStatus . . . . .	180
8.30.7	CPoolFromClientGUI::runCOSLoop . . . . .	180
8.30.8	CPoolFromClientGUI::saveInObjectStorage . . . . .	180
8.31	/inc/CPoolGUI.php . . . . .	181
8.31.1	CPoolGUI::show . . . . .	181
8.31.2	CPoolGUI::getHelpPage . . . . .	181
8.31.3	CPoolGUI::getHeading . . . . .	181
8.31.4	POOL_showReadCD . . . . .	181
8.31.5	CPoolGUI::DEFINE_convertPackagesToRepositoryStatus . . . . .	181
8.31.6	CPoolGUI::DIALOG_convertPackagesToRepositoryStatus . . . . .	181
8.31.7	CPoolGUI::DEFINE_showDownloadStatus . . . . .	181
8.31.8	CPoolGUI::DIALOG_showDownloadStatus . . . . .	182
8.31.9	CPoolGUI::DIALOG_start . . . . .	182
8.31.10	CPoolGUI::DEFINE_nextStepCopyDownloadPackages . . . . .	182
8.31.11	CPoolGUI::DEFINE_updatePackageIndexAndReSignPool . . . . .	182
8.31.12	CPoolGUI::DEFINE_changePoolDescription . . . . .	182
8.31.13	CPoolGUI::DEFINE_loadDeletePool . . . . .	182
8.31.14	CPoolGUI::DEFINE_createBasicPool . . . . .	182
8.32	/inc/CPoolLister.php . . . . .	183
8.33	/inc/CPool.php . . . . .	184
8.33.1	CPool::__construct . . . . .	184
8.33.2	CPool::signRelease . . . . .	184
8.33.3	CPool::getConvertPackagesToRepositoryLogName . . . . .	184
8.33.4	CPool::getConvertPackagesToRepositoryLogNewLines . . . . .	184
8.33.5	CPool::isConvertPackagesToRepositoryRunning . . . . .	184
8.33.6	CPool::convertPackagesToRepository . . . . .	184
8.33.7	CPool::isPackageTaskRunning . . . . .	185
8.33.8	CPool::isDownloadRunning . . . . .	185
8.33.9	CPool::resetDownloadLog . . . . .	185
8.33.10	CPool::getDownloadLogNewLines . . . . .	185
8.33.11	CPool::getDownloadLogContents . . . . .	185
8.33.12	CPool::getPoolImportedFromSourceslist . . . . .	185
8.33.13	CPool::setPoolImportedFromSourceslist . . . . .	185
8.33.14	CPool::hasPoolDownloadBasePackages . . . . .	185
8.33.15	CPool::setPoolDownloadBasePackages . . . . .	186
8.33.16	CPool::getPoolImportedPackageList . . . . .	186
8.33.17	CPool::setPoolImportedPackageList . . . . .	186
8.33.18	CPool::getPoolDistribution . . . . .	186
8.33.19	CPool::setPoolDistribution . . . . .	186
8.33.20	CPool::createBasicPool . . . . .	186
8.33.21	CPool::getPoolDir . . . . .	186
8.33.22	CPool::setPoolName . . . . .	186
8.33.23	CPool::getPoolName . . . . .	187
8.33.24	CPool::getPoolSourceslist . . . . .	187
8.33.25	CPool::setPoolSourceslist . . . . .	187
8.33.26	CPool::getPoolDescription . . . . .	187
8.33.27	CPool::setPoolDescription . . . . .	187
8.33.28	CPool::getPoolRelease . . . . .	187
8.33.29	CPool::setPoolRelease . . . . .	187
8.33.30	CPool::getPoolType . . . . .	187
8.33.31	CPool::setPoolType . . . . .	188
8.33.32	CPool::getPoolArch . . . . .	188
8.33.33	CPool::setPoolArch . . . . .	188

8.33.34	CPool::setProperty . . . . .	188
8.33.35	CPool::getProperty . . . . .	188
8.33.36	CPool::getPoolSize . . . . .	188
8.33.37	CPool::destroyPool . . . . .	188
8.33.38	CPool::checkForDistributionSpecificPackageFunction . . . . .	189
8.33.39	CPool::preparePool . . . . .	189
8.33.40	CPool::stopDownloadToPool . . . . .	189
8.33.41	CPool::downloadDebsFromClient . . . . .	189
8.33.42	CPool::startDownloadToPool . . . . .	189
8.33.43	CPool::getPoolStatusFileName . . . . .	189
8.33.44	CPool::addPoolStatusFileCommand . . . . .	190
8.33.45	CPool::getPoolStatusFileContents . . . . .	190
8.33.46	CPool::downloadPackagesAndCreatePool . . . . .	190
8.34	/inc/cron.php . . . . .	191
8.34.1	CRON_genCronEntry . . . . .	191
8.34.2	CRON_addJobMinutely . . . . .	191
8.34.3	CRON_addJobBasic . . . . .	191
8.34.4	CRON_rmJob . . . . .	192
8.34.5	CRON_reloadConfig . . . . .	192
8.34.6	CRON_addJobHourly . . . . .	192
8.34.7	CRON_addJobDayly . . . . .	192
8.34.8	CRON_addJobWeekly . . . . .	192
8.34.9	CRON_getTimeBaseArray . . . . .	193
8.34.10	CRON_getDayOfWeekArray . . . . .	193
8.34.11	CRON_checkMinute . . . . .	193
8.34.12	CRON_checkHour . . . . .	193
8.34.13	CRON_isEntryPresent . . . . .	193
8.34.14	CRON_getEntriesByIdentifier . . . . .	194
8.34.15	CRON_getNextIdentifierNr . . . . .	194
8.34.16	CRON_cronManagementDialog . . . . .	194
8.34.17	CRON_cronEntryDeletionDialog . . . . .	194
8.34.18	CRON_translateEveryIntervallValue . . . . .	194
8.35	/inc/CScredit.php . . . . .	195
8.35.1	CScredit::__construct . . . . .	195
8.35.2	CScredit::setCurrentScriptFilename . . . . .	195
8.35.3	CScredit::getCurrentScriptFilename . . . . .	195
8.35.4	CScredit::getCurrentScriptFilenameFullPath . . . . .	195
8.35.5	CScredit::getCurrentScriptFilenameWithoutInstallPhp . . . . .	195
8.35.6	CScredit::getLocalScriptFilenames . . . . .	195
8.35.7	CScredit::updateOnlineScriptInfo . . . . .	195
8.35.8	CScredit::getNextOnlineScriptInfo . . . . .	196
8.35.9	CScredit::loadOnlineScript . . . . .	196
8.35.10	CScredit::getOnlineScriptDialog . . . . .	196
8.35.11	CScredit::isNotSaved . . . . .	196
8.35.12	CScredit::getNewScriptTemplate . . . . .	196
8.35.13	CScredit::uploadScript . . . . .	196
8.35.14	CScredit::getCurrentScript . . . . .	197
8.35.15	CScredit::setCurrentScript . . . . .	197
8.35.16	CScredit::deleteCurrentScript . . . . .	197
8.35.17	CScredit::getViewScriptOutputDialog . . . . .	197
8.35.18	CScredit::saveScript . . . . .	197
8.35.19	CScredit::show . . . . .	197
8.36	/inc/CSsystemProxy.php . . . . .	198
8.36.1	CSYSTEMPROXY_getProxySettingsFromAPT . . . . .	198
8.36.2	CSYSTEMPROXY_addCurlProxySettings . . . . .	198
8.36.3	CSYSTEMPROXY_getUserPasswordString . . . . .	198

8.36.4	CSYSTEMPROXY_getEnvironmentVariables	198
8.36.5	CSYSTEMPROXY_getAptGetProxyParamter	198
8.36.6	CSystemProxy::__construct	198
8.36.7	CSystemProxy::writeEtcProfiles	198
8.36.8	CSystemProxy::writeAPTProxyConf	199
8.36.9	CSystemProxy::writeSquidConf	199
8.36.10	CSystemProxy::save	199
8.36.11	CSystemProxy::showProxyDialog	199
8.36.12	CSystemProxy::loadProxySettings	199
8.36.13	CSystemProxy::isProxyActive	199
8.36.14	CSystemProxy::setProxyActive	199
8.36.15	CSystemProxy::areProxySettingsAvailable	199
8.36.16	CSystemProxy::setProxyHost	199
8.36.17	CSystemProxy::getProxyHost	199
8.36.18	CSystemProxy::getProxyHostIP	200
8.36.19	CSystemProxy::usesUserPassword	200
8.36.20	CSystemProxy::setProxyUser	200
8.36.21	CSystemProxy::getProxyUser	200
8.36.22	CSystemProxy::setProxyPassword	200
8.36.23	CSystemProxy::getProxyPassword	200
8.36.24	CSystemProxy::setProxyScheme	200
8.36.25	CSystemProxy::getProxyScheme	200
8.36.26	CSystemProxy::setProxyPort	200
8.36.27	CSystemProxy::getProxyPort	201
8.36.28	CSystemProxy::getUserPasswordString	201
8.37	./inc/db.php	202
8.37.1	ip2longSafe	202
8.37.2	getArchList	202
8.37.3	isMySQL3used	202
8.37.4	encryptShadow	202
8.37.5	getClientLanguage	202
8.37.6	getInstDev	202
8.37.7	DB_getConnection	202
8.37.8	DB_setConnection	202
8.37.9	DB_isConnectionValid	203
8.37.10	dbConnect	203
8.37.11	dbClose	203
8.37.12	DB_getSuperUserName	203
8.37.13	DB_getSuperUserPassword	203
8.37.14	getServerIP	203
8.37.15	getServerNetmask	203
8.37.16	getServerNetwork	203
8.37.17	getDNSServers	203
8.37.18	getServerGateway	203
8.37.19	sendClientStatus	204
8.37.20	sendClientStageStatus	204
8.37.21	returnClientStageStatus	204
8.37.22	sendClientLogStatus	204
8.37.23	deleteClientLogs	204
8.37.24	workPhpName	204
8.37.25	executeNextWork	204
8.37.26	DB_query	204
8.37.27	DB_queryNoDie	205
8.37.28	DB_genPassword	205
8.37.29	getClientIP	205
8.37.30	implodeAssoc	205

8.37.31	explodeAssoc . . . . .	205
8.37.32	sedSearchReplace . . . . .	205
8.37.33	isProgrammInstalled . . . . .	205
8.37.34	pingIP . . . . .	205
8.37.35	delFromArray . . . . .	206
8.37.36	delValuesFromArray . . . . .	206
8.37.37	DB_getLikeableColumns . . . . .	206
8.37.38	DB_getArrayAssoc . . . . .	206
8.37.39	DB_getTableColumns . . . . .	206
8.37.40	DB_changeAllCollations . . . . .	206
8.38	./inc/dhcp.php . . . . .	207
8.38.1	DHCP_exportDHCPSettingsForExternalDHCPserver . . . . .	207
8.38.2	DHCP_lineNumberAfterLastClient . . . . .	207
8.38.3	DHCP_addSubnetDefinition . . . . .	207
8.38.4	DHCP_delSubnetDefinition . . . . .	207
8.38.5	DHCP_addDynamicRange . . . . .	207
8.38.6	DHCP_getDynamicRanges . . . . .	207
8.38.7	DHCP_getNetmaskOfDynamicRanges . . . . .	208
8.38.8	DHCP_delDynamicRange . . . . .	208
8.38.9	DHCP_bootTypeToNewFormat . . . . .	208
8.38.10	DHCP_runScript . . . . .	208
8.38.11	DHCP_addClient . . . . .	209
8.38.12	DHCP_addLineToDHCPDConf . . . . .	209
8.38.13	DHCP_restartDHCPserver . . . . .	209
8.38.14	DHCP_rmClient . . . . .	209
8.38.15	DHCP_setBootimage . . . . .	209
8.38.16	DHCP_activateBoot . . . . .	209
8.38.17	DHCP_calcPXEIP . . . . .	210
8.38.18	DHCP_writePXEcfg . . . . .	210
8.38.19	DHCP_removePXEcfg . . . . .	210
8.38.20	DHCP_isNetworkBootingActive . . . . .	210
8.39	./inc/distr/debian/clientConfigCommon.php . . . . .	211
8.39.1	CLCFG_setSystime . . . . .	211
8.39.2	CLCFG_disableAptSystemdDaily . . . . .	211
8.39.3	CLCFG_addPAMtoDM . . . . .	211
8.39.4	CLCFG_disableSudoRootLogin . . . . .	211
8.39.5	CLCFG_installFirmware . . . . .	211
8.39.6	CLCFG_monoRemove . . . . .	211
8.39.7	CLCFG_setDebConfDM . . . . .	211
8.39.8	CLCFG_installXDM . . . . .	211
8.39.9	CLCFG_installSDDM . . . . .	211
8.39.10	CLCFG_installMintDM . . . . .	212
8.39.11	CLCFG_copyMBRToAllDevices . . . . .	212
8.39.12	CLCFG_activateBOOT_DEGRADED . . . . .	212
8.39.13	CLCFG_installLightDM . . . . .	212
8.39.14	CLCFG_setDebConfDirect . . . . .	212
8.39.15	TRINITY_installLoginManager . . . . .	212
8.39.16	TRINITY_install . . . . .	212
8.39.17	CLCFG_makeDev . . . . .	212
8.39.18	CLCFG_disablePlymouth . . . . .	212
8.39.19	CLCFG_disableAvahiDaemon . . . . .	213
8.39.20	CLCFG_configUpstartForNormalUsage . . . . .	213
8.39.21	CLCFG_configUpstartForChroot . . . . .	213
8.39.22	CLCFG_createScreenRC . . . . .	213
8.39.23	CLCFG_addGrubPassword . . . . .	213
8.39.24	CLCFG_addLiloPassword . . . . .	213

8.39.25	GNOME_prepare	213
8.39.26	LXDE_install	213
8.39.27	GNOME3_install	213
8.39.28	GNOME_install	214
8.39.29	GNOME_installLoginManager	214
8.39.30	KDE_prepare	214
8.39.31	KDE_install	214
8.39.32	KDE_installLoginManager	214
8.39.33	CLCFG_lilo2Grub	214
8.39.34	CLCFG_getMbrPart	214
8.39.35	CLCFG_setTimeZone	214
8.39.36	CLCFG_writeHosts	215
8.39.37	CLCFG_addUser	215
8.39.38	CLCFG_createBootDeviceNode	215
8.39.39	CLCFG_efi	215
8.39.40	CLCFG_genFstab	215
8.39.41	CLCFG_interfaces	215
8.39.42	CLCFG_hostname	215
8.39.43	CLCFG_resolvConf	216
8.39.44	CLCFG_aptConf	216
8.39.45	CLCFG_sourceslist	216
8.39.46	CLCFG_sourceslistCreateConfigFiles	216
8.39.47	CLCFG_hwdetect	216
8.39.48	CLCFG_language	216
8.39.49	CLCFG_setRootPassword	216
8.39.50	CLCFG_setAuthorized_keys	217
8.39.51	CLCFG_copyClientPackageStatus	217
8.39.52	CLCFG_aptGet	217
8.39.53	CLCFG_importLocalPoolKey	217
8.39.54	CLCFG_installBasePackages	217
8.39.55	CLCFG_setDebconf	217
8.39.56	CLCFG_debootstrap	218
8.39.57	CLCFG_downgradeExt	218
8.39.58	CLCFG_mountRootDir	218
8.39.59	CLCFG_activateDMA	218
8.39.60	CLCFG_fetchm23BasicTools	218
8.39.61	CLCFG_dialogGaugeProcPos	219
8.39.62	CLCFG_dialogInfoBox	219
8.39.63	CLCFG_dialogMsgBox	219
8.39.64	CLCFG_dialogAllBox	219
8.39.65	CLCFG_executeAfterChroot	219
8.39.66	CLCFG_writeCrontabm23fetchjobEvery5Minutes	220
8.39.67	CLCFG_writeM23fetchjob	220
8.39.68	CLCFG_hideKernelWarnings	220
8.39.69	CLCFG_getRootDeviceFS	220
8.39.70	CLCFG_genFakeFstab	220
8.39.71	CLCFG_copySSLCert	220
8.39.72	CLCFG_changeUser	220
8.39.73	CLCFG_patchNsswitchForLDAP	221
8.39.74	CLCFG_enableNFSHome	221
8.39.75	CLCFG_disableNFSHome	221
8.39.76	CLCFG_installDesktopLanguagePackage	221
8.39.77	CLCFG_installApplicationLanguagePackages	221
8.39.78	CLCFG_updateDebootstrapScripts	221
8.39.79	CLCFG_updateDebootstrapScriptsDebian	221
8.40	./inc/distr/debian/clientConfig.php	222



8.40.1	DEBIAN_desktopInstall . . . . .	222
8.40.2	CLCFG_listDebianReleasesGeneric . . . . .	222
8.40.3	CLCFG_listDebianReleasesGeneric . . . . .	222
8.40.4	CLCFG_listDebianReleases . . . . .	222
8.40.5	CLCFG_addDistributionSpecificOptions . . . . .	222
8.40.6	CLCFG_showDistributionSpecificOptions . . . . .	223
8.40.7	CLIENT_enableLDAP . . . . .	223
8.41	./inc/distr/debian/packages.php . . . . .	224
8.41.1	PKG_updatePackageSearchCacheFile . . . . .	224
8.41.2	PKG_getSearchCacheFileName . . . . .	224
8.41.3	PKG_genPackageSearchCacheFileCMD . . . . .	224
8.41.4	PKG_searchPackageSearchCacheFileCMD . . . . .	224
8.41.5	PKG_ncTarDebsFromClientToServer_Client . . . . .	224
8.41.6	PKG_ncTarDebsFromClientToServer_Server . . . . .	225
8.41.7	PKG_rsyncDebsFromClientToServer . . . . .	225
8.41.8	PKG_preparePool . . . . .	225
8.41.9	PKG_convertPackagesToRepository . . . . .	225
8.41.10	PKG_fastGetInstalledPackages . . . . .	225
8.41.11	PKG_fastGetNewInstalledPackages . . . . .	225
8.41.12	PKG_searchFor . . . . .	226
8.41.13	PKG_getDescription . . . . .	226
8.41.14	PKG_getPackageName . . . . .	226
8.41.15	PKG_getLongPackageDescription . . . . .	226
8.41.16	PKG_getPackageDescriptionSize . . . . .	226
8.41.17	PKG_printStatus . . . . .	226
8.41.18	PKG_listPackages . . . . .	227
8.41.19	PKG_getAptArchOptions . . . . .	227
8.41.20	PKG_addAPTConfigFiles . . . . .	227
8.41.21	PKG_preparePackageDir . . . . .	227
8.41.22	PKG_updatePackageInfo . . . . .	228
8.41.23	PKG_previewInstall . . . . .	228
8.41.24	PKG_getKernels . . . . .	228
8.41.25	PKG_downloadPool . . . . .	228
8.41.26	PKG_getDebootStrapBasePackages . . . . .	228
8.42	./inc/distr/halfSister/clientConfig.php . . . . .	229
8.42.1	CLCFG_copyClientPackageStatus . . . . .	229
8.42.2	CLCFG_listHSReleasesGeneric . . . . .	229
8.42.3	CLCFG_listHSReleasesGeneric . . . . .	229
8.42.4	CLCFG_listHSReleases . . . . .	229
8.42.5	CLCFG_addDistributionSpecificOptions . . . . .	229
8.42.6	CLCFG_showDistributionSpecificOptions . . . . .	229
8.43	./inc/distr/halfSister/packages.php . . . . .	230
8.43.1	PKG_updatePackageSearchCacheFile . . . . .	230
8.43.2	PKG_fastGetInstalledPackages . . . . .	230
8.43.3	pkgUpdateCacheOnServer . . . . .	230
8.43.4	PKG_listPackages . . . . .	230
8.43.5	PKG_previewInstall . . . . .	231
8.43.6	PKG_getKernels . . . . .	231
8.43.7	PKG_translateClientPackageStatus . . . . .	231
8.44	./inc/distr/imaging/clientConfigCommon.php . . . . .	232
8.44.1	CLCFG_setSystime . . . . .	232
8.44.2	CLCFG_disableAptSystemdDaily . . . . .	232
8.44.3	CLCFG_addPAMtoDM . . . . .	232
8.44.4	CLCFG_disableSudoRootLogin . . . . .	232
8.44.5	CLCFG_installFirmware . . . . .	232
8.44.6	CLCFG_monoRemove . . . . .	232

8.44.7	CLCFG_setDebConfDM	232
8.44.8	CLCFG_installXDM	232
8.44.9	CLCFG_installSDDM	232
8.44.10	CLCFG_installMintDM	233
8.44.11	CLCFG_copyMBRToAllDevices	233
8.44.12	CLCFG_activateBOOT_DEGRADED	233
8.44.13	CLCFG_installLightDM	233
8.44.14	CLCFG_setDebConfDirect	233
8.44.15	TRINITY_installLoginManager	233
8.44.16	TRINITY_install	233
8.44.17	CLCFG_makeDev	233
8.44.18	CLCFG_disablePlymouth	233
8.44.19	CLCFG_disableAvahiDaemon	234
8.44.20	CLCFG_configUpstartForNormalUsage	234
8.44.21	CLCFG_configUpstartForChroot	234
8.44.22	CLCFG_createScreenRC	234
8.44.23	CLCFG_addGrubPassword	234
8.44.24	CLCFG_addLiloPassword	234
8.44.25	GNOME_prepare	234
8.44.26	LXDE_install	234
8.44.27	GNOME3_install	234
8.44.28	GNOME_install	235
8.44.29	GNOME_installLoginManager	235
8.44.30	KDE_prepare	235
8.44.31	KDE_install	235
8.44.32	KDE_installLoginManager	235
8.44.33	CLCFG_lilo2Grub	235
8.44.34	CLCFG_getMbrPart	235
8.44.35	CLCFG_setTimeZone	235
8.44.36	CLCFG_writeHosts	236
8.44.37	CLCFG_addUser	236
8.44.38	CLCFG_createBootDeviceNode	236
8.44.39	CLCFG_efi	236
8.44.40	CLCFG_genFstab	236
8.44.41	CLCFG_interfaces	236
8.44.42	CLCFG_hostname	236
8.44.43	CLCFG_resolvConf	237
8.44.44	CLCFG_aptConf	237
8.44.45	CLCFG_sourceslist	237
8.44.46	CLCFG_sourceslistCreateConfigFiles	237
8.44.47	CLCFG_hwdetect	237
8.44.48	CLCFG_language	237
8.44.49	CLCFG_setRootPassword	237
8.44.50	CLCFG_setAuthorized_keys	238
8.44.51	CLCFG_copyClientPackageStatus	238
8.44.52	CLCFG_aptGet	238
8.44.53	CLCFG_importLocalPoolKey	238
8.44.54	CLCFG_installBasePackages	238
8.44.55	CLCFG_setDebconf	238
8.44.56	CLCFG_debootstrap	239
8.44.57	CLCFG_downgradeExt	239
8.44.58	CLCFG_mountRootDir	239
8.44.59	CLCFG_activateDMA	239
8.44.60	CLCFG_fetchm23BasicTools	239
8.44.61	CLCFG_dialogGaugeProcPos	240
8.44.62	CLCFG_dialogInfoBox	240

8.44.63	CLCFG_dialogMsgBox	240
8.44.64	CLCFG_dialogAllBox	240
8.44.65	CLCFG_executeAfterChroot	240
8.44.66	CLCFG_writeCrontabm23fetchjobEvery5Minutes	241
8.44.67	CLCFG_writeM23fetchjob	241
8.44.68	CLCFG_hideKernelWarnings	241
8.44.69	CLCFG_getRootDeviceFS	241
8.44.70	CLCFG_genFakeFstab	241
8.44.71	CLCFG_copySSLCert	241
8.44.72	CLCFG_changeUser	241
8.44.73	CLCFG_patchNsswitchForLDAP	242
8.44.74	CLCFG_enableNFSHome	242
8.44.75	CLCFG_disableNFSHome	242
8.44.76	CLCFG_installDesktopLanguagePackage	242
8.44.77	CLCFG_installApplicationLanguagePackages	242
8.44.78	CLCFG_updateDebootstrapScripts	242
8.44.79	CLCFG_updateDebootstrapScriptsDebian	242
8.45	./inc/distr/imaging/clientConfig.php	243
8.45.1	CLCFG_addDistributionSpecificOptions	243
8.45.2	CLCFG_showDistributionSpecificOptions	243
8.46	./inc/distr/imaging/packages.php	244
8.46.1	PKG_updatePackageSearchCacheFile	244
8.46.2	PKG_getSearchCacheFileName	244
8.46.3	PKG_genPackageSearchCacheFileCMD	244
8.46.4	PKG_searchPackageSearchCacheFileCMD	244
8.46.5	PKG_ncTarDebsFromClientToServer_Client	244
8.46.6	PKG_ncTarDebsFromClientToServer_Server	245
8.46.7	PKG_rsyncDebsFromClientToServer	245
8.46.8	PKG_preparePool	245
8.46.9	PKG_convertPackagesToRepository	245
8.46.10	PKG_fastGetInstalledPackages	245
8.46.11	PKG_fastGetNewInstalledPackages	245
8.46.12	PKG_searchFor	246
8.46.13	PKG_getDescription	246
8.46.14	PKG_getPackageName	246
8.46.15	PKG_getLongPackageDescription	246
8.46.16	PKG_getPackageDescriptionSize	246
8.46.17	PKG_printStatus	246
8.46.18	PKG_listPackages	247
8.46.19	PKG_getAptArchOptions	247
8.46.20	PKG_addAPTConfigFiles	247
8.46.21	PKG_preparePackageDir	247
8.46.22	PKG_updatePackageInfo	248
8.46.23	PKG_previewInstall	248
8.46.24	PKG_getKernels	248
8.46.25	PKG_downloadPool	248
8.46.26	PKG_getDebootstrapBasePackages	248
8.47	./inc/distr.php	249
8.47.1	DISTR_getUbuntuUserGroups	249
8.47.2	DISTR_getDebianUserGroups	249
8.47.3	DISTR_releaseVersionTranslator	249
8.47.4	DISTR_listDistributions	249
8.47.5	DISTR_getDescriptionValues	249
8.47.6	DISTR_DistributionsSelections	250
8.47.7	DISTR_geti18nValue	250
8.47.8	DISTR_listCommaSeperated	250

8.47.9	DISTR_commaSeperatedSelections . . . . .	250
8.47.10	DISTR_getDesktopsCBList . . . . .	250
8.47.11	DISTR_getDesktopDescription . . . . .	250
8.47.12	DISTR_getSelectedDesktopsArr . . . . .	251
8.47.13	DISTR_getSelectedDesktopsStr . . . . .	251
8.48	./inc/distr/ubuntu/clientConfigCommon.php . . . . .	252
8.48.1	CLCFG_setSysime . . . . .	252
8.48.2	CLCFG_disableAptSystemdDaily . . . . .	252
8.48.3	CLCFG_addPAMtoDM . . . . .	252
8.48.4	CLCFG_disableSudoRootLogin . . . . .	252
8.48.5	CLCFG_installFirmware . . . . .	252
8.48.6	CLCFG_monoRemove . . . . .	252
8.48.7	CLCFG_setDebConfDM . . . . .	252
8.48.8	CLCFG_installXDM . . . . .	252
8.48.9	CLCFG_installSDDM . . . . .	252
8.48.10	CLCFG_installMintDM . . . . .	253
8.48.11	CLCFG_copyMBRToAllDevices . . . . .	253
8.48.12	CLCFG_activateBOOT_DEGRADED . . . . .	253
8.48.13	CLCFG_installLightDM . . . . .	253
8.48.14	CLCFG_setDebConfDirect . . . . .	253
8.48.15	TRINITY_installLoginManager . . . . .	253
8.48.16	TRINITY_install . . . . .	253
8.48.17	CLCFG_makeDev . . . . .	253
8.48.18	CLCFG_disablePlymouth . . . . .	253
8.48.19	CLCFG_disableAvahiDaemon . . . . .	254
8.48.20	CLCFG_configUpstartForNormalUsage . . . . .	254
8.48.21	CLCFG_configUpstartForChroot . . . . .	254
8.48.22	CLCFG_createScreenRC . . . . .	254
8.48.23	CLCFG_addGrubPassword . . . . .	254
8.48.24	CLCFG_addLiloPassword . . . . .	254
8.48.25	GNOME_prepare . . . . .	254
8.48.26	LXDE_install . . . . .	254
8.48.27	GNOME3_install . . . . .	254
8.48.28	GNOME_install . . . . .	255
8.48.29	GNOME_installLoginManager . . . . .	255
8.48.30	KDE_prepare . . . . .	255
8.48.31	KDE_install . . . . .	255
8.48.32	KDE_installLoginManager . . . . .	255
8.48.33	CLCFG_lilo2Grub . . . . .	255
8.48.34	CLCFG_getMbrPart . . . . .	255
8.48.35	CLCFG_setTimeZone . . . . .	255
8.48.36	CLCFG_writeHosts . . . . .	256
8.48.37	CLCFG_addUser . . . . .	256
8.48.38	CLCFG_createBootDeviceNode . . . . .	256
8.48.39	CLCFG_efi . . . . .	256
8.48.40	CLCFG_genFstab . . . . .	256
8.48.41	CLCFG_interfaces . . . . .	256
8.48.42	CLCFG_hostname . . . . .	256
8.48.43	CLCFG_resolvConf . . . . .	257
8.48.44	CLCFG_aptConf . . . . .	257
8.48.45	CLCFG_sourceslist . . . . .	257
8.48.46	CLCFG_sourceslistCreateConfigFiles . . . . .	257
8.48.47	CLCFG_hwdetect . . . . .	257
8.48.48	CLCFG_language . . . . .	257
8.48.49	CLCFG_setRootPassword . . . . .	257
8.48.50	CLCFG_setAuthorized_keys . . . . .	258

8.48.51	CLCFG_copyClientPackageStatus . . . . .	258
8.48.52	CLCFG_aprGet . . . . .	258
8.48.53	CLCFG_importLocalPoolKey . . . . .	258
8.48.54	CLCFG_installBasePackages . . . . .	258
8.48.55	CLCFG_setDebconf . . . . .	258
8.48.56	CLCFG_debootstrap . . . . .	259
8.48.57	CLCFG_downgradeExt . . . . .	259
8.48.58	CLCFG_mountRootDir . . . . .	259
8.48.59	CLCFG_activateDMA . . . . .	259
8.48.60	CLCFG_fetchm23BasicTools . . . . .	259
8.48.61	CLCFG_dialogGaugeProcPos . . . . .	260
8.48.62	CLCFG_dialogInfoBox . . . . .	260
8.48.63	CLCFG_dialogMsgBox . . . . .	260
8.48.64	CLCFG_dialogAllBox . . . . .	260
8.48.65	CLCFG_executeAfterChroot . . . . .	260
8.48.66	CLCFG_writeCrontabm23fetchjobEvery5Minutes . . . . .	261
8.48.67	CLCFG_writeM23fetchjob . . . . .	261
8.48.68	CLCFG_hideKernelWarnings . . . . .	261
8.48.69	CLCFG_getRootDeviceFS . . . . .	261
8.48.70	CLCFG_genFakeFstab . . . . .	261
8.48.71	CLCFG_copySSLCert . . . . .	261
8.48.72	CLCFG_changeUser . . . . .	261
8.48.73	CLCFG_patchNsswitchForLDAP . . . . .	262
8.48.74	CLCFG_enableNFSHome . . . . .	262
8.48.75	CLCFG_disableNFSHome . . . . .	262
8.48.76	CLCFG_installDesktopLanguagePackage . . . . .	262
8.48.77	CLCFG_installApplicationLanguagePackages . . . . .	262
8.48.78	CLCFG_updateDebootstrapScripts . . . . .	262
8.48.79	CLCFG_updateDebootstrapScriptsDebian . . . . .	262
8.49	./inc/distr/ubuntu/clientConfig.php . . . . .	263
8.49.1	CLCFG_showDistributionSpecificOptionsUbuntu . . . . .	263
8.49.2	UBUNTU_installLanguagePacks . . . . .	263
8.49.3	UBUNTU_desktopInstall . . . . .	263
8.49.4	UBUNTU_fixBeforeBaseInstall . . . . .	263
8.49.5	UBUNTU_fixAfterBaseInstall . . . . .	263
8.49.6	CLCFG_listUbuntuReleases . . . . .	264
8.49.7	CLCFG_updateDebootstrapScriptsUbuntu . . . . .	264
8.49.8	CLIENT_enableLDAP . . . . .	264
8.50	./inc/distr/ubuntu/packages.php . . . . .	265
8.50.1	PKG_updatePackageSearchCacheFile . . . . .	265
8.50.2	PKG_getSearchCacheFileName . . . . .	265
8.50.3	PKG_genPackageSearchCacheFileCMD . . . . .	265
8.50.4	PKG_searchPackageSearchCacheFileCMD . . . . .	265
8.50.5	PKG_ncTarDebsFromClientToServer_Client . . . . .	265
8.50.6	PKG_ncTarDebsFromClientToServer_Server . . . . .	266
8.50.7	PKG_rsyncDebsFromClientToServer . . . . .	266
8.50.8	PKG_preparePool . . . . .	266
8.50.9	PKG_convertPackagesToRepository . . . . .	266
8.50.10	PKG_fastGetInstalledPackages . . . . .	266
8.50.11	PKG_fastGetNewInstalledPackages . . . . .	266
8.50.12	PKG_searchFor . . . . .	267
8.50.13	PKG_getDescription . . . . .	267
8.50.14	PKG_getPackageName . . . . .	267
8.50.15	PKG_getLongPackageDescription . . . . .	267
8.50.16	PKG_getPackageDescriptionSize . . . . .	267
8.50.17	PKG_printStatus . . . . .	267

8.50.18	PKG_listPackages	268
8.50.19	PKG_getAptArchOptions	268
8.50.20	PKG_addAPTConfigFiles	268
8.50.21	PKG_preparePackageDir	268
8.50.22	PKG_updatePackageInfo	269
8.50.23	PKG_previewInstall	269
8.50.24	PKG_getKernels	269
8.50.25	PKG_downloadPool	269
8.50.26	PKG_getDebootStrapBasePackages	269
8.51	/inc/edit.php	270
8.51.1	EDIT_sedSearchAddInsert	270
8.51.2	EDIT_writeToFile	270
8.51.3	EDIT_setOption	270
8.51.4	EDIT_genClientm23Random	270
8.51.5	EDIT_commentoutInsert	271
8.51.6	EDIT_commentoutAll	271
8.51.7	EDIT_countMatches	271
8.51.8	EDIT_calc	271
8.51.9	EDIT_uncomment	271
8.51.10	EDIT_commentout	271
8.51.11	EDIT_insertAfterLineNumber	272
8.51.12	EDIT_insertAtLineNumber	272
8.51.13	EDIT_insertLineNumber	272
8.51.14	EDIT_searchLineNumber	272
8.51.15	EDIT_searchLastLineNumber	272
8.51.16	EDIT_searchNextLineNumber	273
8.51.17	EDIT_replace	273
8.51.18	EDIT_prepareStr	273
8.51.19	EDIT_savePerms	273
8.51.20	EDIT_restorePerms	273
8.51.21	EDIT_deleteLines	273
8.51.22	EDIT_deleteLinesAmount	273
8.51.23	EDIT_addIfNotExists	274
8.51.24	EDIT_deleteMatching	274
8.51.25	EDIT_replaceLineElseAdd	274
8.51.26	EDIT_appendToFile	274
8.52	/inc/fdisk.php	275
8.52.1	FDISK_showFdiskCombinedGUIFunctions	275
8.52.2	FDISK_showCombinedFdiskGUIDialog	275
8.52.3	FDISK_mdToEndOfArray	275
8.52.4	FDISK_getFstabArray	275
8.52.5	FDISK_findFstabMountPointByDev	275
8.52.6	FDISK_swapFilesystems	275
8.52.7	FDISK_formatInstallAndSwappart	276
8.52.8	FDISK_finalChecksAndRealPartitionAndFormatStart	276
8.52.9	FDISK_installFilesystems	276
8.52.10	FDISK_getUnusedMDs	276
8.52.11	FDISK_listDrivesAndPartitions2	276
8.52.12	FDISK_printAllBars2	276
8.52.13	FDISK_showAllPartTables	276
8.52.14	FDISK_fdiskSessionPartJobs	277
8.52.15	FDISK_fdiskSessionInstPart	277
8.52.16	FDISK_fdiskSessionSwapPart	277
8.52.17	FDISK_fdiskSessionSetter	277
8.52.18	FDISK_fdiskSessionClient	277
8.52.19	FDISK_fdiskSessionParam	277

8.52.20 FDISK_fdiskSessionInstallDrive . . . . .	278
8.52.21 FDISK_fdiskSessionvDevInstall . . . . .	278
8.52.22 FDISK_fdiskSessionFreeSpaces . . . . .	278
8.52.23 FDISK_fdiskSessionReset . . . . .	278
8.52.24 FDISK_fdiskSessionPartMethod . . . . .	278
8.52.25 FDISK_fdiskSessionPage . . . . .	278
8.52.26 FDISK_fdiskSessionHelpPage . . . . .	279
8.52.27 FDISK_fdiskSessionTitle . . . . .	279
8.52.28 FDISK_fdiskSessionFstab . . . . .	279
8.52.29 FDISK_getPartitionByType . . . . .	279
8.52.30 FDISK_getDrivesAndPartitions . . . . .	279
8.52.31 FDISK_listDrivesAndPartitions . . . . .	280
8.52.32 FDISK_selectDrives . . . . .	280
8.52.33 FDISK_printAllBars . . . . .	280
8.52.34 FDISK_getFirstDrive . . . . .	280
8.52.35 FDISK_formatPart . . . . .	280
8.52.36 FDISK_getBiggestValueOf . . . . .	281
8.52.37 FDISK_devNrExists . . . . .	281
8.52.38 FDISK_nextLogicalDevNr . . . . .	281
8.52.39 FDISK_nextPrimaryDevNr . . . . .	281
8.52.40 FDISK_correctLogical . . . . .	281
8.52.41 FDISK_findDevNrPosition . . . . .	281
8.52.42 FDISK_partCreationSelect . . . . .	282
8.52.43 FDISK_canPartTypeBeCreated . . . . .	282
8.52.44 FDISK_checkFreeSpace . . . . .	282
8.52.45 FDISK_installExistingDialog . . . . .	282
8.52.46 FDISK_addFstab . . . . .	282
8.52.47 FDISK_delFstab . . . . .	283
8.52.48 FDISK_listFstab . . . . .	283
8.52.49 FDISK_fstabAddDialog2 . . . . .	283
8.52.50 FDISK_fstabAddDialog . . . . .	283
8.52.51 FDISK_adjustFstabParam . . . . .	283
8.52.52 FDISK_genManualFstab . . . . .	283
8.52.53 FDISK_getBelongingRaidDev . . . . .	284
8.52.54 FDISK_delPart . . . . .	284
8.52.55 FDISK_addPart . . . . .	284
8.52.56 FDISK_listPartitions . . . . .	284
8.52.57 FDISK_definePartitionSelection . . . . .	285
8.52.58 FDISK_getPartitionsFromParam . . . . .	285
8.52.59 FDISK_getAllDrives . . . . .	285
8.52.60 FDISK_colorFS . . . . .	285
8.52.61 FDISK_getPartitionPercent . . . . .	285
8.52.62 FDISK_getAfterPartition . . . . .	286
8.52.63 FDISK_getBeforeFristPartition . . . . .	286
8.52.64 FDISK_getPartitions . . . . .	286
8.52.65 FDISK_getPartInfoString . . . . .	286
8.52.66 FDISK_getDriveInfoString . . . . .	286
8.52.67 FDISK_getDriveInfoIcon . . . . .	287
8.52.68 FDISK_getPartInfoIcon . . . . .	287
8.52.69 FDISK_printBars . . . . .	287
8.52.70 FDISK_getSupportedFS . . . . .	287
8.52.71 FDISK_listSupportedFS . . . . .	287
8.52.72 FDISK_deletePartitionFromParam . . . . .	288
8.52.73 FDISK_virtualDeletePartition . . . . .	288
8.52.74 FDISK_virtualAddPartition . . . . .	288
8.52.75 FDISK_listPartTable . . . . .	288

8.52.76	FDISK_listInstPartSelector . . . . .	288
8.52.77	FDISK_formatExisting . . . . .	289
8.52.78	FDISK_getvPart . . . . .	289
8.52.79	FDISK_rmJob . . . . .	289
8.52.80	FDISK_addJob . . . . .	289
8.52.81	FDISK_bootflagJob . . . . .	289
8.52.82	FDISK_formatJob . . . . .	290
8.52.83	FDISK_countPartitions . . . . .	290
8.52.84	FDISK_getFreeSpaces . . . . .	290
8.52.85	FDISK_autoPart . . . . .	290
8.52.86	FDISK_printColorDefinitions . . . . .	290
8.52.87	FDISK_showDiskDefine . . . . .	290
8.52.88	FDISK_defineDrive . . . . .	291
8.52.89	FDISK_dev2LDevLPart . . . . .	291
8.52.90	FDISK_rereadPartTable . . . . .	291
8.52.91	FDISK_genPartedCommands . . . . .	291
8.52.92	FDISK_listPartJobs . . . . .	292
8.52.93	FDISK_getDiskType . . . . .	292
8.52.94	FDISK_getDriveAndNr . . . . .	292
8.52.95	FDISK_getNextFdiskFormatJobNr . . . . .	292
8.52.96	FDISK_AFPselectDrive . . . . .	292
8.52.97	FDISK_AFPlinearScale . . . . .	292
8.52.98	FDISK_AFPgetPartSizes . . . . .	293
8.52.99	FDISK_adjustFdiskParams . . . . .	293
8.52.100	FDISK_virtualAddDrive . . . . .	293
8.52.101	FDISK_getDrivePartitionSize . . . . .	293
8.52.102	FDISK_listRaidTable . . . . .	293
8.52.103	FDISK_addDrivePartitionToRaid . . . . .	294
8.52.104	FDISK_buildRaidDialog . . . . .	294
8.52.105	FDISK_addRaidJobs . . . . .	294
8.52.106	FDISK_addRaidBeforeFormat . . . . .	294
8.52.107	FDISK_raidJob . . . . .	295
8.52.108	FDISK_virtualDeleteDrive . . . . .	295
8.52.109	FDISK_deleteDriveFromParam . . . . .	295
8.53	./inc/groups.php . . . . .	296
8.53.1	GRP_exists . . . . .	296
8.53.2	GRP_add . . . . .	296
8.53.3	GRP_getIdByName . . . . .	296
8.53.4	GRP_getNameById . . . . .	296
8.53.5	GRP_del . . . . .	296
8.53.6	GRP_isClientInGroup . . . . .	296
8.53.7	GRP_addClientToGroup . . . . .	297
8.53.8	GRP_delClientFromGroup . . . . .	297
8.53.9	GRP_setDescrGroup . . . . .	297
8.53.10	GRP_getDescrGroup . . . . .	297
8.53.11	GRP_listGroupsAndCount . . . . .	297
8.53.12	GRP_showGroupsAndCount . . . . .	297
8.53.13	GRP_ren . . . . .	297
8.53.14	GRP_HTMLBackToDetails . . . . .	297
8.53.15	GRP_countClients . . . . .	298
8.53.16	GRP_showGeneralInfo . . . . .	298
8.53.17	GRP_showRenDialog . . . . .	298
8.53.18	GRP_showChangeDescriptionDialog . . . . .	298
8.53.19	GRP_moveClientToGroup . . . . .	298
8.53.20	GRP_listGroups . . . . .	298
8.53.21	GRP_groupSelection . . . . .	298



8.53.22	GRP_showDelDialog . . . . .	298
8.53.23	GRP_doClientMoreGroups . . . . .	299
8.53.24	GRP_listClientGroups . . . . .	299
8.53.25	GRP_showClientGroups . . . . .	299
8.53.26	GRP_listAllClientsInGroup . . . . .	299
8.53.27	GRP_getDistrsAndSourcesLists . . . . .	299
8.53.28	GRP_showSelDistrSources . . . . .	299
8.53.29	GRP_listAllClientsInGroups . . . . .	300
8.53.30	GRP_HTMLBackToOverview . . . . .	300
8.53.31	GRP_getAllPackages . . . . .	300
8.53.32	GRP_desasterRecovery . . . . .	300
8.54	./inc/halfSister.php . . . . .	301
8.54.1	HS_ASSI_getClientSettingsCommand . . . . .	301
8.54.2	HS_getClientSettingsCommand . . . . .	301
8.54.3	HS_ASSI_statusFileCommand . . . . .	301
8.54.4	HS_ASSI_prepareClient . . . . .	301
8.54.5	HS_getm23HSAdminPath . . . . .	301
8.54.6	HS_getPackageCacheName . . . . .	301
8.54.7	HS_pkgUpdateCacheOnServer . . . . .	302
8.54.8	HS_fetchAndExtractOSImage . . . . .	302
8.54.9	HS_fetchm23HSAdminAndm23hwscannerByOS . . . . .	302
8.54.10	HS_fetchm23HSAdminAndm23hwscanner . . . . .	302
8.54.11	HS_netConfig . . . . .	302
8.54.12	HS_setPackageProxy . . . . .	302
8.54.13	HS_setSourcesList . . . . .	303
8.54.14	HS_normalUpdate . . . . .	303
8.54.15	HS_pkgFullUpdate . . . . .	303
8.54.16	HS_sysSetm23ClientID . . . . .	303
8.54.17	HS_netEnableNTP . . . . .	303
8.54.18	HS_netDisableNTP . . . . .	303
8.54.19	HS_hookBeginAfterChroot . . . . .	303
8.54.20	HS_hookEndAfterChroot . . . . .	303
8.54.21	HS_pkgInstallBasePackages . . . . .	303
8.54.22	HS_netSetm23SSLCertificate . . . . .	304
8.54.23	HS_sysSetLanguage . . . . .	304
8.54.24	HS_sysSetRootPW . . . . .	304
8.54.25	HS_sysSetTimeZone . . . . .	304
8.54.26	HS_sysHWsetup . . . . .	304
8.54.27	HS_sysAddUser . . . . .	304
8.54.28	HS_sysChangeUser . . . . .	304
8.54.29	HS_netEnableSSHDAndImportKey . . . . .	304
8.54.30	HS_writeHosts . . . . .	305
8.54.31	HS_sysWriteM23fetchjob . . . . .	305
8.54.32	HS_sysWriteCrontabm23fetchjobEvery5Minutes . . . . .	305
8.54.33	HS_sysInstallKernel . . . . .	305
8.54.34	HS_netEnableNFSHome . . . . .	305
8.54.35	HS_netEnableLDAP . . . . .	305
8.54.36	HS_sysAddFstabEntries . . . . .	305
8.54.37	HS_sysMakeBootable . . . . .	305
8.54.38	HS_sysConfigurePrinter . . . . .	305
8.54.39	HS_sysInstallPrinter . . . . .	306
8.54.40	HS_pkgInstallKDE . . . . .	306
8.54.41	HS_pkgInstallX . . . . .	306
8.54.42	HS_pkgInstalledList . . . . .	306
8.54.43	HS_pkgInstall . . . . .	306
8.54.44	HS_runClientPackageConfDB . . . . .	306

8.54.45	HS_pkgSearch	306
8.54.46	HS_pkgInstallPreview	306
8.54.47	HS_pkgDeinstallPreview	306
8.54.48	HS_statusFileCommand	307
8.54.49	HS_pkgUpdateCache	307
8.54.50	HS_pkgInstallGnome	307
8.54.51	HS_pkgInstallLXDE	307
8.54.52	HS_pkgInstallXFce	307
8.54.53	HS_pkgDeinstall	307
8.54.54	HS_wrapperHS_wrapperReturn	307
8.54.55	HS_wrapper	307
8.54.56	HS_sendCommandExecutionStatus	307
8.55	/inc/helper.php	308
8.55.1	HELPER_logToFile	308
8.55.2	HELPER_splitDayHourMinuteString	308
8.55.3	HELPER_showScriptHeader	308
8.55.4	HELPER_showScriptFooter	308
8.55.5	HELPER_URIencode	308
8.55.6	HELPER_isUpper	309
8.55.7	HELPER_filesize	309
8.55.8	HELPER_isExecutedInCLI	309
8.55.9	HELPER_isExecutedOnUCS	309
8.55.10	HELPER_getContentFromURL	309
8.55.11	HELPER_trimValue	309
8.55.12	HELPER_xargsRecursive	310
8.55.13	HELPER_resetNewLogLines	310
8.55.14	HELPER_getNewLogLines	310
8.55.15	HELPER_rmRecursive	310
8.55.16	HELPER_showBAfH	310
8.55.17	HELPER_ucrc32	310
8.55.18	HELPER_md5x5	311
8.55.19	HELPER_netmaskAmountOfSetBits	311
8.55.20	HELPER_networkCalculator	311
8.55.21	HELPER_netmaskCalculator	311
8.55.22	HELPER_importAllIntoPOST	311
8.55.23	HELPER_randomUsername	311
8.55.24	HELPER_createSelfSignedCAAndServerCertificate	312
8.55.25	HELPER_arrayReOrderKeynumbers	312
8.55.26	HELPER_arrayInsertBeforeKeynumber	312
8.55.27	HELPER_arrayInsertAfterKeynumber	312
8.55.28	HELPER_m23Array2Array	313
8.55.29	HELPER_array2m23Array	313
8.55.30	print_r2	313
8.55.31	HELPER_debugBacktraceToFile	313
8.55.32	HELPER_getRemoteFileContents	313
8.55.33	HELPER_passGenerator	314
8.55.34	HELPER_array2AssociativeArray	314
8.55.35	HELPER_randomMAC	314
8.55.36	HELPER_generateSalt	314
8.55.37	HELPER_grubMd5Crypt	314
8.55.38	HELPER_listFilesInDir	314
8.55.39	HELPER_getBootLoaders	315
8.55.40	HELPER_getTimeZones	315
8.55.41	HELPER_calcMBSize	315
8.55.42	HELPER_grep	315
8.55.43	HELPER_grepNot	315

8.55.44	HELPER_grepCount . . . . .	316
8.55.45	HELPER_getFdiskMountPoints . . . . .	316
8.55.46	HELPER_getApacheUser . . . . .	316
8.55.47	HELPER_getApacheGroup . . . . .	316
8.55.48	HELPER_putFileContents . . . . .	316
8.55.49	HELPER_getFileContents . . . . .	316
8.55.50	HELPER_showFileContents . . . . .	316
8.55.51	HELPER_maxPhpUploadSize . . . . .	317
8.55.52	HELPER_compareLengthAbc . . . . .	317
8.55.53	HELPER_sortByLength . . . . .	317
8.55.54	HELPER_hostname2IP . . . . .	317
8.56	./inc/help.php . . . . .	318
8.56.1	HELP_showHelp . . . . .	318
8.56.2	HELP_getHelp . . . . .	318
8.56.3	HELP_getHelpString . . . . .	318
8.56.4	HELP_showHelpTex . . . . .	318
8.57	./inc/html.php . . . . .	319
8.57.1	HTML_waitAnimation . . . . .	319
8.57.2	HTML_imgSwitch . . . . .	319
8.57.3	HTML_getOriginalUploadFilename . . . . .	319
8.57.4	HTML_uploadFile . . . . .	320
8.57.5	HTML_urlButton . . . . .	320
8.57.6	HTML_weekdayTimeChooser . . . . .	320
8.57.7	HTML_sourceViewer . . . . .	320
8.57.8	HTML_getOutputBuffer . . . . .	320
8.57.9	HTML_setOutputBuffer . . . . .	320
8.57.10	HTML_AJAXAutoSubmit . . . . .	321
8.57.11	HTML_liveSpan . . . . .	321
8.57.12	HTML_manipulateOutputBuffer . . . . .	321
8.57.13	HTML_showTitle . . . . .	321
8.57.14	HTML_showSmallTitle . . . . .	321
8.57.15	HTML_hiddenVar . . . . .	322
8.57.16	HTML_liveLogArea . . . . .	322
8.57.17	HTML_checkboxChangerButtons . . . . .	322
8.57.18	HTML_jsCheckboxChanger . . . . .	322
8.57.19	HTML_logArea . . . . .	322
8.57.20	HTML_getQuestionnaireURL . . . . .	322
8.57.21	HTML_questionnaire . . . . .	323
8.57.22	HTML_esel . . . . .	323
8.57.23	HTML_JSMenueCloseAllEntries . . . . .	323
8.57.24	HTML_JSMenueOpener . . . . .	323
8.57.25	HTML_jQueryMenu . . . . .	323
8.57.26	HTML_jQueryMenuHeader . . . . .	323
8.57.27	HTML_jQueryReStoreYWindowPosition . . . . .	324
8.57.28	HTML_jQueryMenuEnd . . . . .	324
8.57.29	HTML_incStatusBarPercentByName . . . . .	324
8.57.30	HTML_setStatusBarPercentPointByName . . . . .	324
8.57.31	HTML_setStatusBarStatusByName . . . . .	325
8.57.32	HTML_setStatusBarStatusByID . . . . .	325
8.57.33	HTML_setStatusBarStatus . . . . .	325
8.57.34	HTML_getStatusBarID . . . . .	326
8.57.35	HTML_newStatusBar . . . . .	326
8.57.36	HTML_showStatusBar . . . . .	326
8.57.37	HTML_showStatusBarHTML . . . . .	326
8.57.38	HTML_multiCheckBox . . . . .	327
8.57.39	HTML_multiCheckBoxShow . . . . .	327

8.57.40	HTML_setPage . . . . .	327
8.57.41	HTML_storableInput . . . . .	327
8.57.42	HTML_storableSelection . . . . .	328
8.57.43	HTML_storableMultiSelection . . . . .	328
8.57.44	HTML_storableCheckBox . . . . .	329
8.57.45	HTML_getElementValue . . . . .	329
8.57.46	HTML_listSelection . . . . .	329
8.57.47	HTML_showTableHeader . . . . .	329
8.57.48	HTML_showTableEnd . . . . .	330
8.57.49	HTML_showFormHeader . . . . .	330
8.57.50	HTML_showFormEnd . . . . .	330
8.57.51	HTML_submit . . . . .	330
8.57.52	HTML_submitImg . . . . .	330
8.57.53	HTML_input . . . . .	331
8.57.54	array_makeFirst . . . . .	331
8.57.55	HTML_getValidSelected . . . . .	331
8.57.56	HTML_selection . . . . .	331
8.57.57	HTML_checkBox . . . . .	332
8.57.58	HTML_checkBoxCheckAll . . . . .	332
8.57.59	HTML_submitDefine . . . . .	332
8.57.60	HTML_submitCheck . . . . .	332
8.57.61	HTML_showTableRow . . . . .	332
8.57.62	HTML_showTableHeading . . . . .	333
8.57.63	HTML_textArea . . . . .	333
8.57.64	HTML_showPagePrintButton . . . . .	333
8.58	./inc/hwinfo.php . . . . .	334
8.58.1	HWINFO_getParam . . . . .	334
8.58.2	HWINFO_getMemory . . . . .	334
8.58.3	HWINFO_getHDSize . . . . .	334
8.58.4	HWINFO_printPartitions . . . . .	334
8.58.5	DMI_getAllTextBox . . . . .	334
8.58.6	DMI_getParam . . . . .	334
8.58.7	DMI_getBoard . . . . .	334
8.58.8	DMI_getMemory . . . . .	335
8.58.9	DMI_getCPU . . . . .	335
8.58.10	DMI_getSlot . . . . .	335
8.59	./inc/i18n.php . . . . .	336
8.59.1	I18N_getHumanReadableDayHourMinute . . . . .	336
8.59.2	I18N_getWeekDayArray . . . . .	336
8.59.3	I18N_number_format . . . . .	336
8.59.4	I18N_convertToHumanReadableName . . . . .	336
8.59.5	I18N_m23instLanguage . . . . .	336
8.59.6	I18N_addLanguage . . . . .	337
8.59.7	I18N_listClientLanguages . . . . .	337
8.59.8	I18N_countCachedLanguages . . . . .	337
8.59.9	I18N_cacheWebinterfaceLanguages . . . . .	337
8.59.10	I18N_getAllCachedLanguages . . . . .	337
8.59.11	I18N_listWebinterfaceLanguages . . . . .	338
8.59.12	I18N_addClientLanguageToCache . . . . .	338
8.59.13	I18N_cacheClientLanguages . . . . .	338
8.59.14	I18N_getLangVars . . . . .	338
8.60	./inc/ldap.php . . . . .	339
8.60.1	LDAP_getTypes . . . . .	339
8.60.2	LDAP_connectServer . . . . .	339
8.60.3	LDAP_makeConnection . . . . .	339
8.60.4	LDAP_getValueFromConfigLine . . . . .	339

8.60.5	LDAP_listServers	339
8.60.6	LDAP_loadServer	339
8.60.7	LDAP_addPosix	339
8.60.8	LDAP_fqdn2dn	340
8.60.9	LDAP_installServer	340
8.60.10	LDAP_addServerTophpLdapAdmin	340
8.60.11	LDAP_delServerFromphpLdapAdmin	340
8.60.12	LDAP_checkphpLdapAdminConfiguration	340
8.60.13	LDAP_showServerManagementDialog	340
8.60.14	LDAP_I18NLdapType	341
8.60.15	LDAP_getNextID	341
8.60.16	LDAP_getNextUserID	341
8.60.17	LDAP_addNewID	341
8.60.18	LDAP_deleteID	341
8.60.19	LDAP_addNewUserID	341
8.60.20	LDAP_addNewGroupID	341
8.60.21	LDAP_getNextGroupID	341
8.60.22	LDAP_getFreeIDs	342
8.60.23	LDAP_getFreeUserIDs	342
8.60.24	LDAP_getFreeGroupIDs	342
8.60.25	LDAP_matchLDAPserver	342
8.61	./inc/m23shared/m23shared.php	343
8.61.1	m23SHARED_deleteBills	343
8.61.2	m23SHARED_deleteUserdataFromDB	343
8.61.3	m23SHARED_isMarkedForDeletion	343
8.61.4	m23SHARED_markForDeletion	343
8.61.5	m23SHARED_DBname	343
8.61.6	m23SHARED_getDBnameByInterface	343
8.61.7	m23SHARED_getDBnameByClient	343
8.61.8	m23SHARED_init	343
8.61.9	m23SHARED_new	344
8.61.10	m23SHARED_setLicense	344
8.61.11	m23SHARED_evaluationDaysLeft	344
8.61.12	m23SHARED_evaluationEndDate	344
8.61.13	m23SHARED_paidClients	344
8.61.14	m23SHARED_getCompleteClientName	344
8.61.15	m23SHARED_getServerIP	344
8.61.16	m23SHARED_getPayTypeArray	345
8.61.17	m23SHARED_getSalutationsArray	345
8.61.18	m23SHARED_getSalutationHeadline	345
8.61.19	m23SHARED_getPayTypeHumanReadable	345
8.61.20	m23SHARED_getMonthlyFee	345
8.61.21	m23SHARED_getCustomerEmail	345
8.61.22	m23SHARED_setCustomerEmail	345
8.61.23	m23SHARED_generateActivationKey	345
8.61.24	m23SHARED_getActivationKey	346
8.61.25	m23SHARED_generateCustomerNumber	346
8.61.26	m23SHARED_getCustomerNr	346
8.61.27	m23SHARED_activate	346
8.61.28	m23SHARED_setCustomerLanguage	346
8.61.29	m23SHARED_getCustomerLanguage	346
8.61.30	m23SHARED_sendActivationMail	346
8.61.31	m23SHARED_sendActivationMail	347
8.61.32	m23SHARED_setRealName	347
8.61.33	m23SHARED_getRealName	347
8.61.34	m23SHARED_changeClientAmount	347

8.61.35	m23SHARED_showBill	347
8.61.36	m23SHARED_billDateSelection	347
8.61.37	m23SHARED_priceFormater	347
8.61.38	m23SHARED_calculateBill	348
8.61.39	m23SHARED_getLicenseType	348
8.61.40	m23SHARED_sendAdminMail	348
8.61.41	m23SHARED_getCurrentUser	348
8.61.42	m23SHARED_changePasswordDialog	348
8.61.43	m23SHARED_changePasswordDialog	348
8.61.44	m23SHARED_getBillDates	348
8.61.45	m23SHARED_downloadBillPDFLink	349
8.61.46	m23SHARED_getCustomerAddress	349
8.61.47	m23SHARED_getCustomerBankHTML	349
8.61.48	m23SHARED_pdfBill	349
8.61.49	m23SHARED_setBankAccount	349
8.61.50	m23SHARED_setAddress	349
8.61.51	m23SHARED_unusedPaidClientsAvailable	350
8.61.52	m23SHARED_showLicenseDialog	350
8.61.53	m23SHARED_getAllm23sharedUsers	350
8.61.54	m23SHARED_switchUser	350
8.61.55	m23SHARED_prepareBillMailSending	350
8.61.56	m23SHARED_markBillMailAsSent	350
8.61.57	m23SHARED_sendAllBillMails	350
8.61.58	m23SHARED_addExtraBill	351
8.61.59	m23SHARED_showDonationDialog	351
8.61.60	m23SHARED_showBootMediaDownloadDialog	351
8.61.61	m23SHARED_allUserDBQuery	351
8.61.62	m23SHARED_getUserByResident	351
8.61.63	m23SHARED_getUserByCustomerNr	351
8.61.64	m23SHARED_getUserByInfo	352
8.61.65	m23SHARED_showAdminDialog	352
8.61.66	m23SHARED_blockAccount	352
8.61.67	m23SHARED_isAccountBlocked	352
8.61.68	m23SHARED_showAddExtraBillDialog	352
8.61.69	m23SHARED_showBillDialog	352
8.61.70	m23SHARED_showPriceListTable	352
8.61.71	m23SHARED_getInformationForBootingYourClientLink	352
8.62	./inc/mail.php	353
8.62.1	MAIL_getKeyFromMailAddress	353
8.62.2	MAIL_AESencode	353
8.62.3	MAIL_AESDecode	353
8.62.4	MAIL_sendAESMail	353
8.62.5	MAIL_attach	353
8.62.6	MAIL_getHeader	354
8.62.7	MAIL_gpgSignDetached	354
8.62.8	MAIL_gpgSignClear	354
8.62.9	MAIL_gpgSign	354
8.62.10	MAIL_gpgGetKey	355
8.62.11	MAIL_gpgCheckKey	355
8.62.12	MAIL_gpgMail	355
8.62.13	MAIL_getGpgKeyList	355
8.62.14	MAIL_importGPGKey	355
8.62.15	MAIL_deleteGPGKey	355
8.62.16	MAIL_manageGPGKeysDialog	356
8.62.17	MAIL_sendMail	356
8.62.18	MAIL_cryptMailServer	356

8.63	./inc/massTools.php	357
8.63.1	MASS_EGKradioBoxes	357
8.63.2	MASS_FHradioBoxes	357
8.63.3	MASS_showFileHandDialog	357
8.63.4	MASS_propertyKeys	357
8.63.5	MASS_showFileFormatDialog	357
8.63.6	MASS_keyToI18N	357
8.63.7	MASS_I18NTokey	357
8.63.8	MASS_showTableDefinition	358
8.63.9	MASS_checkAndSaveFields	358
8.63.10	MASS_openDBFile	358
8.63.11	MASS_readDBFile	358
8.63.12	MASS_readDBFileRaw	358
8.63.13	MASS_closeDBFile	358
8.63.14	MASS_getXProperties	358
8.63.15	MASS_showGeneratorOptions	359
8.63.16	MASS_passGenerator	359
8.63.17	MASS_loginGenerator	359
8.63.18	MASS_ipGenerator	359
8.63.19	MASS_minMaxIP	359
8.63.20	MASS_generateNetmask	360
8.63.21	MASS_generateClientNames	360
8.63.22	MASS_saveGeneratorOptions	360
8.63.23	MASS_showOverview	360
8.63.24	MASS_getAllFromFile	360
8.63.25	MASS_getLongestLength	360
8.63.26	MASS_startInstall	360
8.64	./inc/menu.php	361
8.64.1	MENU_showEntry	361
8.64.2	MENU_startGroup	361
8.64.3	MENU_endGroup	361
8.65	./inc/message.php	362
8.65.1	MSG_showMessageBoxPlaceholder	362
8.65.2	MSG_placeOrReturnMessageBox	362
8.65.3	MSG_getm23UpdateFeed	362
8.65.4	MSG_getm23DevelopmentBlog	362
8.65.5	MSG_getRSSFeed	362
8.65.6	MSG_showInfo	363
8.65.7	MSG_showError	363
8.65.8	MSG_showWarning	363
8.65.9	MSG_show	363
8.65.10	MSG_showMessageBoxHeader	363
8.65.11	MSG_showMessageBoxFooter	363
8.65.12	MSG_showMessageBox	364
8.65.13	MSG_showUpdateInfo	364
8.65.14	MSG_showUpdateInfo	364
8.65.15	MSG_showNewFeature	364
8.65.16	MSG_DeActivateBlogDialog	364
8.66	./inc/messageReceive.php	365
8.66.1	MSR_decodeMessage	365
8.66.2	MSR_curDynIPCommand	365
8.66.3	MSR_buildPoolFromClientDebsCMD	365
8.66.4	MSR_buildPoolFromClientDebs	365
8.66.5	MSR_statusBarInc	365
8.66.6	MSR_statusBarIncCommand	365
8.66.7	MSR_statusBarCommand	365

8.66.8	MSR_genericSendCommand . . . . .	366
8.66.9	MSR_statusBar . . . . .	366
8.66.10	MSR_curDynIP . . . . .	366
8.66.11	MSR_copyClientPackageStatus . . . . .	366
8.66.12	MSR_importDiffFile . . . . .	366
8.66.13	MSR_CommandCopyClientPackageStatus . . . . .	366
8.66.14	MSR_copyDiffFileFromClient . . . . .	366
8.66.15	MSR_markm23normalAsDone . . . . .	367
8.66.16	MSR_decodeClientSideBase64 . . . . .	367
8.66.17	MSR_clientSideBase64Encode . . . . .	367
8.66.18	MSR_genSendBinayFileCommand . . . . .	367
8.66.19	MSR_m23ImagerMBR . . . . .	367
8.66.20	MSR_m23ImagerSize . . . . .	367
8.66.21	MSR_getm23clientIDCMD . . . . .	367
8.66.22	MSR_importPackageStatus . . . . .	368
8.66.23	MSR_importLog . . . . .	368
8.66.24	MSR_logCommand . . . . .	368
8.66.25	MSR_statusFileCommand . . . . .	368
8.66.26	MSR_importStatusFile . . . . .	368
8.66.27	MSR_genSendCommand . . . . .	368
8.66.28	MSR_clientChangeCommand . . . . .	368
8.66.29	MSR_clientChange . . . . .	368
8.66.30	MSR_partHwDataCommand . . . . .	368
8.66.31	MSR_importPartHwData . . . . .	369
8.66.32	MSR_getClientSettingsCommand . . . . .	369
8.66.33	MSR_clientSettings . . . . .	369
8.66.34	MSR_setOnline . . . . .	369
8.66.35	MSR_addMonitor . . . . .	369
8.66.36	MSR_listMonitors . . . . .	369
8.67	/inc/packageBuilder.php . . . . .	370
8.67.1	PKGBUILDER_showDialog . . . . .	370
8.67.2	PKGBUILDER_showKeySelectionDialog . . . . .	370
8.67.3	PKGBUILDER_showUploadDialog . . . . .	370
8.67.4	PKGBUILDER_listFiles . . . . .	370
8.67.5	array_keysSearch . . . . .	370
8.67.6	PKGBUILDER_tar2deb . . . . .	370
8.68	/inc/packages.php . . . . .	371
8.68.1	PKG_addHSUser . . . . .	371
8.68.2	PKG_addUbuntuUser . . . . .	371
8.68.3	PKG_addDebianUser . . . . .	371
8.68.4	PKG_addUser . . . . .	371
8.68.5	PKG_cleanPackageLine . . . . .	372
8.68.6	PKG_combinem23normal . . . . .	372
8.68.7	PKG_importSelectedPackagesFromFile . . . . .	372
8.68.8	PKG_exportSelectedPackages . . . . .	372
8.68.9	PKG_getPackageStatusCSV . . . . .	372
8.68.10	PKG_getDebootstrapCacheFilename . . . . .	372
8.68.11	PKG_getDebootstrapCacheSfURL . . . . .	373
8.68.12	PKG_baseSysDownloadedCompletelyTom23Server . . . . .	373
8.68.13	PKG_downloadBaseSysTom23Server . . . . .	373
8.68.14	PKG_getDebootstrapCacheServerURL . . . . .	373
8.68.15	PKG_getDebootstrapCacheServerFile . . . . .	373
8.68.16	PKG_isReconfiguredWithExtraDistr . . . . .	374
8.68.17	PKG_translateClientjobsStatus . . . . .	374
8.68.18	PKG_isSpecialPackageAvailableForClient . . . . .	374
8.68.19	PKG_OptionPageHeader2 . . . . .	374



8.68.20	PKG_OptionPageTail2	374
8.68.21	PKG_decodeDebconfDescription	374
8.68.22	PKG_OptionPageRender2	375
8.68.23	PKG_countSpecialPackages	375
8.68.24	PKG_countWaitingJobs	375
8.68.25	PKG_closeSearch	375
8.68.26	PKG_getNextPackage	375
8.68.27	PKG_listRecommendPackages	375
8.68.28	PKG_listRecommendSubPackages	375
8.68.29	PKG_addRecommendPackages	376
8.68.30	PKG_addPackageSelection	376
8.68.31	PKG_addNormalPackagesToWait4Aac	376
8.68.32	PKG_addSpecialPackagesToWait4Aac	376
8.68.33	PKG_countJobsWithStatus	376
8.68.34	PKG_getClientjobsStatus	377
8.68.35	PKG_addNormalPackages	377
8.68.36	PKG_changePrioritySelectedPackages	377
8.68.37	PKG_changeInstallReasonSelectedPackages	377
8.68.38	PKG_rmSelectedPackages	377
8.68.39	PKG_listSelectedpackages	378
8.68.40	PKG_countSelectedpackages	378
8.68.41	PKG_countJobs	378
8.68.42	PKG_hasOptions	378
8.68.43	PKG_savePackageselection	378
8.68.44	PKG_addPackageToPackageselection	379
8.68.45	PKG_listSpecialpackages	379
8.68.46	PKG_addSpecialPackages	379
8.68.47	PKG_getSpecialPackagePriority	379
8.68.48	PKG_getSpecialPackageDescription	379
8.68.49	PKG_getSpecialPackageInfo	379
8.68.50	PKG_getPackageID	380
8.68.51	PKG_rmNormalJob	380
8.68.52	PKG_addJob	380
8.68.53	PKG_discardNormalJob	380
8.68.54	PKG_addWait4AccJob	380
8.68.55	PKG_addStatusJob	381
8.68.56	PKG_acceptJobs	381
8.68.57	PKG_discardJobs	381
8.68.58	PKG_discardJob	381
8.68.59	PKG_changeClientPackageAction	381
8.68.60	PKG_setClientPackageWait4Rm	381
8.68.61	PKG_setClientPackageInstalledOK	382
8.68.62	PKG_addShutdownPackage	382
8.68.63	PKG_addShutdownOrRebootPackage	382
8.68.64	PKG_getAllParams	382
8.68.65	PKG_setAllParams	382
8.68.66	PKG_OptionPageHeader	382
8.68.67	PKG_OptionPageRender	382
8.68.68	PKG_OptionPageTail	382
8.68.69	PKG_OptionPageSaveAlsParameters	383
8.68.70	PKG_OptionPageGetValue	383
8.68.71	PKG_listParams	383
8.68.72	PKG_getRecommendPackageAllInstalledSize	383
8.68.73	PKG_previewInstallationDeinstallation	383
8.68.74	PKG_showPreviewInstallationDeinstallation	383
8.68.75	PKG_updateSourcesListAtAllClients	383

8.68.76	PKG_executeOnClientJobs . . . . .	384
8.68.77	PKG_removeFromJobList . . . . .	384
8.68.78	PKG_changeClientJobsStatus . . . . .	384
8.68.79	PKG_removeSpecialFromJobList . . . . .	384
8.68.80	PKG_getHigestIDOfSpecialPackage . . . . .	384
8.68.81	PKG_previewUpdateSystem . . . . .	384
8.68.82	PKG_showPreviewUpdateSystem . . . . .	384
8.68.83	PKG_rmAllSpecialPackagesByName . . . . .	385
8.68.84	PKG_getClientsWithPackage . . . . .	385
8.68.85	PKG_getClientsWithWaitingJobs . . . . .	385
8.68.86	PKG_getClientsByPackages . . . . .	385
8.68.87	PKG_countPackages . . . . .	385
8.68.88	PKG_copyWait4accPackagesToClient . . . . .	385
8.68.89	PKG_copyPackagesToClient . . . . .	386
8.68.90	PKG_remNormalPackages . . . . .	386
8.68.91	PKG_addRemovePackagesToWait4Aac . . . . .	386
8.68.92	PKG_discardRemoveJob . . . . .	386
8.68.93	PKG_deletePackageSelection . . . . .	386
8.68.94	PKG_getAllPackageSelections . . . . .	386
8.68.95	PKG_multiPackageSelectionsSelection . . . . .	386
8.68.96	PKG_showAllPackageSelections . . . . .	387
8.68.97	PKG_getPackageParams . . . . .	387
8.68.98	PKG_getClientbyPackageID . . . . .	387
8.68.99	PKG_getInfoFromPackageID . . . . .	387
8.68.100	PKG_getClientIDbyPackageID . . . . .	387
8.68.101	PKG_getPackageParamsVar . . . . .	387
8.68.102	PKG_getPackageIDsByName . . . . .	387
8.68.103	PKG_getClientPackages . . . . .	388
8.68.104	PKG_getPackagesListMarker . . . . .	388
8.68.105	PKG_savePackagesList . . . . .	388
8.68.106	PKG_getDuplicatedIdenticalJobs . . . . .	388
8.68.107	PKG_removeDuplicatedIdenticalJobs . . . . .	388
8.68.108	PKG_loadPackagesList . . . . .	388
8.68.109	PKG_deletePackagesList . . . . .	388
8.68.110	PKG_addNormalJob . . . . .	389
8.68.111	PKG_recountAllClientPackages . . . . .	389
8.69	./inc/pdf.php . . . . .	390
8.69.1	PDF_showTableRow . . . . .	390
8.69.2	PDF_showTableHeader . . . . .	390
8.69.3	PDF_showTableEnd . . . . .	390
8.69.4	PDF_init . . . . .	390
8.69.5	PDF_output . . . . .	390
8.70	./inc/plugin.php . . . . .	391
8.70.1	PLG_listMenuPlugins . . . . .	391
8.70.2	PLG_isPluginSelected . . . . .	391
8.70.3	PLG_isPluginSelected . . . . .	391
8.70.4	PLG_getPLGName . . . . .	391
8.70.5	PLG_getPLGPage . . . . .	391
8.70.6	PLG_getPLGAuthor . . . . .	391
8.70.7	PLG_getPLGUpdateURL . . . . .	391
8.70.8	PLG_getPLGClientRequires . . . . .	392
8.70.9	PLG_getPLGVersion . . . . .	392
8.70.10	PLG_showDownloadStatus . . . . .	392
8.70.11	PLG_showDownloadStatus . . . . .	392
8.70.12	PLG_checkOverwriting . . . . .	392
8.70.13	PLG_DBInstall . . . . .	392

8.70.14	PLG_realInstall . . . . .	392
8.70.15	PLG_getTempDir . . . . .	393
8.70.16	PLG_getFilename . . . . .	393
8.70.17	PLG_install . . . . .	393
8.70.18	PLG_showPluginOverview . . . . .	393
8.70.19	PLG_install . . . . .	393
8.70.20	PLG_getUpdateFile . . . . .	393
8.70.21	PLG_update . . . . .	393
8.70.22	PLG_listInfofile . . . . .	393
8.70.23	PLG_realUpdate . . . . .	394
8.71	./inc/pool.php . . . . .	395
8.71.1	POOL_selectPoolType . . . . .	395
8.71.2	POOL_getPools . . . . .	395
8.71.3	POOL_showLoadDeleteCreate . . . . .	395
8.71.4	POOL_create . . . . .	395
8.71.5	POOL_setProperty . . . . .	395
8.71.6	POOL_getProperty . . . . .	395
8.71.7	POOL_delete . . . . .	395
8.71.8	POOL_showReadCD . . . . .	396
8.71.9	POOL_readCD . . . . .	396
8.71.10	POOL_createExtendedPackageIndex . . . . .	396
8.71.11	POOL_showCreatePackageIndex . . . . .	396
8.71.12	POOL_getSize . . . . .	396
8.71.13	POOL_getDir . . . . .	396
8.71.14	POOL_download . . . . .	396
8.71.15	POOL_showDownloadStatus . . . . .	397
8.71.16	POOL_prepare . . . . .	397
8.71.17	POOL_makeRepository . . . . .	397
8.71.18	POOL_getCDDistributionRelease . . . . .	397
8.71.19	POOL_showSourcesList . . . . .	397
8.72	./inc/preferences.php . . . . .	398
8.72.1	PREF_preferencLoadManagerHandler . . . . .	398
8.72.2	PREF_preferencSaveManagerHandler . . . . .	398
8.72.3	PREF_showPreferenceManager . . . . .	398
8.72.4	PREF_saveAllPreferenceValues . . . . .	398
8.72.5	PREF_loadAllPreferenceValues . . . . .	398
8.72.6	PREF_getClientPreferences . . . . .	398
8.72.7	PREF_getValue . . . . .	398
8.72.8	PREF_putValue . . . . .	399
8.72.9	PREF_delete . . . . .	399
8.72.10	PREF_exists . . . . .	399
8.72.11	PREF_putAllOptions . . . . .	399
8.72.12	PREF_getAllValues . . . . .	399
8.73	./inc/remotovar.php . . . . .	400
8.73.1	RMV_exists4IP . . . . .	400
8.73.2	RMV_set4IP . . . . .	400
8.73.3	RMV_get4IP . . . . .	400
8.73.4	RMV_set . . . . .	400
8.73.5	RMV_get . . . . .	400
8.73.6	RMV_rm4IP . . . . .	400
8.73.7	RMV_rm . . . . .	401
8.73.8	RMV_rm_old . . . . .	401
8.74	./inc/scredit.php . . . . .	402
8.74.1	SCREDIT_correctScriptFilename . . . . .	402
8.74.2	SCREDIT_showEditor . . . . .	402
8.74.3	SCREDIT_newScriptTemplate . . . . .	402

8.75	/inc/serverBackup.php . . . . .	403
8.75.1	SERVERBACKUP_getBackupConfiguration . . . . .	403
8.75.2	SERVERBACKUP_storeBackupConfiguration . . . . .	403
8.75.3	SERVERBACKUP_showConfigurationDialog . . . . .	403
8.75.4	SERVERBACKUP_runBackupNowDialog . . . . .	403
8.75.5	SERVERBACKUP_getBackupList . . . . .	403
8.75.6	SERVERBACKUP_rmBackup . . . . .	403
8.75.7	SERVERBACKUP_backupOverviewDialog . . . . .	403
8.76	/inc/server.php . . . . .	404
8.76.1	SERVER_importGPGPackageSignKey . . . . .	404
8.76.2	SERVER_logLocalScreenSessionToFile . . . . .	404
8.76.3	SERVER_setSSLCertCheckDisabled . . . . .	404
8.76.4	SERVER_isSSLCertCheckDisabled . . . . .	404
8.76.5	SERVER_isLiveLogDisabled . . . . .	404
8.76.6	SERVER_setLiveLogDisabled . . . . .	404
8.76.7	SERVER_isInstallReasonEnabled . . . . .	404
8.76.8	SERVER_setInstallReasonEnabled . . . . .	405
8.76.9	SERVER_isUpdatePackageInfosDisabled . . . . .	405
8.76.10	SERVER_setUpdatePackageInfosDisabled . . . . .	405
8.76.11	SERVER_isClientOnlineStatusEnabled . . . . .	405
8.76.12	SERVER_startUpdateClientOnlineInDBBackgroundJob . . . . .	405
8.76.13	SERVER_setClientOnlineStatusEnabled . . . . .	405
8.76.14	SERVER_setServerBoolSetting . . . . .	405
8.76.15	SERVER_getServerBoolSetting . . . . .	405
8.76.16	SERVER_setServerSetting . . . . .	406
8.76.17	SERVER_getServerSetting . . . . .	406
8.76.18	SERVER_existsServerSetting . . . . .	406
8.76.19	SERVER_killPID . . . . .	406
8.76.20	SERVER_killBackgroundJob . . . . .	406
8.76.21	SERVER_insertLineNumber . . . . .	406
8.76.22	SERVER_addAdmin . . . . .	407
8.76.23	SERVER_delAdmin . . . . .	407
8.76.24	SERVER_fileExists . . . . .	407
8.76.25	SERVER_getPublicSSHKeyOfm23Server . . . . .	407
8.76.26	SERVER_changeHtpasswd . . . . .	407
8.76.27	SERVER_delFromHtpasswd . . . . .	408
8.76.28	SERVER_addToHtpasswd . . . . .	408
8.76.29	SERVER_dhcpServerInNetWarn . . . . .	408
8.76.30	SERVER_sendScriptToSF . . . . .	408
8.76.31	SERVER_dynamicIPWarn . . . . .	408
8.76.32	SERVER_tmpNotWritable . . . . .	408
8.76.33	SERVER_rootFreeSpace . . . . .	409
8.76.34	SERVER_isProgramRunning . . . . .	409
8.76.35	SERVER_checkPackageInstalled . . . . .	409
8.76.36	SERVER_daemonStartStop . . . . .	409
8.76.37	SERVER_getAptGetInstallCommand . . . . .	409
8.76.38	SERVER_installTool . . . . .	409
8.76.39	SERVER_installToolInBackground . . . . .	409
8.76.40	SERVER_programmStatus . . . . .	410
8.76.41	SERVER_apacheInfo . . . . .	410
8.76.42	SERVER_mysqlInfo . . . . .	410
8.76.43	SERVER_dhcpInfo . . . . .	410
8.76.44	SERVER_LDAPInfo . . . . .	410
8.76.45	SERVER_programmStatusTableHeader . . . . .	410
8.76.46	SERVER_runInBackground . . . . .	410
8.76.47	SERVER_runningInBackground . . . . .	410

8.76.48	SERVER_runningInScreen . . . . .	411
8.76.49	SERVER_addLineToFile . . . . .	411
8.76.50	SERVER_deleteFile . . . . .	411
8.76.51	SERVER_getFileContents . . . . .	411
8.76.52	SERVER_putFileContents . . . . .	411
8.76.53	SERVER_delLineFromFile . . . . .	412
8.76.54	SERVER_addEtcHosts . . . . .	412
8.76.55	SERVER_delEtcHosts . . . . .	412
8.76.56	SERVER_getInstallationMedium . . . . .	412
8.76.57	SERVER_getOS . . . . .	412
8.76.58	SERVER_checkDownload . . . . .	412
8.76.59	SERVER_checkDiskFree . . . . .	412
8.76.60	SERVER_checkRunInVM . . . . .	413
8.76.61	SERVER_checkKernel . . . . .	413
8.76.62	SERVER_multiMkDir . . . . .	413
8.76.63	SERVER_commandAvailable . . . . .	413
8.77	./inc/sourceslist.php . . . . .	414
8.77.1	SRCLST_getAddToFile . . . . .	414
8.77.2	SRCLST_getRelease . . . . .	414
8.77.3	SRCLST_genList . . . . .	414
8.77.4	SRCLST_saveArchitectures . . . . .	414
8.77.5	SRCLST_saveList . . . . .	414
8.77.6	SRCLST_querySourceslists . . . . .	415
8.77.7	SRCLST_genSelection . . . . .	415
8.77.8	SRCLST_getValue . . . . .	415
8.77.9	SRCLST_loadSourceListFromDB . . . . .	415
8.77.10	SRCLST_sourceListExists . . . . .	415
8.77.11	SRCLST_loadSourceList . . . . .	415
8.77.12	SRCLST_getDescription . . . . .	415
8.77.13	SRCLST_delete . . . . .	416
8.77.14	SRCLST_checkList . . . . .	416
8.77.15	SRCLST_packageInformationChangeInformationHumanReadable . . . . .	416
8.77.16	SRCLST_packageInformationChangeTime . . . . .	416
8.77.17	SRCLST_packageInformationOlderThan . . . . .	416
8.77.18	SRCLST_getStorageFS . . . . .	417
8.77.19	SRCLST_supportedFS . . . . .	417
8.77.20	SRCLST_alternativeFS . . . . .	417
8.77.21	SRCLST_getParameter . . . . .	417
8.77.22	SRCLST_getMirror . . . . .	417
8.77.23	SRCLST_getDesktopList . . . . .	418
8.77.24	SRCLST_showDesktopsSel . . . . .	418
8.77.25	SRCLST_doesDistrSupportEFI . . . . .	418
8.77.26	SRCLST_getListnamesWithEfiSupport . . . . .	418
8.77.27	SRCLST_clientUsesEfiButSourcesListDoesntSupportEfi . . . . .	418
8.77.28	SRCLST_showErrorIfClientUsesEfiButSourcesListDoesntSupportEfi . . . . .	418
8.77.29	SRCLST_showAlternativeArchitectureSelection . . . . .	419
8.77.30	SRCLST_isArchAvailable . . . . .	419
8.77.31	SRCLST_getArchitectures . . . . .	419
8.77.32	SRCLST_showEditor . . . . .	419
8.77.33	SRCLST_getListnames . . . . .	419
8.77.34	SRCLST_cleanList . . . . .	419
8.77.35	SRCLST_matchList . . . . .	420
8.77.36	SRCLST_possiblem23debsMirrors . . . . .	420
8.77.37	SRCLST_checkm23debsMirror . . . . .	420
8.77.38	SRCLST_getWorkingm23debsMirror . . . . .	420
8.78	./inc/ucs.php . . . . .	421

8.78.1	UCS_getPrefnameByClient	421
8.78.2	UCS_addUCSClientTom23ClientPreferences	421
8.78.3	UCS_delUCSClientFrom23ClientPreferences	421
8.78.4	UCS_createNetworkObject	421
8.78.5	UCS_getGenericNetworkName	421
8.78.6	UCS_ensureNetworkObjectExists	422
8.78.7	UCS_modifyClientIP	422
8.78.8	UCS_getAllClientNamesLDAP	422
8.78.9	UCS_getUDMInfo	422
8.78.10	UCS_networkObjectExists	422
8.78.11	UCS_getNetworkLDAPInfo	422
8.78.12	UCS_getFirstElementFromDN	423
8.78.13	UCS_getClientLDAPInfo	423
8.78.14	UCS_addClient	423
8.78.15	UCS_delClient	423
8.78.16	UCS_enableClientPXEBoot	423
8.78.17	UCS_disableClientPXEBoot	423
8.78.18	UCS_setClientDistrAndRelease	423
8.78.19	UCS_openFirewallPort	424
8.78.20	UCS_addLDAPUser	424
8.78.21	UCS_getEtc_ucr_master	424
8.78.22	UCS_enableClientLDAP	424
8.78.23	UCS_udmSuccessOrErrorMessage	424
8.78.24	UCS_getUsedIPs	424
8.79	./inc/update.php	425
8.79.1	UPDATE_doUpdate	425
8.79.2	UPDATE_running	425
8.79.3	UPDATE_getUrl	425
8.79.4	UPDATE_getInfo	425
8.80	./inc/vm.php	426
8.80.1	VM_isHost	426
8.80.2	VM_captureVMScreenAsMovie	426
8.80.3	VM_CloudStackDeleteClientVM	426
8.80.4	VM_isCloudStackClient	426
8.80.5	VM_CloudStackCheckConstants	427
8.80.6	VM_CloudStackConfigGUI	427
8.80.7	VM_CloudStackWriteConfFile	427
8.80.8	VM_CloudStackUploadIso	427
8.80.9	VM_CloudStackEnablePortForwarding	428
8.80.10	VM_CloudStackDisablePortForwarding	428
8.80.11	VM_CloudStackSendSetVisualURL	428
8.80.12	VM_CloudStackStartVM	428
8.80.13	VM_CloudStackStopVM	428
8.80.14	VM_CloudStackGetVMStatus	429
8.80.15	VM_CloudStackClientName2ClientID	429
8.80.16	VM_CloudStackNetBootActivate	429
8.80.17	VM_CloudStackCreateVM	429
8.80.18	VM_CloudStack_getServerIP	429
8.80.19	VM_GUIstepCreateCloudStackVM	429
8.80.20	VM_CloudStack_available	430
8.80.21	VM_CloudStack_getObject	430
8.80.22	VM_CloudStack_getVersion	430
8.80.23	VM_shutdownAndDisableNetbootAfterInstall	430
8.80.24	VM_rebootAndActivateNetboot	430
8.80.25	VM_shutdownAndDisableNetbootAfterInstall	430
8.80.26	VM_shutdownAndDisableNetboot	430

8.80.27 VM_rebootChangeBootDevice . . . . .	431
8.80.28 VM_getVBoxVersion . . . . .	431
8.80.29 VM_setVBoxAddonAsDefault . . . . .	431
8.80.30 VM_downloadedVBoxAddons . . . . .	431
8.80.31 VM_getVBoxAddonDefaultVersion . . . . .	431
8.80.32 VM_generateVBOXaddonDownloadCMD . . . . .	431
8.80.33 VM_downloadVBOXaddons . . . . .	431
8.80.34 VM_VBOXaddonDownloadDialog . . . . .	432
8.80.35 VM_wasVBoxAddonDownloaded . . . . .	432
8.80.36 VM_listDownloadableVBoxAddons . . . . .	432
8.80.37 VM_stopVM . . . . .	432
8.80.38 VM_pauseVM . . . . .	432
8.80.39 VM_resumeVM . . . . .	432
8.80.40 VM_webAction . . . . .	433
8.80.41 VM_delete . . . . .	433
8.80.42 VM_vmSwNr2Name . . . . .	433
8.80.43 VM_getHTMLStatusBlock . . . . .	433
8.80.44 VM_activateNetboot . . . . .	433
8.80.45 VM_convertSwitchStatusInfo . . . . .	434
8.80.46 VM_getSWandHost . . . . .	434
8.80.47 VM_getStatus . . . . .	434
8.80.48 VM_GUIstepCreateGuest . . . . .	434
8.80.49 VM_GUIstepCheckHost . . . . .	434
8.80.50 VM_GUIstepSelectHost . . . . .	434
8.80.51 VM_getAllVMHosts . . . . .	435
8.80.52 VM_setVisualURL . . . . .	435
8.80.53 VM_setHostInDB . . . . .	435
8.80.54 VM_setGuestInDB . . . . .	435
8.80.55 VM_statusIcons . . . . .	435
8.80.56 VM_createDiskImage . . . . .	436
8.80.57 VM_delVMCMD . . . . .	436
8.80.58 VM_activateNetbootCMD . . . . .	436
8.80.59 VM_createVM . . . . .	436
8.80.60 VM_insertBootISO . . . . .	437
8.80.61 VM_startVMInExistingXSession . . . . .	437
8.80.62 VM_startVM . . . . .	437
8.80.63 VM_startVMCommandFile . . . . .	437
8.80.64 VM_stopVMCommandFile . . . . .	437
8.80.65 VM_status . . . . .	438
8.80.66 VM_parseVBOXdisk . . . . .	438
8.80.67 VM_parseVBOXstate . . . . .	438
8.80.68 VM_parseVBOXNic . . . . .	438
8.80.69 VM_parseStatus . . . . .	438





# 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 expect 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 Sub-section b above.)

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

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

- 4. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.
- 5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.
- 6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.
- 7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

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

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

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

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

8. If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.
9. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.  
Each version is given a distinguishing version number. If the Program specifies a version number of this License which applies to it and “any later version”, you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.
10. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

## NO WARRANTY

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM “AS IS” WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.
12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

## END OF TERMS AND CONDITIONS

## Appendix: How to Apply These Terms to Your New Programs

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

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

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

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

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

You should have received a copy of the GNU General Public License along with this program; if not, write to the Free Software Foundation, Inc., 59 Temple Place - Suite 330, Boston, MA 02111-1307, USA.

Also add information on how to contact you by electronic and paper mail.

If the program is interactive, make it output a short notice like this when it starts in an interactive mode:

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

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

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

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

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

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





## Chapter 3

# MDK menu system

### 3.1 Introduction

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

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

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

### 3.2 Starting the MDK menu system

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

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

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



# Chapter 4

## database

### 4.1 overview

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

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

### 4.2 the tables

#### 4.2.1 clientjobs

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

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

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

### 4.2.2 clientlogs

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

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

### 4.2.3 clientpackages

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

### 4.2.4 clientpreferences

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

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

### 4.2.5 clients

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

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

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

#### 4.2.6 ftpusers

This table can store information about ftpusers with their permissions.

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

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

#### 4.2.7 groups

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

- groupname: name for the group

#### 4.2.8 plugins

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

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

#### 4.2.9 recommendpackages

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

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

#### 4.2.10 remotevar

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

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

# Chapter 5

## HowTos

### 5.1 Using new Debian/Ubuntu releases with m23

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

#### 5.1.1 Test and development

- Build a compressed root file system for the new release via `/mdk/m23helper/compressedDebootstrap` and put it (for testing) into the directory `/m23/data+scripts/packages/baseSys` on the m23 server.
- Sign the baseSys files by running `/mdk/m23helper/signBaseSys.sh` in the directory containing the compresses base system archives.
- Create an empty file (touch) with the release name of the new distribution release under `/m23/data+scripts/distr/debian/d`
- Add the new release to the list in `DISTRreleaseVersionTranslator`.
- Search all occurrences with the release name of the old version in the PHP files to get hints for finding places where new hacks or adaption need to be made. Search for "DebianVersionSpecific" too.
- Adjust `/mdk/m23Debs/m23-initscripts/postinst` when needed.
- Check, if all desired desktop environments are present in the file `/m23/inc/distr/<distribution>/info.txt` and add missing desktops.
- Write a new package source list (based on a previous release) in the m23 webinterface and choose the release with the name of the previously touched file name. Hook all desktops that should be deployed with this release.
- Make a base client install, see errors and fix them ;-)
- Run `"debootstrap --print-debs <distribution> /tmp/debootstrap.tmp"` on the installed client and place the found packages on the m23 server under `/m23/inc/distr/baseSysFileLists/<distribution>`.
- Copy `/mdk/m23helper/getAllDebianDesktopPackages` to the client and run it there.
- Develop missing desktop installation scripts under `/m23/inc/distr/<distribution>/packages/m23<desktop>Install.php`.
- Then make an installation/test/fix run with all desktop environments that should be supported by the release.

### 5.1.2 Building the packages

- Generate the package template files with `/mdk/m23helper/getDebianTemplates` and run `/mdk/m23helper/template2confpage.sh` in the directory containing the template files. Afterwards move the `*OptionPage.php` files from the subdirectory "out" to `/m23/data+scripts/m23admin/packages/<distribution>/<release>/.`
- Add the sources list name in `/mdk/bin/exportDBsourceslist.php`.
- Build new packages via the MDK.

## 5.2 Releasing a new m23 version

### 5.2.1 Switch to relase

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

### 5.2.2 Documentation

- Translate all new and changed texts (`/m23/inc/help/*` and `/m23/inc/i18n/*`)
- Fetch and merge the files via `/mdk/m23helper/gitCloneAndMeldTranslations` .
- Check for missing I18N variables. Run `/mdk/doc/manual/bin/checkForMissingi18n.sh`
- Manual
  - Check, if all parts seem to be there. Run: `/mdk/doc/manual/bin/checkForMissingHlp.sh`
  - Check, if all HTML entities are in the HTML to LaTeX translation index. Run: `/mdk/doc/manual/bin/checkForMissingHT`
  - Check, if there are all screenshots present. Run: `/mdk/doc/manual/bin/checkMissingScreenshots.sh`
  - Generate and update the screenshots
  - Run `/mdk/bin/menuStart`  $\Rightarrow$  *doc*  $\Rightarrow$  *manualChoosethelanguage*  $\Rightarrow$  *lang.Chooseall*
- Generate the tex files:  $\Rightarrow$  *textOptimisethePNGs*.  $\Rightarrow$  *optimisePNGs*
- Generate PDF and HTML version of the manual:  $\Rightarrow$  *pdf* – *htmlUpload*  $\Rightarrow$  *upload*
- Go back
  - Development guide
    - Add (maybe existing new) LaTeX files
    - Build the development guide:  $\Rightarrow$  *devguideGenerateit*  $\Rightarrow$  *generate*
    - Upload:  $\Rightarrow$  *upload*

### 5.2.3 Building the files

- Debian packages
  - `/mdk/bin/menuStart`  $\Rightarrow$  *debs*  $\Rightarrow$  *buildMayberecreatetheUCSextrapackage* : *CallquickBuild.shm23-ucs - extra*
- Test the packages
  - \* UCS VM (local repository): Call `/mdk/ucs/installLocalm23DebsOnUCS` to install m23 and create the tar with the unmaintained Debian packages.
  - \* UCS VM (online test repository): Copy `/mdk/ucs/addUCSTestRepository` to the UCS and install m23
  - \* Debian: Call `/mdk/m23helper/addLocalm23ServerRepository`, install m23
- Upload the packages, if all seems to be right:  $\Rightarrow$  *directuplinstLoginintot5*, call `/var/www/ucs/upload-4.1`



- Server installation ISO
  - `/mdk/bin/menuStart ⇒ serverISO ⇒ iso.Exitafterbuilding.Upload : cd/mdk/server; ./upiso.file`
- VirtualBox appliance
  - Create a new VM (Linux/Debian, 512 MB Ram, growing VDI 16GB HDD)
  - Change the Network card to Network bridge
  - Start the VM and "insert" the ISO into the virtual CD drive
  - Select "English".
  - Use "m23s" as server name, "test" as password and local network settings.
  - Choose the automatic partitioning and formatting.
  - Call `/mdk/bin/prepareOSForCompression` after reboot.
  - Shutdown the VM
  - Export it (`m23server_xx.y_rock.ova`) and compress (`7zm / 7zr -t7z -m0=lzma -mx=9 -mfb=64 -md=32m -ms=on a m23server_xx.y_rock.7z m23server_xx.y_rock.ova`) the exported files.
- - Write a SD card with Raspbian.
  - Configure `/etc/network/interfaces` to static IP for `eth0`.
  - Boot the Raspberry Pi.
  - Log into it via SSH (user: `pi`, password: `raspberry`)
  - Run `"sudo raspi-config"`.
  - Call in the menu "Expand Filesystem", set "Advanced Options" "Memory split" to 16 MB, call "Update".
  - Reboot.
  - Enable root access (set password to test): `"sudo passwd"`
  - Enable root login for SSH in `/etc/ssh/sshd_config`
  - Log out and log into it via SSH (user: `root`, password: `test`)
  - Use commands from `/mdk/m23helper/copyPasteTestm23ServerInstall`
  - `/mdk/bin/prepareOSForCompression`.
  - Compress the SD card: `dd if=/dev/sdX | 7zr a -t7z -m0=lzma -mx=9 -mfb=64 -md=32m -ms=on -si m23server_xx.y_rock-RaPi.7z`

#### 5.2.4 Update the m23admin-offline-copy

- VM: Start the VM and make sure it's available under 192.168.1.23
- Host: Export and transfer a new SQL dump: `/mdk/m23helper/m23admin-offline-copy/export-dataset.sh`
- VM: Import the SQL dump and data files: `/mdk/m23helper/m23admin-offline-copy/import.sh`
- Host: Call `/mdk/m23helper/m23admin-offline-copy/dlhttrack` to make the offline copy and upload the files to SF.

#### 5.2.5 CMS

- Write an article
- Create the link files by calling `/mdk/doc/CMSLinkCreator.sh` for the ISO, OVA and RasPi image.
- Check in into Bazaar
- Upload the files.

### 5.2.6 Announcement

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

## 5.3 How to translate m23?

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

### 5.3.1 make directories

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

Make the directories:

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

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

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

### 5.3.2 generate the language file

To give your language a name and make it available to m23 create a *language.info* file.

The language.info has to contain the following lines:

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

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

Your language.info may look like this:

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

### 5.3.3 translating the messages

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

Now translate the text between the ' ' 's. e.g.

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

becomes

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

Please don't delete any other text and make changes only between the ' ' ' letters. Don't translate something like \$I18N\_help to \$I18N\_hilfe. If you do so, m23 can't find it and will leave the place, the text should be appear empty. For your information: m23 stores the texts as variables that are inserted at the right places in the m23admin interface.

### 5.3.4 translating the help texts

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

## 5.4 How to generate a new base system?

### 5.4.1 install necessary system files

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

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

### 5.4.2 editing files

#### **fstab**

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

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

#### **lilo.conf**

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

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

default=m23Server3

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

**sources.list**

Your sources.list should look like this:

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

You can add more sources if preferred.

**/etc/network**

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

**5.4.3 step in to your new system**

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

```
chroot .
```

**5.4.4 source update and cleanup**

I think we don't need mailx :).

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

**5.4.5 installing a kernel**

You can list all available kernels with the following command:

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

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

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

**in case of an error**

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

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

**5.4.6 system update and installation of additional packages**

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

**5.4.7 booting your new system**

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

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

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

## 5.5 How to create a bootable server installation CD?

### 5.5.1 Introduction

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

### 5.5.2 what does the `makeCD` script?

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

#### **initrd**

While booting from the CD the kernel is started, if the kernel starts up it loads an initial file system from a file, this is the `initrd` (here called `rescue`). This file is compressed and will be extracted to the memory and mounted as root file system. All needed files like basic commands (`ls,cp`) and the install script are contained in this root file system.

In the `/mdk/bootCD/root` are all files needed for the `initrd`. If you want to make changes to the start up process of the boot CD you should edit the `/mdk/bootCD/root` directory. The `mkRescue.sh` script makes an empty `initrd` file, formats it with `ext2` file system, copies needed files to `initrd` and compresses it to `rescue.gz`. For further information have a look at the `mkRescue.sh` script.

#### **compress operating system**

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

#### **compresses the `m23` files**

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

#### **generating the ISO image**

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

## 5.6 How to make network bootimages?

### 5.6.1 Introduction

What are bootimages? Bootimages are used to boot up a client over the network. The bootimage contains a Linux kernel, an initial ramdisk and a script for fetching the jobs for the client.

A client runs for first start up thru the following sequence:

- get IP, netmask, bootimage name, ... from the `m23` DHCP server

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

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

### 5.6.2 creating a new bootimage

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

#### How it works?

it creates a blank file of 17MB size

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

sets up a loop device

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

the 23MB file will be formatted with EXT2

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

mount the file

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

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

```
> cp -rdpR root2/* mnt/
```

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

ummount the image

```
> umount mnt
```

set "down" the loop device

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

set correct file permissions

```
> chmod 0555 bzImage
```

```
> chown root:root bzImage
```

set boot device in kernel

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

generate bootimage for Etherboot

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

generate files for PXE

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

### 5.6.3 What can you do with this SDK?

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

#### Build a new kernel

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

## 5.7 How to generate an Etherboot bootimage for booting from hard-disk?

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

### 5.7.1 exchange epic100.c with the special boot code

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

Here you can see how its done:

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

#define LINUX_OUT_MACROS

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

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

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

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

/*
```

```

#define DEBUG_RX
#define DEBUG_TX
#define DEBUG_EEPROM
*/

#define EPIC_DEBUG 0      /* debug level */

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

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

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

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

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

static int     ioaddr;

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

```



```

static int      eththr ;

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

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

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

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

static void
epic100_open(void)
{
}

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

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

```

```

        unsigned int len, const char *data)
{
}

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

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

static void
epic100_disable(struct nic *nic)
{
}

#ifdef DEBUG_EEPROM
/* Serial EEPROM section. */

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

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

#define eeprom_delay(n) delay(n)

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

```

```

#define MII_READOP      1
#define MII_WRITEOP     2

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

```

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

```

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

```

with

```
#define ANS_DEFAULT ANS_LOCAL
```

and

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

with

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

Now execute the following commands:

```

make clean
make

```

### 5.7.2 Making the bootimage

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

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

## 5.8 How to create the server update files?

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

### 5.8.1 The files

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

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

**xxx.begin**

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

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

**xxx.end**

```
rm newdata.tb2
```

**xxx.info**

```
shiver
```

**xxx.text**

You can use html tags in your release information.

This is the new shiver update. There will be the following changes:

```
<ul>
  <li> bigger</li>
  <li> louder</li>
  <li> ...
</ul>
```

## 5.9 debugging m23 scripts

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

### 5.9.1 Activate PHP debugging

You can activate the PHP warning messages if you call:

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

to deactivate the debugging mode simply call:

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

### 5.9.2 Turn on m23 script debugging

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

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

### 5.9.3 Turn off m23 script debugging

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

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

### 5.9.4 Set script status

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

- call `http://serverIP/m23admin/phpMyAdmin/` with your webbrowser
- select the database "m23"
- select the table "clientjobs"
- click on "Browse"
- search for your job script
- click on "Edit"
- enter "waiting" at the "status" field
- click on "Go" to save your changes

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

### 5.9.5 get the output of your script

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

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

## 5.10 Additional installation parameters for normal packages

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

### 5.10.1 The \*OptionPage.php

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

### 5.10.2 Filling the \*OptionPage.php

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

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

Your OptionPage needs to have the following elements:

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

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

$layout[0]...

PKG_OptionPageTail($layout);
```

### 5.10.3 Layoutoptions

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

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

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

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

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

- *inputline*: An inputline is a editable text field with the height of one. The inputline has a lot of attributes:

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

```

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

```

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

```

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

```

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

```

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

```





## Chapter 6

# m23 helpers

### 6.1 mdoc

#### 6.1.1 what is mdoc?

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

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

#### 6.1.2 how to make your source code mdoc compatible?

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

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

#### 6.1.3 mdoc info block

In the mdoc info block you can leave all information you want, e.g. you can write down your name, the function of the file etc. . This block is parsed by mdoc first and will appear in the documentation at the beginning of the chapter. A mdoc info block begins with `"/*$mdocInfo"` and ends with `"$*/"`. All between these lines will be treated as a comment and is copied 1 to 1 to the documentation file. If you make a line break this line break will appear in the documentation too.

**Attention: For the 'I' in mdocInfo you have to use an upcase letter. Otherwise the mdoc info block will be ignored.**

Here an example for a mdoc info block:

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

#### 6.1.4 example for a mdoc comment

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

#### 6.1.5 using mdoc

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

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

#### 6.1.6 example

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

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

# Chapter 7

## m23customPatch

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

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

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

### 7.1 Indicating patchable areas in the source code

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

#### 7.1.1 Start position of a patchable area

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

#### 7.1.2 End position of a patchable area

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

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



## 7.2 m23customPatch file format

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

### 7.2.1 Example (logo.php.m23customPatch)



## 7.3 /m23/bin/m23customPatch

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

### 7.3.1 Return/error codes

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

## 7.4 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 8

# m23 API reference

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

## 8.1 `./data+scripts/packages/m23CommonInstallRoutines.php`

Author: Hauke Goos-Habermann (HHabermann@pc-kiel.de)

Description: Installatio routines shared among distributions.

### 8.1.1 `CIR_setDateAndTimeTemorarily`

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

### 8.1.2 `CIR_WorkaroundForMissingModulesDep`

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

### 8.1.3 `CIR_transferClientIP`

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

### 8.1.4 `CIR_writeClientID`

**Description:** writes the client ID to /m23clientID

**Parameter:**

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

### 8.1.5 `CIR_detectSCSI`

**Description:** detects SCSI controlers and loads the modules

### 8.1.6 `CIR_waitForNextJob`

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

### 8.1.7 `CIR_enableDropbear`

**Description:** sets up and starts dropbear SSH server

## 8.2 *./inc/assimilate.php*

Author: Hauke Goos-Habermann (HHabermann@pc-kiel.de)

Description: Contains functions for assimilation of clients

### 8.2.1 *ASSI\_showClientAddDialog*

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

### 8.2.2 *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

### 8.2.3 *ASSI\_addUbuntuRoot*

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

### 8.2.4 *ASSI\_prepareClient*

**Description:** Prepares a client for assimilation.

## 8.3 `./inc/autoTest.php`

Author: Hauke Goos-Habermann (HHabermann@pc-kiel.de)

Description: Functions for test automation.

### 8.3.1 `AUTOTEST_getKey`

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

**Parameter:**

- `charIn`: Input character.

**Returns:**

- Needed scan codes to produce a given character.

### 8.3.2 `AUTOTEST_calcScancodes`

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

**Parameter:**

- `in`: Input string with normal and special keys.

**Returns:**

- Scancodes that represent the input string.

### 8.3.3 `AUTOTEST_keyAndRelease`

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

**Parameter:**

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

**Returns:**

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

### 8.3.4 `AUTOTEST_VM_create`

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

**Parameter:**

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

**Returns:**

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



### 8.3.5 AUTOTEST\_VM\_enableCapture

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

**Parameter:**

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

### 8.3.6 AUTOTEST\_VM\_delete

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

**Parameter:**

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

### 8.3.7 AUTOTEST\_VM\_start

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

**Parameter:**

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

### 8.3.8 AUTOTEST\_VM\_insertBootISO

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

**Parameter:**

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

### 8.3.9 AUTOTEST\_VM\_rebootFromHD

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

**Parameter:**

- vmName: Name of the VM.

### 8.3.10 AUTOTEST\_executePHPFunction

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

**Parameter:**

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

### 8.3.11 AUTOTEST\_VM\_keyboardWrite

**Description:** Emulates the keystrokes into a VM.

**Parameter:**

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

### 8.3.12 AUTOTEST\_VM\_ocrScreen

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

**Parameter:**

- vmName: Name of the VM.

**Returns:**

- The recognised text of the display.

### 8.3.13 AUTOTEST\_VM\_getStatus

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

**Parameter:**

- vmName: Name of the VM.

**Returns:**

- Array with the current state of the VM.

### 8.3.14 AUTOTEST\_VM\_isRunning

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

**Parameter:**

- vmName: Name of the VM.

**Returns:**

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

## 8.4 *./inc/backup.php*

Author: Hauke Goos-Habermann (HHabermann@pc-kiel.de)

Description: functions for controlling BackupPC

### 8.4.1 **BACKUP\_showClientSettings**

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

**Parameter:**

- client: name of the client

### 8.4.2 **BACKUP\_getBackupDirs**

**Description:** Returns the comma separated list of directories that should be backed up on the client

**Parameter:**

- client: name of the client

### 8.4.3 **BACKUP\_saveBackupDirs**

**Description:** Saves the list of backup directories on the client in the BackupPC file

**Parameter:**

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

### 8.4.4 **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.

### 8.4.5 **BACKUP\_addAdmin**

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

**Parameter:**

- admin: Name of the admin.

### 8.4.6 **BACKUP\_delAdmin**

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

**Parameter:**

- admin: Name of the admin.

## 8.5 `./inc/bittorrent.php`

Author: Hauke Goos-Habermann (HHabermann@pc-kiel.de)

Description: Functions for distributing files via BitTorrent.

### 8.5.1 `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.

### 8.5.2 `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`.

### 8.5.3 `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.

### 8.5.4 `BT_stopService`

**Description:** Stops a Bittorrent service.

**Parameter:**

- `job`: Name of the job.

**Returns:**

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

### 8.5.5 `BT_startService`

**Description:** Starts a Bittorrent service.

**Parameter:**

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

**Returns:**

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

### 8.5.6 **BT\_stopTracker**

**Description:** Stops the Bittorrent tracker.

### 8.5.7 **BT\_startTracker**

**Description:** Starts the tracker.

### 8.5.8 **BT\_restartTracker**

**Description:** Restarts the tracker.

### 8.5.9 **BT\_autostart**

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

### 8.5.10 **BT\_updateWhitelist**

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

### 8.5.11 **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.

### 8.5.12 **BT\_startClient**

**Description:** Starts the Bittorrent client.

### 8.5.13 **BT\_stopClient**

**Description:** Stops the Bittorrent client.

### 8.5.14 **BT\_restartClient**

**Description:** Restarts the client.

## 8.6 `./inc/burn.php`

Author: Hauke Goos-Habermann (HHabermann@pc-kiel.de)

Description: functions for burning CDs

### 8.6.1 `BURN_listBurners`

**Description:** returns a selection of the available burners

**Parameter:**

- first: entry that should be shown first

### 8.6.2 `BURN_getDevice`

**Description:** returns the device name for cdrecord from a specific burner

**Parameter:**

- burner: device name and burner name

### 8.6.3 `BURN_blank`

**Description:** blanks a CD-RW

**Parameter:**

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

### 8.6.4 `BURN_burn`

**Description:** burns an ISO

**Parameter:**

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

### 8.6.5 `BURN_getStatus`

**Description:** returns the status of the burner (BURNSTATE\_IDLE, BURNSTATE\_BLANK, BURNSTATE\_BURN)

### 8.6.6 `BURN_showLog`

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

### 8.6.7 `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).

### 8.6.8 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.

## 8.7 `./inc/capture.php`

Author: Hauke Goos-Habermann (HHabermann@pc-kiel.de)

Description: routines storing and loading POST and GET values in forms

### 8.7.1 `CAPTURE_getKeys`

**Description:** gets all POST or GET variables and returns all keys and values as an associative array. Values of buttons are filtered out.

**Parameter:**

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

### 8.7.2 `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

### 8.7.3 `CAPTURE_load`

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

**Parameter:**

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

### 8.7.4 `CAPTURE_deActivate`

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

**Parameter:**

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

### 8.7.5 `CAPTURE_isActive`

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

### 8.7.6 `CAPTURE_captureImg`

**Description:** returns the status image URL of the current capture state

### 8.7.7 `CAPTURE_toggle`

**Description:** toggles the current capture state



**8.7.8 CAPTURE\_showMessageBox**

**Description:** shows a message box, if capturing is enabled

**8.7.9 CAPTURE\_showEntries**

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

**8.7.10 CAPTURE\_deleteById**

**Description:** deletes a capture entry.

**Parameter:**

- id: the id of the capture entry to delete

**8.7.11 CAPTURE\_showMarker**

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

**8.7.12 CAPTURE\_showTableWith**

**Description:** Adds a width element if in captureLoad mode.

## 8.8 ./inc/CAutoTest.php

Author: Hauke Goos-Habermann (HHabermann@pc-kiel.de)

Description: Class for test automation.

### 8.8.1 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 nonexistent (to create) client or associative array of parameters.
- checkMode: Check for the input variable.

### 8.8.2 CAutoTest::readAndDefineOrDieXML

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

**Parameter:**

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

### 8.8.3 CAutoTest::readSettings

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

### 8.8.4 CAutoTest::VMCreate

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

### 8.8.5 CAutoTest::VMStart

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

### 8.8.6 CAutoTest::setISO

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

**Parameter:**

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

### 8.8.7 CAutoTest::getISO

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

**Returns:**

- Filename and path to the ISO image for booting.

### 8.8.8 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.

### 8.8.9 CAutoTest::getVM\_hdsiz

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

**Returns:**

- Hard disk size of the VM.

### 8.8.10 CAutoTest::getVMRam

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

**Returns:**

- Ram size of the VM.

### 8.8.11 CAutoTest::getMachine

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

**Returns:**

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

### 8.8.12 CAutoTest::getMovieFileName

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

**Returns:**

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

### 8.8.13 CAutoTest::getLogFileName

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

**Returns:**

- Log file name (with full path).

### 8.8.14 CAutoTest::addToLogFile

**Description:** Appends lines to the log file.

**Parameter:**

- lines: The lines to add.

### 8.8.15 CAutoTest::getTimestampString

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

**Returns:**

- Date and time in human readable form.

### 8.8.16 CAutoTest::setTriggered

**Description:** Sets the trigger state of the current sequence event.

**Parameter:**

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

**Returns:**

- true, when the current sequence event was triggered.

### 8.8.17 CAutoTest::isTriggered

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

**Returns:**

- true, when the current sequence event was triggered.

### 8.8.18 CAutoTest::setTimeout

**Description:** Sets the timeout for the current sequence element.

### 8.8.19 CAutoTest::decTimeout

**Description:** Decrements the remaining time for the timeout.

### 8.8.20 CAutoTest::addToSequence

**Description:** Adds an element to the sequence.

**Parameter:**

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

### 8.8.21 CAutoTest::matchArray

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

**Parameter:**

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

### 8.8.22 **CAutoTest::getAnswersA**

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

**Returns:**

- Array with the answers of the current sequence element.

### 8.8.23 **CAutoTest::getTriggerTypes**

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

**Returns:**

- Trigger type of the current sequence element.

### 8.8.24 **CAutoTest::getTriggerParams**

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

**Returns:**

- Trigger parameter of the current sequence element.

### 8.8.25 **CAutoTest::getExecTypes**

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

**Returns:**

- Type of execution of the current sequence element.

### 8.8.26 **CAutoTest::getExecParams**

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

**Returns:**

- Parameter for execution of the current sequence element.

### 8.8.27 **CAutoTest::checkTriggerResult**

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

### 8.8.28 **CAutoTest::executeTriggerAction**

**Description:** Executes the action of the current sequence element.

### 8.8.29 **CAutoTest::waitForTrigger**

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

### 8.8.30 **CAutoTest::getCurElement**

**Description:** Returns the current sequence element.

**Returns:**

- Trigger type of the current sequence element.

### 8.8.31 CAutoTest::nextCurElement

**Description:** Increments the current sequence element number.

### 8.8.32 CAutoTest::showAndLogMessage

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

**Parameter:**

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

### 8.8.33 CAutoTest::elemOk

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

**Parameter:**

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

### 8.8.34 CAutoTest::elemWarn

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

**Parameter:**

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

### 8.8.35 CAutoTest::elemBad

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

**Parameter:**

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

### 8.8.36 CAutoTest::setVariableFromXML

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

**Parameter:**

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

**Returns:**

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

### 8.8.37 CAutoTest::triggerTypeToConstant

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

**Parameter:**

- type: Trigger type (string).

**Returns:**

- Trigger type constant.

### 8.8.38 **CAutoTest::parseTriggerFromXML**

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

**Parameter:**

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

### 8.8.39 **CAutoTest::parseActionFromXML**

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

**Parameter:**

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

### 8.8.40 **CAutoTest::parseAnswersFromXML**

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

**Parameter:**

- xmlO: Prased XML structure pointing to the array.
- goodWarnBad: Answer type (CAutoTest::GWB\_GOOD, GWB\_WARN or GWB\_BAD).

**Returns:**

- Associative array with the good, warn or bad answers, how to fetch the answer from the client/webbrowser/etc. and the answer type (GWB\_GOOD, GWB\_WARN or GWB\_BAD).

### 8.8.41 **CAutoTest::parseXML**

**Description:** Parses the XML test description file.

**Parameter:**

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

## 8.9 `./inc/CAutoUpdate.php`

Author: Hauke Goos-Habermann (HHabermann@pc-kiel.de)

Description: Class for handling automatic updates.

### 8.9.1 `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.

### 8.9.2 `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.

### 8.9.3 `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.

### 8.9.4 `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

### 8.9.5 `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.



### 8.9.6 **CAutoUpdate::setType**

**Description:** Sets the type for a set.

**Parameter:**

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

### 8.9.7 **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.

### 8.9.8 **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.

### 8.9.9 **CAutoUpdate::delSet**

**Description:** Deletes one set.

**Parameter:**

- id: ID of the set to delete.

**Returns:**

- : true, if the set with the given ID could be deleted, otherwise false.

### 8.9.10 **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 output array (CAutoUpdate::ORDERBY\_ID or CAutoUpdate::ORDERBY\_LASTRUN)

**Returns:**

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

### 8.9.11 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.

### 8.9.12 CAutoUpdate::saveSets

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

**Returns:**

- : true, if a set is active.

### 8.9.13 CAutoUpdate::getSetAmount

**Description:** Get the amount of update sets.

**Returns:**

- : Amount of update sets.

### 8.9.14 CAutoUpdate::configureCron

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

### 8.9.15 CAutoUpdate::isASetActive

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

**Returns:**

- : true, if a set is active.

### 8.9.16 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.

### 8.9.17 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.

### 8.9.18 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.

### 8.9.19 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.

### 8.9.20 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.

### 8.9.21 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).

### 8.9.22 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.

### 8.9.23 CAutoUpdate::delGroup

**Description:** Deletes a group.

**Parameter:**

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

### 8.9.24 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).

### 8.9.25 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 matching the regex was clicked, otherwise false.

### 8.9.26 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.

### 8.9.27 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.

### 8.9.28 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.

### 8.9.29 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.

### 8.9.30 CAutoUpdate::run

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

### 8.9.31 CAutoUpdate::showAutoUpdateManager

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

## 8.10 `./inc/CChecks.php`

Author: Hauke Goos-Habermann (HHabermann@pc-kiel.de)

Description: Class for checking values.

### 8.10.1 `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.

### 8.10.2 `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.

### 8.10.3 `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.

### 8.10.4 `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.

### 8.10.5 `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.

### 8.10.6 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.

### 8.10.7 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.

### 8.10.8 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.

### 8.10.9 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.

### 8.10.10 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.

### 8.10.11 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.

### 8.10.12 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.

### 8.10.13 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.

### 8.10.14 CChecks::checkGroupname

**Description:** Checks if the groupname is valid.

**Parameter:**

- group: Groupname to check.

**Returns:**

- true, if the groupname is valid otherwise false.

### 8.10.15 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.

### 8.10.16 CChecks::checkLanguage

**Description:** Checks if the language is valid.

**Parameter:**

- language: Language value to check.

**Returns:**

- true, if the language is valid otherwise false.

### 8.10.17 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.



#### 8.10.18 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.

#### 8.10.19 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.

#### 8.10.20 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.

#### 8.10.21 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.

#### 8.10.22 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.

### 8.10.23 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.

### 8.10.24 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.

### 8.10.25 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.

### 8.10.26 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.

### 8.10.27 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.

### 8.10.28 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.

### 8.10.29 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.

### 8.10.30 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.

### 8.10.31 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.

### 8.10.32 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.

### 8.10.33 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.

### 8.10.34 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.

### 8.10.35 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.

### 8.10.36 CChecks::checkIPRange

**Description:** Checks if the input IPs are valid and if the first IP is "smaller" 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 "smaller" otherwise false.

### 8.10.37 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.

### 8.10.38 CChecks::checkBootType

**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.

### 8.10.39 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.

### 8.10.40 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.

## 8.11 `./inc/CClientLister.php`

Author: Hauke Goos-Habermann (HHabermann@pc-kiel.de)

Description: Class for generating lists of clients.

### 8.11.1 `CClientLister::showClientTable`

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

### 8.11.2 `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.

### 8.11.3 `CClientLister::importCheckedClientNamesIds`

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

### 8.11.4 `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.

### 8.11.5 `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.

### 8.11.6 `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).

### 8.11.7 `CClientLister::getClientNames`

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

**Returns:**

- Array with all clients matching the filter rules.

**8.11.8 CClientLister::showClientList**

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

**8.11.9 CClientLister::showClientListExtraLine**

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

**8.11.10 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.

**8.11.11 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.

**8.11.12 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.

**8.11.13 CClientLister::cmpArrayElements**

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

**Parameter:**

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

**8.11.14 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.

**8.11.15 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.

**8.11.16 CClientLister::getStatusHeader**

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

**8.11.17 CClientLister::getClientNameHeader**

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

**8.11.18 CClientLister::getInstallDateHeader**

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

**8.11.19 CClientLister::getLastModifyHeader**

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

**8.11.20 CClientLister::getIPHeader**

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

**8.11.21 CClientLister::getMACHeader**

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

**8.11.22 CClientLister::getJobsHeader**

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

**8.11.23 CClientLister::getPackagesHeader**

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

**8.11.24 CClientLister::getGroupHeader**

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

**8.11.25 CClientLister::showClientListHeader**

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

**8.11.26 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.

**8.11.27 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).



### 8.11.28 CClientLister::getActionString

**Description:** Gets the action string.

**Returns:**

- The action string

### 8.11.29 CClientLister::setOrderBy

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

**Parameter:**

- order: ORDERBY\_\* mode to sort the output.

**Returns:**

- true on success.

### 8.11.30 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

### 8.11.31 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.

### 8.11.32 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.

### 8.11.33 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.

### 8.11.34 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.

### 8.11.35 CClientLister::isAscending

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

**Returns:**

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

### 8.11.36 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.

### 8.11.37 CClientLister::resetGetting

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

### 8.11.38 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 could be got.

### 8.11.39 CClientLister::getAllMatchingClients

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

**Returns:**

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

### 8.11.40 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.

### 8.11.41 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.

### 8.11.42 CClientLister::IPexists

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

**Parameter:**

- ip: IP to check

#### 8.11.43 CClientLister::MACexists

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

**Parameter:**

- mac: MAC to check

#### 8.11.44 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

## 8.12 `./inc/CClient.php`

Author: Hauke Goos-Habermann (HHabermann@pc-kiel.de)

Description: Class for handling clients.

### 8.12.1 `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.

### 8.12.2 `CClient::__destruct`

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

### 8.12.3 `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.

### 8.12.4 `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.

### 8.12.5 `CClient::copyImagingParameters`

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

**Parameter:**

- options: Associative array with all options.

### 8.12.6 `CClient::copyMassOptions`

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

**Parameter:**

- options: Associative array with all options.

### 8.12.7 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.

### 8.12.8 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.

### 8.12.9 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

### 8.12.10 CClient::includeDistributionSpecificPackagesPHP

**Description:** Includes distribution specific packages.php.

### 8.12.11 CClient::addNormalJob

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

**Parameter:**

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

### 8.12.12 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).

### 8.12.13 CClient::addUpdateSourcesListJob

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

### 8.12.14 CClient::addUpdatePackageInfosJob

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

### 8.12.15 CClient::addNormalUpdateJob

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

**Parameter:**

- type: normal or complete.

### 8.12.16 CClient::addCompleteUpdateJob

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

**Parameter:**

- type: normal or complete.

### 8.12.17 CClient::addUpdateJob

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

**Parameter:**

- type: normal or complete.

### 8.12.18 CClient::startAutoUpdate

**Description:** Starts auto update on a client.

**Parameter:**

- type: Normal (CAutoUpdate::UPDATE\_NORMAL) or full (CAutoUpdate::UPDATE\_FULL) update.

### 8.12.19 CClient::stopAutoUpdate

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

### 8.12.20 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.

### 8.12.21 CClient::hasAutoUpdateJob

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

**Returns:**

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

### 8.12.22 CClient::checkAutoUpdate

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

**8.12.23 CClient::unsetInstPartDev**

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

**8.12.24 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.

**8.12.25 CClient::setUEFI**

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

**Parameter:**

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

**8.12.26 CClient::isUEFIActive**

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

**Returns:**

- true, if the client uses UEFI otherwise false.

**8.12.27 CClient::isHalfSisterClient**

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

**Returns:**

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

**8.12.28 CClient::getEFIBootPartDev**

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

**Returns:**

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

**8.12.29 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.

**8.12.30 CClient::unsetEFIBootPartDev**

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

### 8.12.31 CClient::getInstPartDev

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

**Returns:**

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

### 8.12.32 CClient::unsetSwapPartDev

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

### 8.12.33 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.

### 8.12.34 CClient::getSwapPartDev

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

**Returns:**

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

### 8.12.35 CClient::isDerivedClient

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

**Returns:**

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

### 8.12.36 CClient::setLanguage

**Description:** Sets the language of the client.

**Parameter:**

- language: Language of the client.

**Returns:**

- true, if the language is valid otherwise false.

### 8.12.37 CClient::getLanguage

**Description:** Gets the language of the client.

**Returns:**

- The language of the client.

### 8.12.38 CClient::setTimeZone

**Description:** Sets the timezone of the client.

**Parameter:**

- timeZone: Timezone of the client.

**Returns:**

- true, if the timezone is valid otherwise false.



### 8.12.39 CClient::getTimeZone

**Description:** Gets the timezone of the client.

**Returns:**

- The timezone of the client.

### 8.12.40 CClient::setBootloader

**Description:** Sets the bootloader of the client.

**Parameter:**

- bootloader: Bootloader of the client.

**Returns:**

- true, if the bootloader is valid otherwise dies.

### 8.12.41 CClient::getBootloader

**Description:** Gets the bootloader of the client.

**Returns:**

- The bootloader of the client.

### 8.12.42 CClient::getClientGroup

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

**Returns:**

- The m23 group of the client.

### 8.12.43 CClient::setNetRootPwd

**Description:** Generates and sets the netboot root password.

### 8.12.44 CClient::getNetRootPwd

**Description:** Gets the netboot root password.

**Returns:**

- Netboot root password.

### 8.12.45 CClient::setNfshomeserver

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

**Parameter:**

- nfshomeserver: NFS share with path.

**Returns:**

- true, if the NFS share is valid.

### 8.12.46 CClient::getNfshomeserver

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

**Returns:**

- NFS share of the client.

### 8.12.47 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 successfully setting.

### 8.12.48 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 successfully setting.

### 8.12.49 CClient::getInstallPrinter

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

**Returns:**

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

### 8.12.50 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 successfully setting.

### 8.12.51 CClient::getAddNewLocalLogin

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

**Returns:**

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

### 8.12.52 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 successfully setting.

### 8.12.53 **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.

### 8.12.54 **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.

### 8.12.55 **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.

### 8.12.56 **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.

### 8.12.57 **CClient::getGroupID**

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

**Returns:**

- Group ID.

### 8.12.58 **CClient::getUserID**

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

**Returns:**

- User ID.

### 8.12.59 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.

### 8.12.60 CClient::getFirstpw

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

**Returns:**

- The user's first password.

### 8.12.61 CClient::getFamilyname

**Description:** Returns the user's familyname.

**Returns:**

- The user's familyname.

### 8.12.62 CClient::getForename

**Description:** Returns the user's forename.

**Returns:**

- The user's forename.

### 8.12.63 CClient::getLogin

**Description:** Returns the user's login.

**Returns:**

- The user's login.

### 8.12.64 CClient::setLDAPType

**Description:** Sets the LDAP type.

**Parameter:**

- type: LDAP usage type to set.

### 8.12.65 CClient::getLDAPType

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

**Returns:**

- LDAP type of the client.

### 8.12.66 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.

### 8.12.67 CClient::getLDAPServer

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

**Returns:**

- LDAP server of the client.

### 8.12.68 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.

### 8.12.69 CClient::getBackToDetailsLink

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

**Parameter:**

- section: section to jump on the page

### 8.12.70 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.

### 8.12.71 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.

### 8.12.72 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).

### 8.12.73 CClient::save

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

### 8.12.74 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).

### 8.12.75 CClient::getSourcesList

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

**Returns:**

- Sources list of the client.

### 8.12.76 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.

### 8.12.77 CClient::getArch

**Description:** Returns the architecture of the client.

**Returns:**

- Architecture of the client.

### 8.12.78 CClient::setDistribution

**Description:** sets the distribution of the client.

**Parameter:**

- distr: Distribution of the client.

**Returns:**

- true on sucessfully setting, otherwise false.

### 8.12.79 CClient::getDistribution

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

**Returns:**

- Distribution of the client.

**8.12.80 CClient::setRelease**

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

**Parameter:**

- release: Distribution release of the client.

**Returns:**

- true on successfully setting, otherwise false.

**8.12.81 CClient::getRelease**

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

**Returns:**

- Distribution release of the client.

**8.12.82 CClient::updateModifyDate**

**Description:** Updates the last modified date.

**8.12.83 CClient::getModifyDate**

**Description:** Returns the last modified date.

**Returns:**

- Last modified date of the client.

**8.12.84 CClient::getModifyDateHumanReadable**

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

**Returns:**

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

**8.12.85 CClient::updateInstallDate**

**Description:** Updates the installation date.

**8.12.86 CClient::getInstallDate**

**Description:** Returns the installation date.

**Returns:**

- Installation date of the client.

**8.12.87 CClient::getInstallDateHumanReadable**

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

**Returns:**

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

**8.12.88 CClient::export**

**Description:** Exports all client settings as associativearray

### 8.12.89 CClient::setClientName

**Description:** Renames a client.

**Parameter:**

- clientName: New name of the client.

**Returns:**

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

### 8.12.90 CClient::getClientName

**Description:** Returns the name of the client.

**Returns:**

- Name of the client.

### 8.12.91 CClient::getID

**Description:** Returns the ID of the client.

**Returns:**

- ID of the client.

### 8.12.92 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.

### 8.12.93 CClient::getDNS1

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

**Returns:**

- Main DNS server IP of the client.

### 8.12.94 CClient::getDNS2

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

**Returns:**

- Backup DNS server IP of the client.

### 8.12.95 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.



### 8.12.96 CClient::setIP

**Description:** Sets the IP of the client to an unused IP.

**Parameter:**

- IP: unused IP for the client.

**Returns:**

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

### 8.12.97 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.

### 8.12.98 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.

### 8.12.99 CClient::getNetmask

**Description:** Returns the client's netmask.

**Returns:**

- Netmask of the client.

### 8.12.100 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.

### 8.12.101 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.

### 8.12.102 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.

### 8.12.103 CClient::getGateway

**Description:** Returns the client's gateway.

**Returns:**

- Gateway of the client.

### 8.12.104 CClient::setStatus

**Description:** Sets the client's status.

**Parameter:**

- status: The status number to check.

**Returns:**

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

### 8.12.105 CClient::getStatus

**Description:** Returns the client's status.

**Returns:**

- Status of the client.

### 8.12.106 CClient::isDefinedClient

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

**Returns:**

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

### 8.12.107 CClient::generateHTMLStatusBar

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

### 8.12.108 CClient::getBootType

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

**Returns:**

- Network boot type of the client.

### 8.12.109 CClient::usesDynamicIP

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

**Returns:**

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

**8.12.110 CClient::setBootType**

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

**Parameter:**

- bootType: CClient::BOOTTYPE\_PXE, CClient::BOOTTYPE\_NOBOOT, CClient::BOOTTYPE\_ETHERBOOT, CClient::BOOTTYPE\_GRUB2EFI64

**Returns:**

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

**8.12.111 CClient::getNetworkBootTypesArrayForSelection**

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

**Returns:**

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

**8.12.112 CClient::startInstall**

**Description:** Starts the installation on the client

**8.12.113 CClient::addShutdownPackage**

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

**Returns:**

- true, if a shutdown package is added.

**8.12.114 CClient::activateNetboot**

**Description:** Activates network booting for the client.

**Returns:**

- true on successfully restarting the DHCP server.

**8.12.115 CClient::deactivateNetboot**

**Description:** Deactivates network booting for the client.

**Returns:**

- true on successfully restarting the DHCP server.

**8.12.116 CClient::wol**

**Description:** Wakes a client over the network.

**8.12.117 CClient::isNetbootActive**

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

**Returns:**

- true when network booting is active otherwise false.

### 8.12.118 CClient::isPingable

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

### 8.12.119 CClient::sshFetchJob

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

### 8.12.120 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.

### 8.12.121 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.

### 8.12.122 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.

### 8.12.123 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.

### 8.12.124 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.

**8.12.125 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.

**8.12.126 CClient::destroy**

**Description:** Destroys a client finally.

**8.12.127 CClient::getVMRole**

**Description:** Returns the client's vmRole.

**Returns:**

- vmRole of the client.

**8.12.128 CClient::getVmSoftware**

**Description:** Returns the client's vmSoftware.

**Returns:**

- vmSoftware of the client.

## 8.13 `./inc/CFDiskBasic.php`

Author: Hauke Goos-Habermann (HHabermann@pc-kiel.de)

Description: Class for basic partitioning and formatting functions.

### 8.13.1 `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.

### 8.13.2 `CFDiskBasic::__destruct`

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

### 8.13.3 `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.

### 8.13.4 `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`)

### 8.13.5 `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

### 8.13.6 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

### 8.13.7 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

### 8.13.8 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.

### 8.13.9 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)

### 8.13.10 CFDiskBasic::makeInstOrEFIPartBootable

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

### 8.13.11 CFDiskBasic::rereadPartTable

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

**Parameter:**

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

### 8.13.12 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).

### 8.13.13 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.

### 8.13.14 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.

### 8.13.15 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.

### 8.13.16 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.

### 8.13.17 CFDiskBasic::formatPartition

**Description:** Formats a partition.

**Parameter:**

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



#### 8.13.18 **CFDiskBasic::deleteAllPartitions**

**Description:** Deletes all partitions on a disk.

**Parameter:**

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

#### 8.13.19 **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).

#### 8.13.20 **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).

#### 8.13.21 **CFDiskBasic::autoPartitionDisk**

**Description:** Automatically partitions and formats a disk.

**Parameter:**

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

#### 8.13.22 **CFDiskBasic::PM\_auto2Disk1SysSwap2Home**

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

#### 8.13.23 **CFDiskBasic::createAllRaidJobs**

**Description:** Generates the jobs to create all RAID.

**Returns:**

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

#### 8.13.24 **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.

### 8.13.25 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.

### 8.13.26 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.

### 8.13.27 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.

### 8.13.28 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.

### 8.13.29 CFDiskBasic::deleteAllPartitionsOnDisk

**Description:** Deletes all partitions on a disk.

**Parameter:**

- dev: Device name of the disk.

### 8.13.30 CFDiskBasic::fdiskAdjustPartitioningLinearScale

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

**Parameter:**

- diskDevToAdjust: Device name of the disk to adjust.

### 8.13.31 CFDiskBasic::fdiskAdjustPartitioning

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

## 8.14 ./inc/CFDiskGUI.php

Author: Hauke Goos-Habermann (HHabermann@pc-kiel.de)

Description: Class for visualisation of partitioning and formatting.

### 8.14.1 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.

### 8.14.2 CFDiskGUI::\_\_destruct

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

### 8.14.3 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.

### 8.14.4 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.

### 8.14.5 FDISK\_fdiskSessionReset

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

**Parameter:**

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

### 8.14.6 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.

### 8.14.7 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.

### 8.14.8 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.

### 8.14.9 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.

### 8.14.10 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.

### 8.14.11 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.

#### 8.14.12 CFDiskGUI::getFileSystemTranslator

**Description:** Translates the filesystem names.

**Parameter:**

- fs: Filesystem to translate.

**Returns:**

- Translated filesystem.

#### 8.14.13 CFDiskGUI::getPartitionTypeTranslator

**Description:** Translates the partition types.

**Parameter:**

- type: Partition type to translate.

**Returns:**

- Translated partition type.

#### 8.14.14 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.

#### 8.14.15 CFDiskGUI::getDiskLayoutEntriesAmount

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

**Parameter:**

- vDisk: Virtual (internally used) device number.

**Returns:**

- The amount of disk layout entries.

#### 8.14.16 CFDiskGUI::getFreeSpaceHTMLColor

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

**Returns:**

- HTML color for marking free space.

#### 8.14.17 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.

#### 8.14.18 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.

#### 8.14.19 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.

#### 8.14.20 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.

#### 8.14.21 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.

**8.14.22 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.

**8.14.23 CFDiskGUI::showPartTable****Description:** Shows the partition information for a disk as table.**Parameter:**

- vDisk: Virtual (internally used) device number.

**8.14.24 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)

**8.14.25 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

**8.14.26 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

**8.14.27 CFDiskGUI::showCombinedFdiskGUIDialog****Description:** Shows the new partition and formatting screen.

#### 8.14.28 CFDiskGUI::getCurrentDiskDev

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

**Returns:**

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

#### 8.14.29 CFDiskGUI::getCurrentDiskvDev

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

**Returns:**

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

#### 8.14.30 CFDiskGUI::getCurrentPartDev

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

**Returns:**

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

#### 8.14.31 CFDiskGUI::finalChecksAndRealPartitionAndFormatStart

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

#### 8.14.32 CFDiskGUI::showFdiskCombinedGUIFunctions

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

#### 8.14.33 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)

#### 8.14.34 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.



### 8.14.35 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

### 8.14.36 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 partition can be created.

### 8.14.37 CFDiskGUI::printAllBars

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

### 8.14.38 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.

### 8.14.39 CFDiskGUI::getPartInfoString

**Description:** Generates an info string, that shows information about the device name of the partition, its filesystem and belonging 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.

#### 8.14.40 CFDiskGUI::getDiskInfoString

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

**Parameter:**

- vDisk: Virtual (internally used) device number.

**Returns:**

- Info string.

#### 8.14.41 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.

#### 8.14.42 CFDiskGUI::listPartJobs

**Description:** Show all part jobs in the table.

#### 8.14.43 CFDiskGUI::showAllPartTables

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

#### 8.14.44 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.

#### 8.14.45 CFDiskGUI::showColorDefinitions

**Description:** Shows the color definitions for the supported filesystems.

#### 8.14.46 CFDiskGUI::fstabAddDialog

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

#### 8.14.47 CFDiskGUI::getFstabTable

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

**Returns:**

- HTML table with the fstab.

**8.14.48 CFDiskGUI::getHDSizes**

**Description:** Returns 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.

## 8.15 `./inc/CFDiskIO.php`

Author: Hauke Goos-Habermann (HHabermann@pc-kiel.de)

Description: Class for lowlevel organisation of partitioning and formating.

### 8.15.1 `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.

### 8.15.2 `CFDiskIO::__destruct`

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

### 8.15.3 `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.

### 8.15.4 `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).

### 8.15.5 `CFDiskIO::getCurrentPartitioning`

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

**Returns:**

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

### 8.15.6 `CFDiskIO::setCurrentPartitioning`

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

### 8.15.7 CFDiskIO::fdiskLoadFromDB

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

### 8.15.8 CFDiskIO::setCFDiskTemp

**Description:** Sets CFDiskTemp of the client.

**Parameter:**

- CFDiskTemp: Array with the partition and format information.

### 8.15.9 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.

### 8.15.10 CFDiskIO::getDiskAmount

**Description:** Returns amount of disks.

**Returns:**

- Amount of disks.

### 8.15.11 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.

### 8.15.12 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.

### 8.15.13 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.

#### 8.15.14 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.

#### 8.15.15 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.

#### 8.15.16 CFDiskIO::backToPreviousPartitionStep

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

#### 8.15.17 CFDiskIO::findAndSetEFIBootPartDev

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

#### 8.15.18 CFDiskIO::discardUndo

**Description:** Discards all undo steps.

#### 8.15.19 CFDiskIO::updateUndoMd5

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

#### 8.15.20 CFDiskIO::getUndoMd5

**Description:** Gets the md5 sum of the partitionSteps arrays.

#### 8.15.21 CFDiskIO::addUndo

**Description:** Adds an undo step to the undoArray.

### 8.15.22 **CFDiskIO::addPartitionStepAtTheBeginning**

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

**Parameter:**

- partJob: The job to add.

### 8.15.23 **CFDiskIO::addPartitionStepBeforeFormat**

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

**Parameter:**

- partJob: The RAID creation commands to add.

### 8.15.24 **CFDiskIO::addPartitionStepBeforeFormatArray**

**Description:** Adds a step to the partitionSteps and partitionStepsForShift arrays before the forming 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 forming job for the RAID device. false, if the new job was placed at the end.

### 8.15.25 **CFDiskIO::addPartitionStep**

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

**Parameter:**

- cmd: The commands to add.

### 8.15.26 **CFDiskIO::shiftPartitionStep**

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

### 8.15.27 **CFDiskIO::resetWantedPartitioningAndSteps**

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

### 8.15.28 **CFDiskIO::resetWantedPartitioning**

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

### 8.15.29 **CFDiskIO::fdiskSaveToDB**

**Description:** Saves CFDiskTemp values to the DB.

### 8.15.30 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.

### 8.15.31 CFDiskIO::saveDefinedDiskSizesToDB

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

### 8.15.32 CFDiskIO::getDefinedDiskSizes

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

**Returns:**

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

### 8.15.33 CFDiskIO::convertPartitioning2Array

**Description:** Converts old partitioning data to array.

### 8.15.34 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.

### 8.15.35 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.

### 8.15.36 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.



### 8.15.37 **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 the RAID lock, false for removing the lock.

### 8.15.38 **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.

### 8.15.39 **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.

### 8.15.40 **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.

### 8.15.41 **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.

### 8.15.42 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)

### 8.15.43 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.

### 8.15.44 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.

### 8.15.45 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.

### 8.15.46 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.

#### 8.15.47 **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.

#### 8.15.48 **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.

#### 8.15.49 **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).

#### 8.15.50 **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.

### 8.15.51 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.

### 8.15.52 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.

### 8.15.53 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.

### 8.15.54 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.

### 8.15.55 **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)

### 8.15.56 **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.

### 8.15.57 **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).

### 8.15.58 **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.

### 8.15.59 **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.

### 8.15.60 **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.

### 8.15.61 CFDiskIO::devNrExists

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

**Parameter:**

- vDisk: Virtual (internally used) device number.
- devNr: Device number you want to check (e.g. 2 for /dev/sda2)

**Returns:**

- true, when the (physical) partition number exists on the disk.

### 8.15.62 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.

### 8.15.63 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.

### 8.15.64 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.

**8.15.65 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.

**8.15.66 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.

**8.15.67 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.

**8.15.68 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 equal to the partition number of the real drive (e.g. /dev/hda5 can be \$vPart == 3). If it is set to "empty", only vDev is calculated.

**Returns:**

- true if the search worked otherwise false.

### 8.15.69 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.

### 8.15.70 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 there is free space, otherwise false.

### 8.15.71 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.

### 8.15.72 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.



### 8.15.73 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.

### 8.15.74 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.

### 8.15.75 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.

### 8.15.76 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.

### 8.15.77 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.

### 8.15.78 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.

### 8.15.79 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.

### 8.15.80 CFDiskIO::getWantedPartitioning

**Description:** Returns the wantedPartitioning array.

**Returns:**

- wantedPartitioning array.

### 8.15.81 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.

### 8.15.82 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.

### 8.15.83 CFDiskIO::getRaidLevelNumbers

**Description:** Returns an array with valid RAID levels.

**Returns:**

- Array with valid RAID levels.

#### 8.15.84 **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.

#### 8.15.85 **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.

#### 8.15.86 **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.

#### 8.15.87 **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.

#### 8.15.88 **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.

### 8.15.89 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.

### 8.15.90 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.

### 8.15.91 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.

### 8.15.92 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.

### 8.15.93 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.

#### 8.15.94 **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.

#### 8.15.95 **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.

#### 8.15.96 **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.

#### 8.15.97 **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.

#### 8.15.98 **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.

#### 8.15.99 **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.

### 8.15.100 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 contain the filter string (without the "!").
- **addSizesAndTypes:** If set to true, the array will contain the sizes, filesystems and types of the partitions and drives.
- **filterOutSetRaidLvmLock:** If set to true, drives and partitions with set raidLvmLock will not be listed.

**Returns:**

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

### 8.15.101 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.

### 8.15.102 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.

### 8.15.103 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 used MDs (e.g. /dev/md0, /dev/md1, ...) as key and value.

### 8.15.104 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.

### 8.15.105 CFDiskIO::getNotFS

**Description:** Generates and returns an array with the list of not file systems (free space or unformatted partitions).

**Returns:**

- Associative array with the list of not file systems as key and value.

**8.15.106 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.

**8.15.107 CFDiskIO::swapFilesystems**

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

**Returns:**

- Array with the filesystems usable for swapping.

**8.15.108 CFDiskIO::installFilesystems**

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

**Returns:**

- Array with the filesystems usable for installation.

**8.15.109 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.

**8.15.110 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.

**8.15.111 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.

### 8.15.112 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

### 8.15.113 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.

### 8.15.114 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.

### 8.15.115 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

### 8.15.116 CFDiskIO::fdiskGetFstabArray

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

**Returns:**

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

### 8.15.117 CFDiskIO::getPartitionStepsArray

**Description:** Gets the partitionSteps array of a client.

**Returns:**

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



**8.15.118 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.

**8.15.119 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).

**8.15.120 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.

**8.15.121 CFDiskIO::getNVMeMknodCommand**

**Description:** Generates the mknod commands for creating the NVMe devices.

**Returns:**

- mknod commands for creating the NVMe devices.

**8.15.122 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).

## 8.16 `./inc/CFDiskTest.php`

Author: Hauke Goos-Habermann (HHabermann@pc-kiel.de)

Description: Class for testing partitioning and formatting functions.

### 8.16.1 `CFDiskTest::getTestDiskDev`

**Description:** Gets the disk device that is used for testing.

**Returns:**

- Disk device that is used for testing.

### 8.16.2 `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.

### 8.16.3 `CFDiskTest::addLineToReplay`

**Description:** Adds a line to the run log file.

**Parameter:**

- line: The line to add.

**Returns:**

- true on successfully writing the line, otherwise false.

### 8.16.4 `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.

### 8.16.5 `CFDiskTest::getTestDiskvDisk`

**Description:** Gets the vDisk for the testing fisk.

**Returns:**

- vDisk for the testing fisk.

### 8.16.6 `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.

### 8.16.7 **CFDiskTest::getRadomStartEnd**

**Description:** Gets random start / end positions (in MB) for e.g. creating new partitions on the test disk.

### 8.16.8 **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.

### 8.16.9 **CFDiskTest::checkConsistencyDiskKey**

**Description:** Checks for consistency 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 allowed difference between the (numeric) values.

**Returns:**

- true if virtual calculation and actual client value are equal and both present, otherwise false.

### 8.16.10 **CFDiskTest::checkConsistencyvDiskvPartKey**

**Description:** Checks for consistency 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 allowed difference between the (numeric) values.

**Returns:**

- true if virtual calculation and actual client value are equal and both present, otherwise false.

### 8.16.11 **CFDiskTest::checkPartitionConsistency**

**Description:** Checks for consistency of the partitioning between virtual calculation and actual client partitioning. Exists the script, if it is not consistent.

### 8.16.12 **CFDiskTest::getDiskArrayFromClient**

**Description:** Returns the disk array with all drives and their partitions.

**Returns:**

- Disk array with all drives and their partitions.

### 8.16.13 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.

### 8.16.14 CFDiskTest::getPartInfoFromClient

**Description:** Fetches current partitioning and formatting of the current client from a given disk.

**Parameter:**

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

**Returns:**

- Array with current partitioning and formatting of the current client from a given disk.

### 8.16.15 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.

### 8.16.16 CFDiskTest::executePartedCommands

**Description:** Executes all partitioning and formatting commands on the given client.

### 8.16.17 CFDiskTest::createPartition

**Description:** Creates a new partition on a disk (if possible) and checks for consistency of the partitioning 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)

### 8.16.18 CFDiskTest::randomCreatePartition

**Description:** Tries to create a new partition on the test disk with random type and start and end position.

### 8.16.19 CFDiskTest::deletePartition

**Description:** Deletes a partition from a disk and checks for consistency of the partitioning 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.

#### 8.16.20 **CFDiskTest::randomDeletePartition**

**Description:** Tries to delete a randomly picked partition on the test disk.

#### 8.16.21 **CFDiskTest::formatPartition**

**Description:** Formats a partition.

**Parameter:**

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

#### 8.16.22 **CFDiskTest::randomFormatPartition**

**Description:** Tries to format a randomly picked partition on the test disk.

#### 8.16.23 **CFDiskTest::getRandomFilesystem**

**Description:** Returns an random filesystem.

**Returns:**

- Random filesystem.

#### 8.16.24 **CFDiskTest::getRandomPartitionType**

**Description:** Returns an random partition type.

**Returns:**

- Random partition type.

#### 8.16.25 **CFDiskTest::createPartitionsForRaid**

**Description:** Create partitions for usage in a RAID.

#### 8.16.26 **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.

#### 8.16.27 **CFDiskTest::randomPartTest**

**Description:** Randomly creates, formats and deletes partitions of random size, type and with random file systems.

#### 8.16.28 **CFDiskTest::getRandomRaidLevel**

**Description:** Returns an random RAID level.

**Returns:**

- Random RAID level.

### 8.16.29 CFDiskTest::getRandomUnusedMD

**Description:** Returns a random unused MD.

**Returns:**

- Random unused MD.

### 8.16.30 CFDiskTest::getRandomUsedMD

**Description:** Returns a random used MD.

**Returns:**

- Random used MD.

### 8.16.31 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.

### 8.16.32 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.

### 8.16.33 CFDiskTest::randomCreateRaid

**Description:** Tries to create a new RAID with randomly choosen partitions and with random RAID level.

### 8.16.34 CFDiskTest::randomDeleteRaid

**Description:** Picks a random RAID and tries to delete it.

### 8.16.35 CFDiskTest::randomFormatRaid

**Description:** Picks a random RAID and tries to delete it.

### 8.16.36 CFDiskTest::randomRaidTest

**Description:** Randomly creates, formates and deletes partitions of random size, type and with random file file systems.

### 8.16.37 CFDiskTest::randomPartTest

**Description:** Randomly creates, formates and deletes partitions of random size, type and with random file file systems.

## 8.17 *./inc/CFirewall.php*

Author: Hauke Goos-Habermann (HHabermann@pc-kiel.de)

Description: Class for managing firewall settings.

### 8.17.1 *CFirewall::\_\_construct*

**Description:** Constructor for new CFirewall 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.

### 8.17.2 *CFirewall::clearIPtablesSettings*

**Description:** Clears all currently set iptables rules.

### 8.17.3 *CFirewall::scriptFileExists*

**Description:** Checks, if the iptables script file exists.

**Returns:**

- true, if the iptables script file exists, otherwise false.

### 8.17.4 *CFirewall::rulesFileExists*

**Description:** Checks, if the iptables rules file exists.

**Returns:**

- true, if the iptables rules file exists, otherwise false.

### 8.17.5 *CFirewall::writeDefaultScript*

**Description:** Writes the default iptables script, if there is no script.

### 8.17.6 *CFirewall::getScript*

**Description:** Gets the contents of the iptables script.

**Returns:**

- Contents of the iptables script.

### 8.17.7 *CFirewall::putScript*

**Description:** Writes script code in the iptables script.

**Parameter:**

- text: The iptables script code to write.

### 8.17.8 *CFirewall::executeScript*

**Description:** Executes the iptables script.

### 8.17.9 CFirewall::exportRules

**Description:** Exports the set iptables rules to a rule file (generated with iptables-save).

### 8.17.10 CFirewall::activateInInterfaces

**Description:** Activates auto-loading of the iptables rules in the interfaces file.

**Returns:**

- true, if the auto-loading is not already active, otherwise false.

### 8.17.11 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.

### 8.17.12 CFirewall::activate

**Description:** Activates the firewall.

**Returns:**

- true, if the firewall was activated, otherwise false.

### 8.17.13 CFirewall::deactivate

**Description:** Deactivates the firewall.

**Returns:**

- true, if the firewall was deactivated, otherwise false.

### 8.17.14 CFirewall::getStatusIconHTML

**Description:** Generates a HTML status icon to indicate the status of the firewall.

### 8.17.15 CFirewall::show

**Description:** Shows a dialog for (de)activating the firewall and changing the rules.



## 8.18 *./inc/CGPGSign.php*

Author: Hauke Goos-Habermann (HHabermann@pc-kiel.de)

Description: Class for managing GPG keys and signing messages.

### 8.18.1 *CGPGSign::\_\_construct*

**Description:** Constructor for new CGPGSign objects.

**Parameter:**

- mode: Save a (new) configuration file or load a (required) configuration file.

### 8.18.2 *CGPGSign::getKeySelectionDialog*

**Description:** Generates a dialog to choose the GPG used for signing the pools.

### 8.18.3 *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.

### 8.18.4 *CGPGSign::exportPublicSignKey*

**Description:** Exports the public key to the webserver directory.

**Parameter:**

- true, if the file was exported, otherwise false.

### 8.18.5 *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.

### 8.18.6 *CGPGSign::getKeyInfo*

**Description:** Gets information about the used GPG key.

**Returns:**

- : Information about the used GPG key.

### 8.18.7 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.

### 8.18.8 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.

### 8.18.9 CGPGSign::hasConfigFile

**Description:** Checks, if the config file exists.

**Returns:**

- true, if the config file is present.

### 8.18.10 CGPGSign::loadConfigFile

**Description:** Loads the config file.

### 8.18.11 CGPGSign::writeConfigFile

**Description:** Writes the config file.

### 8.18.12 CGPGSign::setGPGID

**Description:** Sets the GPG ID to use.

### 8.18.13 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.

#### 8.18.14 CGPGSign::setStoreMode

**Description:** Sets the configuration file store or load mode.

**Parameter:**

- mode: Configuration file store or load mode.

#### 8.18.15 CGPGSign::getStoreMode

**Description:** Gets the configuration file store or load mode.

**Returns:**

- mode: Configuration file store or load mode.

#### 8.18.16 CClient::\_\_destruct

**Description:** Destructor for a CGPGSign object.

## 8.19 ./inc/checks.php

Author: Daniel Kasten (DKasten@pc-kiel.de), Hauke Goos-Habermann (hauke@goos-habermann.de)

Description: different checks for validation of email, ip, netmasks, etc.

### 8.19.1 CHECK\_ip

**Description:** Checks, if an IP is (syntactically) valid.

**Parameter:**

- ip: IP to check.

**Returns:**

- true, if the IP is valid, otherwise false.

### 8.19.2 CHECK\_safeFilename

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

**Parameter:**

- fileName: File/directory name to make safe.

**Returns:**

- The safe made file/directory name.

### 8.19.3 CHECK\_text2db

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

**Parameter:**

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

**Returns:**

- The safe made text.

### 8.19.4 CHECK\_db2text

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

**Parameter:**

- val: String to use.

**Returns:**

- The safe made text.

### 8.19.5 CHECK\_FW

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

**Parameter:**

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

### 8.19.6 CHECK\_deviceName

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

**Parameter:**

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

**Returns:**

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

### 8.19.7 CHECK\_mountPoint

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

**Parameter:**

- mountpoint: Mountpoint to check.

**Returns:**

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

### 8.19.8 CHECK\_int

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

**Parameter:**

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

**Returns:**

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

### 8.19.9 CHECK\_float

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

**Parameter:**

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

**Returns:**

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

### 8.19.10 CHECK\_strAlpha

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

**Parameter:**

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

**Returns:**

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

### 8.19.11 CHECK\_strAlphaNum

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

**Parameter:**

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

**Returns:**

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

### 8.19.12 CHECK\_letFWDie

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

**Parameter:**

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

### 8.19.13 CHECK\_str

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

**Parameter:**

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

**Returns:**

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

#### 8.19.14 countLinesInFile

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

**Parameter:**

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

#### 8.19.15 checkIP

**Description:** checks if an ip is valid

**Parameter:**

- string: ip value to check

**Returns:**

- true if IP is valid, else false

#### 8.19.16 checkMAC

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

**Parameter:**

- mac: MAC address to test.

**Returns:**

- 

#### 8.19.17 checkNetmask

**Description:** checks if a netmask is valid

**Parameter:**

- string: netmask value to check

**Returns:**

- true if netmask is valid, else false

#### 8.19.18 checkEmail

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

**Parameter:**

- string: email address value to check

#### 8.19.19 checkFQDN

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

**Parameter:**

- string: string to check for special characters

#### 8.19.20 checkNormalKeys

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

**Parameter:**

- string: string to check for special characters

## 8.20 `./inc/CIPRanges.php`

Author: Hauke Goos-Habermann (HHabermann@pc-kiel.de)

Description: Class for handling ranges of IP addresses.



## 8.21 *./inc/client\_details.php*

Author: Hauke Goos-Habermann (HHabermann@pc-kiel.de)

Description: Functions for drawing the buttons etc. in client\_details.php.

### 8.21.1 **CLIENT\_DETAILS\_beginCategory**

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

**Parameter:**

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

### 8.21.2 **CLIENT\_DETAILS\_endCategory**

**Description:** Ends the previously opened icon section.

### 8.21.3 **CLIENT\_DETAILS\_addIcon**

**Description:** Adds an icon in a section.

**Parameter:**

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

### 8.21.4 **CLIENT\_DETAILS\_addIcon2**

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

**Parameter:**

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

## 8.22 `./inc/client.php`

Author: Daniel Kasten (DKasten@pc-kiel.de), Hauke Goos-Habermann (HHabermann@pc-kiel.de)

Description: a lot of routines for client handling. routines for: install and deinstall packages on clients, get information about a special client, list all clients,...

### 8.22.1 `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`.

### 8.22.2 `CLIENT_addClientlogsOk`

**Description:** Adds an entry into the client's status log with "ok" status.

**Parameter:**

- `clientName`: Name of the client.
- `text`: Status text.

### 8.22.3 `CLIENT_addClientlogsFailure`

**Description:** Adds an entry into the client's status log with "failure" status.

**Parameter:**

- `clientName`: Name of the client.
- `text`: Status text.

### 8.22.4 `CLIENT_addClientlogsUnknown`

**Description:** Adds an entry into the client's status log with "unknown" status.

**Parameter:**

- `clientName`: Name of the client.
- `text`: Status text.

### 8.22.5 `CLIENT_removeServerCache`

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

**Parameter:**

- `client`: Name of the client.

### 8.22.6 `CLIENT_touchLiveLogFile`

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

**Parameter:**

- `client`: Name of the client.

**Returns:**

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

### 8.22.7 CLIENT\_getNextFreeIp

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

**Returns:**

- Next free IP address.

### 8.22.8 CLIENT\_getAllAsRes

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

**Parameter:**

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

**Returns:**

- MySQL resource ID.

### 8.22.9 CLIENT\_touchLogFile

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

**Parameter:**

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

**Returns:**

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

### 8.22.10 CLIENT\_liveLogJobName

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

**Parameter:**

- client: Name of the client.

**Returns:**

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

### 8.22.11 CLIENT\_stopLiveScreenRecording

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

**Parameter:**

- client: Name of the client.

### 8.22.12 CLIENT\_startLiveScreenRecording

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

**Parameter:**

- client: Name of the client.

### 8.22.13 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.

### 8.22.14 CLIENT\_getOverviewSearchLine

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

**Parameter:**

- amount: Amount of client search dialogs to check.

**Returns:**

- The current client search term.

### 8.22.15 CLIENT\_showOverviewSearchDialog

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

**Parameter:**

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

### 8.22.16 CLIENT\_extraWebAction

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

**Parameter:**

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

**Returns:**

- True if the action was executed or false otherwise.

### 8.22.17 CLIENT\_getOption

**Description:** Returns an option of a client.

**Parameter:**

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

**Returns:**

- Value of the option the client uses.

### 8.22.18 CLIENT\_getDistribution

**Description:** Returns the distribution of a client.

**Parameter:**

- client: Name of the client.

**Returns:**

- Name of the distribution, the client uses.

### 8.22.19 CLIENT\_runDebconf

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

**Parameter:**

- clientName: Name of the client.

### 8.22.20 CLIENT\_copyDebconfDB

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

**Parameter:**

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

### 8.22.21 CLIENT\_setDebconfDB

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

**Parameter:**

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

**Returns:**

- debconf for debconf-set-selections.

### 8.22.22 CLIENT\_getDebconfDB

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

**Parameter:**

- clientName: Name of the client.

**Returns:**

- debconf for debconf-set-selections.

### 8.22.23 CLIENT\_getDebconfDBValue

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

**Parameter:**

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

**Returns:**

- Value of the package variable.

### 8.22.24 CLIENT\_getAllClientNames

**Description:** Gets the names of all clients.

**Returns:**

- Array with the names of all clients.

### 8.22.25 CLIENT\_getClientAmount

**Description:** Gets the amount of all clients.

**Returns:**

- Amount of all clients.

### 8.22.26 CLIENT\_getCurrentMemoryUsage

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

**Parameter:**

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

**Returns:**

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

### 8.22.27 CLIENT\_getCurrentFreeSpaceInDir

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

**Parameter:**

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

**Returns:**

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

### 8.22.28 CLIENT\_getClientID

**Description:** Returns the ID of the calling client.

### 8.22.29 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:**

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

### 8.22.30 CLIENT\_executeOnClientOrIP

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

**Parameter:**

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

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

**Returns:**

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

**8.22.31 CLIENT\_isBasesystemInstalledFromImage**

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

**Parameter:**

- `options`: Array with the client options.

**Returns:**

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

**8.22.32 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['ldapservice']: name of the LDAP server
- options['nfs homeserver']: NFS home server with full path
- options['login']: login name for the user
- options['userID']: user ID for the LDAP account
- options['groupID']: group ID for the LDAP account
- options['addNewLocalLogin'] = \$\_POST[addNewLocalLogin];
- options['timeZone']: POSIX timezone
- options['getSystemtimeByNTP']: "yes", if the system time should be set with NTP
- options['installPrinter']: "yes", if printer drivers should be installed and printers detected
- clientAddType: can be CLIENT\_ADD\_TYPE\_add if the client should be added, CLIENT\_ADD\_TYPE\_define if it should be defined for mass installation or CLIENT\_ADD\_TYPE\_assimilate if the client should be imported into the m23 system.
- cryptRootPw: set to true, if the password should be encrypted or false, if it's already encrypted

### 8.22.33 CLIENT\_IPexists

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

**Parameter:**

- ip: IP to check

### 8.22.34 CLIENT\_MACexists

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

**Parameter:**

- mac: MAC to check

### 8.22.35 CLIENT\_exists

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

**Parameter:**

- clientName: name of the client

### 8.22.36 CLIENT\_getAskingParams

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

### 8.22.37 CLIENT\_getParams

**Description:** returns database parameters of a special client

**Parameter:**

- clientName: name of the client



### 8.22.38 CLIENT\_getClientStatus

**Description:** returns the current client status

**Parameter:**

- client: name of the client

### 8.22.39 CLIENT\_getProperty

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

**Parameter:**

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

### 8.22.40 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 the action selection button, if you set it to true, you can delete packages and discard package deletion jobs

### 8.22.41 CLIENT\_getPossibleActions

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

**Parameter:**

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

### 8.22.42 CLIENT\_acceptChanges

**Description:** removes packages or discards changes

**Parameter:**

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

### 8.22.43 CLIENT\_showHardwareInfo

**Description:** prints a table with hardware informations

**Parameter:**

- client: name of the client

### 8.22.44 CLIENT\_showGeneralInfo

**Description:** prints a table with general information

**Parameter:**

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

### 8.22.45 CLIENT\_showWaitingJobs

**Description:** shows the waiting jobs for the client

**Parameter:**

- client: name of the client

### 8.22.46 CLIENT\_setLastmodify

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

**Parameter:**

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

### 8.22.47 CLIENT\_getNetmaskBits

**Description:** Gets the amount of set bits of a netmask.

**Parameter:**

- netmask: IPv4 netmask.

**Returns:**

- Amount of set bits of a netmask.

### 8.22.48 CLIENT\_getNetmaskFromBitAmount

**Description:** Gets the netmask by the amount of set bits.

**Parameter:**

- bits: Amount if set bits of a netmask.

**Returns:**

- IPv4 netmask

### 8.22.49 CLIENT\_getSubnet

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

**Parameter:**

- ip: ip address
- netmask: netmask

### 8.22.50 CLIENT\_getBroadcast

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

**Parameter:**

- ip: ip address
- netmask: netmask

### 8.22.51 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 leeeave it blank, splitting characters will be removed

### 8.22.52 CLIENT\_getIPbyName

**Description:** returns the ip from a selected clientname

**Parameter:**

- clientName: name of the client

### 8.22.53 CLIENT\_getNamebyIP

**Description:** returns the clientname from a selected ip

**Parameter:**

- ip: ip of the client

### 8.22.54 CLIENT\_getMACbyName

**Description:** returns the mac from a selected clientname

**Parameter:**

- clientName: name of the client

### 8.22.55 CLIENT\_sshFetchJob

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

**Parameter:**

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

### 8.22.56 CLIENT\_backToRed

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

**Parameter:**

- clientName: name of the client

### 8.22.57 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 removed to a m23normalRemove job.
- addShutdownOrRebootPackage: If set to true, a shutdown or reboot package will be added.

### 8.22.58 CLIENT\_recoverClientCombineJobs

**Description:** Recover a client: all client jobs are done again, status is set to 0, all identical jobs are combined.

**Parameter:**

- clientName: name of the client

### 8.22.59 CLIENT\_wol

**Description:** wakes a client over the network

**Parameter:**

- clientName: name of the client

### 8.22.60 CLIENT\_recalculateStatusBar

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

**Parameter:**

- clientName: name of the client

### 8.22.61 CLIENT\_resetStatusBar

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

**Parameter:**

- clientName: name of the client

### 8.22.62 CLIENT\_startInstall

**Description:** starts the installation on a client

**Parameter:**

- clientName: name of the client

### 8.22.63 CLIENT\_resetAndInstall

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

**Parameter:**

- clientName: name of the client

### 8.22.64 CLIENT\_getBootType

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

**Parameter:**

- clientName: name of the client

### 8.22.65 CLIENT\_isrunning

**Description:** tests out whether a client is up (running) or not

**Parameter:**

- clientName: name of the client

### 8.22.66 CLIENT\_reset

**Description:** resets a client

**Parameter:**

- clientName: name of the client

### 8.22.67 CLIENT\_showLastLogError

**Description:** Prints the last error in the client log

**Parameter:**

- clientName: name of the client

### 8.22.68 CLIENT\_showLog

**Description:** prints the log information of the client

**Parameter:**

- clientName: name of the client

### 8.22.69 CLIENT\_getClientName

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

### 8.22.70 CLIENT\_getAllOptions

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

**Parameter:**

- clientName: name of the client

### 8.22.71 CLIENT\_setAllOptions

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

**Parameter:**

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

### 8.22.72 CLIENT\_getAllAskingOptions

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

### 8.22.73 CLIENT\_getSetOption

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

**Parameter:**

- options: name of the options array

### 8.22.74 CLIENT\_options2HiddenForm

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

**Parameter:**

- options: name of the options array

### 8.22.75 CLIENT\_hiddenForm2options

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

**Parameter:**

- options: name of the options array

### 8.22.76 CLIENT\_getStatusimage

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

**Parameter:**

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

### 8.22.77 CLIENT\_showStatusSelection

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

**Parameter:**

- client: the name of the client

### 8.22.78 CLIENT\_listCriticalClients

**Description:** lists clients with critical status'

### 8.22.79 CLIENT\_isInDebugMode

**Description:** returns "true", if a client is in debug mode

**Parameter:**

- clientName: name of the client

### 8.22.80 CLIENT\_toggleDebugMode

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

**Parameter:**

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

### 8.22.81 CLIENT\_getStatusimage

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

**Parameter:**

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

### 8.22.82 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.

### 8.22.83 CLIENT\_showDebugSelection

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

**Parameter:**

- client: the name of the client

### 8.22.84 CLIENT\_isInRescueMode

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

**Parameter:**

- clientName: the name of the client

### 8.22.85 CLIENT\_showDirectConnectionHelp

**Description:** returns the help file for directConnection and replaces place holders with the correct values

**Parameter:**

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

### 8.22.86 CLIENT\_isInDebugMode

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

### 8.22.87 CLIENT\_getToDetailsURL

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

**Parameter:**

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

**Returns:**

- Link to the client's control center page

### 8.22.88 CLIENT\_HTMLBackToDetails

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

**Parameter:**

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

### 8.22.89 CLIENT\_getId

**Description:** returns the id of a client

**Parameter:**

- clientName: the name of the client

### 8.22.90 CLIENT\_query

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

**Parameter:**

- o1: operator 1 (can be '=', '<', '>') selects of the first state should be equal, smaler or bigger that the state in s1
- s1: first state to compare with the state of the client
- o2: operator 2
- s2: second state to compare

- groupName: if you want to filter for special group, set it to the group name
- o3: operator 3
- s3: third state to compare
- search: Search string to search all clients for and only list matching clients or all if \$search is empty.

### 8.22.91 CLIENT\_addChangeElement

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

**Parameter:**

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

### 8.22.92 CLIENT\_showDelDialog

**Description:** Shows the dialog for deleting a client.

### 8.22.93 CLIENT\_showAddDialog

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

**Parameter:**

- addType: defines the behaviour and appearance of the dialog

### 8.22.94 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 be shown
- deleteVM: Set to true to delete the VM too.

### 8.22.95 CLIENT\_getNames

**Description:** returns an array with all clients

**Parameter:**

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

### 8.22.96 CLIENT\_getNamesWithPackages

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

**Parameter:**

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

### 8.22.97 CLIENT\_changeClient

**Description:** changes values of the clients



**8.22.98 CLIENT\_setAllParams**

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

**Parameter:**

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

**8.22.99 CLIENT\_plinkFetchJob**

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

**Parameter:**

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

**8.22.100 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 normally installed.

**8.22.101 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.

**8.22.102 CLIENT\_generateHTMLDedicatedAndReachableStatus**

**Description:** Generates HTML code and tooltip containing the status of the client showing if it's dedicated and reachable.

**Parameter:**

- online: true, if client is online

**Returns:**

- Associative array containing HTML code and tooltip

**8.22.103 CLIENT\_sendPlymouthMessage**

**Description:** If Plymouth is running, display a message

**Parameter:**

- message: message to display

### 8.22.104 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 disappears. 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

## 8.23 *./inc/Cm23AdminLister.php*

Author: Maren Hachmann (maren@goos-habermann.de), Hauke Goos-Habermann (HHabermann@pc-kiel.de)

Description: Class for listing m23 administrators.

### 8.23.1 *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

### 8.23.2 *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

### 8.23.3 *Cm23AdminLister::CountAdmins*

**Description:** counts the number of registered admins in m23 password file

**Returns:**

- : number of admins in m23 password file

### 8.23.4 *Cm23AdminLister::ListAdmins*

**Description:** lists all admins in m23 password file

**Returns:**

- : array of admin names in m23 password file

## 8.24 ./inc/Cm23Admin.php

Author: Maren Hachmann (maren@goos-habermann.de), Hauke Goos-Habermann (HHabermann@pc-kiel.de)

Description: Class for managing m23 administrators.

### 8.24.1 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).

### 8.24.2 Cm23Admin::\_\_destruct

**Description:** Destructor for a Cm23Admin object. Prints out messages and saves changed options to the database.

### 8.24.3 Cm23Admin::delete

**Description:** Deletes an m23 administrator from the database and from the password hash files for m23, backup, backuppc and phpmyslap.

**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).

### 8.24.4 Cm23Admin::changePw

**Description:** Changes the password of an m23 administrator in the password hash files for m23, backup, backuppc and phpmyslap.

**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

### 8.24.5 Cm23Admin::setCSS

**Description:** sets the CSS file

**Parameter:**

- \$css: element indicating chosen CSS, see array \$cssList

### 8.24.6 Cm23Admin::getCSS

**Description:** finds the CSS file for the CSS the user has chosen, defines selection for themeChoice

**Returns:**

- : name of CSS file

### 8.24.7 Cm23Admin::setLanguage

**Description:** sets the language for the admin

**Parameter:**

- \$shortLanguage: indicates chosen language (current valid values: de, en or fr, see I18N\_m23instLanguage(\$shortLanguage))

### 8.24.8 Cm23Admin::getLanguage

**Description:** finds the language the user has chosen

**Returns:**

- : name of language

### 8.24.9 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.

### 8.24.10 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.

### 8.24.11 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.

## 8.25 ./inc/CMessageManager.php

Author: Hauke Goos-Habermann (HHabermann@pc-kiel.de)

Description: Class for managing messages in the m23 webinterface.

### 8.25.1 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 messages should be stored.
- foundVar: The reference to the variable that should be set to true when there are messages in the given stack.

### 8.25.2 CMessageManager::addInfoMessage

**Description:** Adds an info message to the info message stack and sets the info flag.

**Parameter:**

- msg: Info message to add.

### 8.25.3 CMessageManager::addWarningMessage

**Description:** Adds a warning message to the warning message stack and sets the warning flag.

**Parameter:**

- msg: Warning message to add.

### 8.25.4 CMessageManager::addErrorMessage

**Description:** Adds an error message to the error message stack and sets the error flag.

**Parameter:**

- msg: Error message to add.

### 8.25.5 CMessageManager::popInfoMessagesHTML

**Description:** Returns all info messages and deletes the info message stack.

**Returns:**

- msg: The complete info message stack.

### 8.25.6 CMessageManager::popWarningMessagesHTML

**Description:** Returns all warning messages and deletes the warning message stack.

**Returns:**

- msg: The complete warning message stack.

### 8.25.7 CMessageManager::popErrorMessageHTML

**Description:** Returns all error messages and deletes the error message stack.

**Returns:**

- msg: The complete error message stack.

### 8.25.8 CMessageManager::deleteAllMessages

**Description:** Deletes all messages from the message stack.

### 8.25.9 CMessageManager::hasInfos

**Description:** Returns if there are infos.

**Returns:**

- true, if there have infos been occurred otherwise false.

### 8.25.10 CMessageManager::hasWarnings

**Description:** Returns if there are warnings.

**Returns:**

- true, if there have warnings been occurred otherwise false.

### 8.25.11 CMessageManager::hasErrors

**Description:** Returns if there are errors.

**Returns:**

- true, if there have errors been occurred otherwise false.

### 8.25.12 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.

### 8.25.13 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.

### 8.25.14 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.

### 8.25.15 CMessageManager::showMessages

**Description:** Shows all existing messages in the according boxes.

**Returns:**

- true, if there have errors or warnings been occurred otherwise false.

## 8.26 `./inc/CObjectStorageManager.php`

Author: Hauke Goos-Habermann (HHabermann@pc-kiel.de)

Description: Class for storing PHP objects in the database.

### 8.26.1 `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.

### 8.26.2 `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.

### 8.26.3 `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.

### 8.26.4 `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.

### 8.26.5 `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.



### 8.26.6 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.

### 8.26.7 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.

### 8.26.8 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.

### 8.26.9 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.

### 8.26.10 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.

### 8.26.11 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.

### 8.26.12 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.

### 8.26.13 CObjectStorageManager::showList

**Description:** Shows a list of existing objects matching criteria set by getBy\* functions.

## 8.27 *./inc/CObjectStorage.php*

Author: Hauke Goos-Habermann (HHabermann@pc-kiel.de)

Description: Class for storing PHP objects in the database.

### 8.27.1 **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.

### 8.27.2 **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.

### 8.27.3 **CObjectStorage::\_\_destruct**

**Description:** Destructor for storing changed values of the object back to the DB.

### 8.27.4 **CObjectStorage::getObject**

**Description:** Gets the object.

**Returns:**

- object.

### 8.27.5 **CObjectStorage::destroy**

**Description:** Destroys the

**Returns:**

- Human readable status.

## 8.28 `./inc/CPackageStatusCompare.php`

Author: Hauke Goos-Habermann (HHabermann@pc-kiel.de)

Description: Class for comparing the package status between clients and files.

### 8.28.1 `CPackageStatusCompare::__construct`

**Description:** Constructor for new `CPackageStatusCompare` objects.

### 8.28.2 `CPackageStatusCompare::__destruct`

**Description:** Destructor for a `CPackageStatusCompare` object.

### 8.28.3 `CPackageStatusCompare::save`

**Description:** Saves parameters as hidden POST values.

### 8.28.4 `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.

### 8.28.5 `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.

### 8.28.6 `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.

### 8.28.7 `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

### 8.28.8 CPackageStatusCompare::setClient1

**Description:** Sets a client, if the given parameter is a valid client name.

**Parameter:**

- cl: Name of the client

### 8.28.9 CPackageStatusCompare::setClient2

**Description:** Sets a client, if the given parameter is a valid client name.

**Parameter:**

- cl: Name of the client

### 8.28.10 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.

### 8.28.11 CPackageStatusCompare::isClient1Set

**Description:** Checks, if the 1st client is set.

**Returns:**

- true, if client name is set.

### 8.28.12 CPackageStatusCompare::isClient2Set

**Description:** Checks, if the 2nd client is set.

**Returns:**

- true, if client name is set.

### 8.28.13 CPackageStatusCompare::isFile1Set

**Description:** Checks, if the 1st combined variable stores a file name.

**Returns:**

- true, if the variable stores a file name, otherwise false.

### 8.28.14 CPackageStatusCompare::isFile2Set

**Description:** Checks, if the 2st combined variable stores a file name.

**Returns:**

- true, if the variable stores a file name, otherwise false.

### 8.28.15 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.

### 8.28.16 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.

### 8.28.17 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.

### 8.28.18 CPackageStatusCompare::diff

**Description:** Compares two package status files.

### 8.28.19 CPackageStatusCompare::show

**Description:** Shows the comparing dialog.

## 8.29 *./inc/CPoolFromClientDebsGUI.php*

Author: Hauke Goos-Habermann (HHabermann@pc-kiel.de)

Description: GUI class for building pools from client packages.

### 8.29.1 *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.

### 8.29.2 *CPoolFromClientDebsGUI::addm23BuildPoolFromClientDebsJob*

**Description:** Adds a m23BuildPoolFromClientDebs job to the client.

**Parameter:**

- `clientName`: Name of the client, the pool should be build from.

### 8.29.3 *CPoolFromClientDebsGUI::DEFINE\_checkboxForAddingm23BuildPoolFromClientDebsJob*

**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.

## 8.30 `./inc/CPoolFromClientGUI.php`

Author: Hauke Goos-Habermann (HHabermann@pc-kiel.de)

Description: GUI class for building pools from client packages.

### 8.30.1 `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.

### 8.30.2 `CPoolFromClientGUI::setImportPoolName`

**Description:** Sets the name of this pool by the client name.

**Parameter:**

- `clientName`: Name of the client, the pool is build from.

### 8.30.3 `CPoolFromClientGUI::getImportPoolName`

**Description:** Returns the name of this pool.

**Returns:**

- The name of this pool.

### 8.30.4 `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.

### 8.30.5 `CPoolFromClientGUI::getCOSStatusHumanReadable`

**Description:** Translates the status code (`COSSTATUS_*`) into human readable word(s).

**Returns:**

- Human readable word(s) representing the status code.

### 8.30.6 `CPoolFromClientGUI::getCOSStatus`

**Description:** Gets the status code (`COSSTATUS_*`).

**Returns:**

- Status code of this object.

### 8.30.7 `CPoolFromClientGUI::runCOSLoop`

**Description:** Function that is called on every run of `CObjectStorageManager::getAllObjectsByRes`.

### 8.30.8 `CPoolFromClientGUI::saveInObjectStorage`

**Description:** Saves this object in the object storage.



## 8.31 *./inc/CPoolGUI.php*

Author: Hauke Goos-Habermann (HHabermann@pc-kiel.de)

Description: GUI class for building pools.

### 8.31.1 *CPoolGUI::show*

**Description:** Shows the start dialog for creating, adding, deleting and changing pools.

### 8.31.2 *CPoolGUI::getHelpPage*

**Description:** Returns the current help page.

**Returns:**

- Current help page.

### 8.31.3 *CPoolGUI::getHeading*

**Description:** Returns the current heading.

**Returns:**

- Current heading.

### 8.31.4 *POOL\_showReadCD*

**Description:** shows a dialog for copying the CD contents to the pool

**Parameter:**

- poolName: name of the pool

### 8.31.5 *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.

### 8.31.6 *CPoolGUI::DIALOG\_convertPackagesToRepositoryStatus*

**Description:** Shows information (status of the Packages\* generation, sources.list) about the currently generated pool.

### 8.31.7 *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.

### 8.31.8 CPoolGUI::DIALOG\_showDownloadStatus

**Description:** Shows the package download status of a pool.

### 8.31.9 CPoolGUI::DIALOG\_start

**Description:** Shows the start dialog for creating, adding, deleting and changing pools.

### 8.31.10 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.

### 8.31.11 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.

### 8.31.12 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.

### 8.31.13 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.

### 8.31.14 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.

## **8.32   *./inc/CPoolLister.php***

Author: Hauke Goos-Habermann (HHabermann@pc-kiel.de)

Description: Class for listing pools.

## 8.33 ./inc/CPool.php

Author: Hauke Goos-Habermann (HHabermann@pc-kiel.de)

Description: GUI class for building pools.

### 8.33.1 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).

### 8.33.2 CPool::signRelease

**Description:** Signs the Release file as Release.gpg and InRelease.

**Returns:**

- : Name the log file.

### 8.33.3 CPool::getConvertPackagesToRepositoryLogName

**Description:** Returns the full file name of the convert packages to repository log file.

**Returns:**

- : Name the log file.

### 8.33.4 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.

### 8.33.5 CPool::isConvertPackagesToRepositoryRunning

**Description:** Checks if the conversation of downloaded packages to a repository is running.

**Returns:**

- : true, if it is running, otherwise false.

### 8.33.6 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.

### 8.33.7 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 separate screen sessions.
- 

**Returns:**

- : true, if the task is running, otherwise false.

### 8.33.8 CPool::isDownloadRunning

**Description:** Checks if a download of packages to the pool is running.

**Returns:**

- : true, if download is running, otherwise false.

### 8.33.9 CPool::resetDownloadLog

**Description:** Deletes the aptDownload.log file and resets the line number of the last read line.

### 8.33.10 CPool::getDownloadLogNewLines

**Description:** Gets the last (new) lines of the (growing) download log file.

**Returns:**

- : UTF8-encoded new lines of the log file.

### 8.33.11 CPool::getDownloadLogContents

**Description:** Gets the contents of the download log file.

**Returns:**

- : Contents of the download log file.

### 8.33.12 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.

### 8.33.13 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.

### 8.33.14 CPool::hasPoolDownloadBasePackages

**Description:** Checks, if base packages should be downloaded.

**Returns:**

- : true, if base packages should be downloaded otherwise false.

### 8.33.15 CPool::setPoolDownloadBasePackages

**Description:** Sets, if base packages should be downloaded.

**Parameter:**

- downloadBasePackages: true, when base packages should be downloaded otherwise false.

### 8.33.16 CPool::getPoolImportedPackageList

**Description:** Gets the list of packages that were downloaded (or have to be downloaded) into the pool.

**Returns:**

- : PackageList of the pool.

### 8.33.17 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.

### 8.33.18 CPool::getPoolDistribution

**Description:** Gets the distribution value of the pool.

**Returns:**

- : Distribution of the pool.

### 8.33.19 CPool::setPoolDistribution

**Description:** Sets the distribution value of the pool.

**Parameter:**

- distribution: Distribution of the pool.

### 8.33.20 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).

### 8.33.21 CPool::getPoolDir

**Description:** Gets the directory of the pool.

**Returns:**

- : Directory of the pool.

### 8.33.22 CPool::setPoolName

**Description:** Sets the name of the pool and create the pool directory.

**Parameter:**

- poolName: Name of the pool.

### 8.33.23 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.

### 8.33.24 CPool::getPoolSourceslist

**Description:** Gets the sourceslist value of the pool.

**Returns:**

- : Sourceslist of the pool.

### 8.33.25 CPool::setPoolSourceslist

**Description:** Sets the sourceslist value of the pool.

**Parameter:**

- sourceslist: Sourceslist of the pool.

### 8.33.26 CPool::getPoolDescription

**Description:** Gets the description value of the pool.

**Returns:**

- : Description of the pool.

### 8.33.27 CPool::setPoolDescription

**Description:** Sets the description value of the pool.

**Parameter:**

- description: Description of the pool.

### 8.33.28 CPool::getPoolRelease

**Description:** Gets the release value of the pool.

**Returns:**

- : Release of the pool.

### 8.33.29 CPool::setPoolRelease

**Description:** Sets the release value of the pool.

**Parameter:**

- release: Release of the pool.

### 8.33.30 CPool::getPoolType

**Description:** Gets the type value of the pool.

**Returns:**

- : Type of the pool.

### 8.33.31 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).

### 8.33.32 CPool::getPoolArch

**Description:** Gets the architecture value of the pool.

**Returns:**

- : Architecture of the pool.

### 8.33.33 CPool::setPoolArch

**Description:** Sets the architecture value of the pool.

**Parameter:**

- arch: Architecture of the pool (POOL\_ARCH\_I386 or CPoolLister::POOL\_ARCH\_AMD64).

### 8.33.34 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

### 8.33.35 CPool::getProperty

**Description:** Reads the contents of a property file.

**Parameter:**

- property: Name of the pool property

**Returns:**

- Contents of a property file

### 8.33.36 CPool::getPoolSize

**Description:** Calculates the disk usage of a pool.

**Returns:**

- Size of the pool in MB

### 8.33.37 CPool::destroyPool

**Description:** Deletes the pool

**Parameter:**

- poolName: Name of the pool (can optionally be set here)



### 8.33.38 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.

### 8.33.39 CPool::preparePool

**Description:** Prepares the pool to make it able to store software packages.

**Returns:**

- false on errors, true on success.

### 8.33.40 CPool::stopDownloadToPool

**Description:** Stops the download of packages to the pool.

### 8.33.41 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.

### 8.33.42 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.

### 8.33.43 CPool::getPoolStatusFileName

**Description:** Returns the name of the pool status file.

**Returns:**

- Name of the pool status file.

### 8.33.44 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.

### 8.33.45 CPool::getPoolStatusFileContents

**Description:** Gets the contents of the pool status file.

**Returns:**

- Contents of the pool status file.

### 8.33.46 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)

## 8.34 *./inc/cron.php*

Author: Hauke Goos-Habermann (HHabermann@pc-kiel.de)

Description: Functions for managing cron jobs.

### 8.34.1 **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 execute.
- 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.

### 8.34.2 **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 execute.
- identifier: A string to identify the cron entry (for deletion)
- runInScreen: Set to true if the command should be run in a screen.

### 8.34.3 **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 execute.
- identifier: A string to identify the cron entry (for deletion)
- runInScreen: Set to true if the command should be run in a screen.

#### 8.34.4 CRON\_rmJob

**Description:** Removes an entry from the crontab.

**Parameter:**

- identifier: A string to identify the cron entry (for deletion)

#### 8.34.5 CRON\_reloadConfig

**Description:** Reloads cron with new crontab.

#### 8.34.6 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 execute.
- identifier: A string to identify the cron entry (for deletion)
- runInScreen: Set to true if the command should be run in a screen.

#### 8.34.7 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 execute.
- identifier: A string to identify the cron entry (for deletion)
- runInScreen: Set to true if the command should be run in a screen.

#### 8.34.8 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 execute.
- identifier: A string to identify the cron entry (for deletion)
- runInScreen: Set to true if the command should be run in a screen.

### 8.34.9 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.

### 8.34.10 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.

### 8.34.11 CRON\_checkMinute

**Description:** Checks if a minute value is valid.

**Parameter:**

- min: Minute value to check.

**Returns:**

- True if it is valid otherwise false.

### 8.34.12 CRON\_checkHour

**Description:** Checks if a hour value is valid.

**Parameter:**

- hour: Hour value to check.

**Returns:**

- True if it is valid otherwise false.

### 8.34.13 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.

#### 8.34.14 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.

#### 8.34.15 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 attached.

#### 8.34.16 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 execute.
- identifier: A string to identify the cron entry (for deletion)
- runInScreen: Set to true if the command should be run in a screen.

#### 8.34.17 CRON\_cronEntryDeletionDialog

**Description:** Shows a list of crontab entries matching the identifier with deletion option.

**Parameter:**

- identifier: A string to identify the cron entry (for deletion)

#### 8.34.18 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.

## 8.35 *./inc/CScredit.php*

Author: Hauke Goos-Habermann (HHabermann@pc-kiel.de)

Description: Functions for the embedded script editor.

### 8.35.1 *CScredit::\_\_construct*

**Description:** Constructor for new CScredit objects.

### 8.35.2 *CScredit::setCurrentScriptFilename*

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

**Parameter:**

- filename: The filename to check and correct.

### 8.35.3 *CScredit::getCurrentScriptFilename*

**Description:** Gets the current script filename.

**Returns:**

- Current script filename.

### 8.35.4 *CScredit::getCurrentScriptFilenameFullPath*

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

**Returns:**

- Current script filename full path.

### 8.35.5 *CScredit::getCurrentScriptFilenameWithoutInstallPhp*

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

**Returns:**

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

### 8.35.6 *CScredit::getLocalScriptFileNames*

**Description:** Gets the filenames of local scripts.

**Returns:**

- Array with the filenames of local scripts.

### 8.35.7 *CScredit::updateOnlineScriptInfo*

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

**Returns:**

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

### 8.35.8 CScredit::getNextOnlineScriptInfo

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

**Parameter:**

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

**Returns:**

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

### 8.35.9 CScredit::loadOnlineScript

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

**Parameter:**

- scriptIdToLoad: Id of the script to load.

### 8.35.10 CScredit::getOnlineScriptDialog

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

**Returns:**

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

### 8.35.11 CScredit::isNotSaved

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

**Returns:**

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

### 8.35.12 CScredit::getNewScriptTemplate

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

**Returns:**

- Text of the script template.

### 8.35.13 CScredit::uploadScript

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

**Parameter:**

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



#### 8.35.14 CScredit::getCurrentScript

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

**Returns:**

- Current text of the editor window.

#### 8.35.15 CScredit::setCurrentScript

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

**Parameter:**

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

#### 8.35.16 CScredit::deleteCurrentScript

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

**Returns:**

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

#### 8.35.17 CScredit::getViewScriptOutputDialog

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

**Returns:**

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

#### 8.35.18 CScredit::saveScript

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

#### 8.35.19 CScredit::show

**Description:** Shows a script editor with syntax highlighting.

## 8.36 ./inc/CSystemProxy.php

Author: Hauke Goos-Habermann (HHabermann@pc-kiel.de)

Description: Class for managing systemwide proxies.

### 8.36.1 CSYSTEMPROXY\_getProxySettingsFromAPT

**Description:** Parses the (maybe existing) proxy settings from the local APT config file.

**Returns:**

- Array with the proxy settings.

### 8.36.2 CSYSTEMPROXY\_addCurlProxySettings

**Description:** Sets proxy settings for curl, if a proxy is active.

### 8.36.3 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 authentication is used.

### 8.36.4 CSYSTEMPROXY\_getEnvironmentVariables

**Description:** Generates BASH proxy variables.

**Parameter:**

- getAlways: If set to true, the variables will always be returned.

**Returns:**

- BASH proxy variables.

### 8.36.5 CSYSTEMPROXY\_getAptGetProxyParamter

**Description:** Generates the apt-get proxy parameters.

**Returns:**

- : apt-get proxy parameters, if the system proxy is active otherwise empty string.

### 8.36.6 CSystemProxy::\_\_construct

**Description:** Constructor for new CSystemProxy objects.

### 8.36.7 CSystemProxy::writeEtcProfiles

**Description:** Writes the proxy settings to the environment file.

**8.36.8 CSystemProxy::writeAPTProxyConf**

**Description:** Writes the proxy settings to the APT configuration file or comments them out.

**8.36.9 CSystemProxy::writeSquidConf**

**Description:** Writes the (parent) proxy settings to the Squid configuration or removed them.

**8.36.10 CSystemProxy::save**

**Description:** Saves the proxy settings in all configuration files.

**8.36.11 CSystemProxy::showProxyDialog**

**Description:** Shows a dialog for editing the the proxy settings.

**8.36.12 CSystemProxy::loadProxySettings**

**Description:** Loads the proxy settings (from APT config file).

**8.36.13 CSystemProxy::isProxyActive**

**Description:** Checks, if a proxy is actively used.

**Returns:**

- true, if a proxy is actively used, otherwise false.

**8.36.14 CSystemProxy::setProxyActive**

**Description:** Activated or deactivates a proxy.

**Parameter:**

- active: Set to true to activate or false to deactivate proxy usage.

**8.36.15 CSystemProxy::areProxySettingsAvailable**

**Description:** Checks, if proxy settings are available.

**Returns:**

- true, if proxy settings are available, otherwise false.

**8.36.16 CSystemProxy::setProxyHost**

**Description:** Sets the proxy host/servername.

**Parameter:**

- hos: Proxy host/servername.

**8.36.17 CSystemProxy::getProxyHost**

**Description:** Gets the proxy host/servername.

**Returns:**

- Proxy host/servername.

### 8.36.18 CSystemProxy::getProxyHostIP

**Description:** Gets the proxy IP.

**Returns:**

- Proxy IP or false, if the proxy IP could not be detected.

### 8.36.19 CSystemProxy::usesUserPassword

**Description:** Checks, if the proxy uses user/password authentication.

**Returns:**

- true, if the proxy uses user/password authentication, otherwise false.

### 8.36.20 CSystemProxy::setProxyUser

**Description:** Sets the proxy authentication username.

**Parameter:**

- user: The username for proxy authentication.

### 8.36.21 CSystemProxy::getProxyUser

**Description:** Gets the proxy authentication username.

**Returns:**

- Proxy authentication username.

### 8.36.22 CSystemProxy::setProxyPassword

**Description:** Sets the proxy authentication password.

**Parameter:**

- pass: Proxy authentication password.

### 8.36.23 CSystemProxy::getProxyPassword

**Description:** Gets the proxy authentication password.

**Returns:**

- Proxy authentication password.

### 8.36.24 CSystemProxy::setProxyScheme

**Description:** Sets the proxy scheme (http/ftp).

**Parameter:**

- scheme: Proxy scheme (http/ftp)

### 8.36.25 CSystemProxy::getProxyScheme

**Description:** Gets the proxy scheme (http/ftp).

**Returns:**

- Proxy scheme (http/ftp)

### 8.36.26 CSystemProxy::setProxyPort

**Description:** Sets the proxy port.

**Parameter:**

- port: Proxy port.

**8.36.27 CSystemProxy::getProxyPort**

**Description:** Gets the proxy port.

**Returns:**

- Proxy port.

**8.36.28 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 authentication is used.

## 8.37 ./inc/db.php

Author: Daniel Kasten (DKasten@pc-kiel.de) ,Hauke Goos-Habermann (HHabermann@pc-kiel.de)

Description: database functions, open, close the database, get ip of the calling client

### 8.37.1 ip2longSafe

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

**Parameter:**

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

**Returns:**

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

### 8.37.2 getArchList

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

**Returns:**

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

### 8.37.3 isMySQL3used

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

### 8.37.4 encryptShadow

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

**Parameter:**

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

### 8.37.5 getClientLanguage

### 8.37.6 getInstDev

**Description:** fetch the device for installation

**Parameter:**

- id: package ID

### 8.37.7 DB\_getConnection

**Description:** Gets the MySQLi connection.

**Returns:**

- MySQLi connection.

### 8.37.8 DB\_setConnection

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

**Parameter:**

- conn: MySQLi connection to use globally.

### 8.37.9 DB\_isConnectionValid

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

**Returns:**

- true on valid MySQLi connection, otherwise false.

### 8.37.10 dbConnect

**Description:** connects to the m23 database

### 8.37.11 dbClose

**Description:** closes the connection to the m23 database

### 8.37.12 DB\_getSuperUserName

**Description:** Returns the name of the super MySQL/MariaDB user.

**Returns:**

- Name of the super MySQL/MariaDB user.

### 8.37.13 DB\_getSuperUserPassword

**Description:** Returns the password (or empty) of the super MySQL/MariaDB user.

**Returns:**

- Password (or empty) of the super MySQL/MariaDB user.

### 8.37.14 getServerIP

**Description:** returns the IP of the m23 server

**Returns:**

- IP of the m23 server.

### 8.37.15 getServerNetmask

**Description:** returns the netmask of the m23 server

### 8.37.16 getServerNetwork

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

**Returns:**

- Network IP of the m23 server.

### 8.37.17 getDNSServers

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

### 8.37.18 getServerGateway

**Description:** Returns the gateway of the m23 server

### 8.37.19 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

### 8.37.20 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 formatted, base system is installed

### 8.37.21 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 formatted, base system is installed

### 8.37.22 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

### 8.37.23 deleteClientLogs

**Description:** deletes the installation logs

**Parameter:**

- clientName: name of the client to delete all logs

### 8.37.24 workPhpName

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

**Returns:**

- Unique name for storing the work.php file.

### 8.37.25 executeNextWork

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

### 8.37.26 DB\_query

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

**Parameter:**

- sql: sql query



### 8.37.27 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.

### 8.37.28 DB\_genPassword

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

**Parameter:**

- length: length of password

### 8.37.29 getClientIP

**Description:** returns the IP of the calling client

### 8.37.30 implodeAssoc

**Description:** makes a string from an associative array

**Parameter:**

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

### 8.37.31 explodeAssoc

**Description:** makes an associative array from a string

**Parameter:**

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

### 8.37.32 sedSearchReplace

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

**Parameter:**

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

### 8.37.33 isProgrammInstalled

**Description:** returns true if a programm can be used

**Parameter:**

- progName: name of the programm

### 8.37.34 pingIP

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

**Parameter:**

- ip

### 8.37.35 delFromArray

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

**Parameter:**

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

### 8.37.36 delValuesFromArray

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

**Parameter:**

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

**Returns:**

- array without the entries in \$delVals.

### 8.37.37 DB\_getLikeableColumns

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

**Parameter:**

- table: Name of the table to search.

**Returns:**

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

### 8.37.38 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.

### 8.37.39 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).

### 8.37.40 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.

## 8.38 *./inc/dhcp.php*

Author: Daniel Kasten (DKasten@pc-kiel.de), Hauke Goos-Habermann (HHabermann@pc-kiel.de)

Description: functions to add and remove a client to/from the dhcp server list.

### 8.38.1 *DHCP\_exportDHCPSettingsForExternalDHCPServer*

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

### 8.38.2 *DHCP\_lineNumberAffterLastClient*

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

**Returns:**

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

### 8.38.3 *DHCP\_addSubnetDefinition*

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

**Parameter:**

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

### 8.38.4 *DHCP\_delSubnetDefinition*

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

**Parameter:**

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

### 8.38.5 *DHCP\_addDynamicRange*

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

**Parameter:**

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

**Returns:**

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

### 8.38.6 *DHCP\_getDynamicRanges*

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

**Returns:**

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

### 8.38.7 DHCP\_getNetmaskOfDynamicRanges

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

**Parameter:**

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

**Returns:**

- Netmask or false in case of an error.

### 8.38.8 DHCP\_delDynamicRange

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

**Parameter:**

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

**Returns:**

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

### 8.38.9 DHCP\_bootTypeToNewFormat

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

**Parameter:**

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

**Returns:**

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

### 8.38.10 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 adresse of the network card
- bootType: Parameter can a string: pxe, etherboot, gpxe, none
- gateway: The gateway for the client.

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

**Returns:**

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

### 8.38.12 DHCP\_addLineToDHCPDConf

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

**Parameter:**

- line: Line to add.

### 8.38.13 DHCP\_restartDHCPserver

**Description:** Restarts the DHCP server.

**Returns:**

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

### 8.38.14 DHCP\_rmClient

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

**Parameter:**

- clientName: name of the client

### 8.38.15 DHCP\_setBootimage

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

**Parameter:**

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

### 8.38.16 DHCP\_activateBoot

**Description:** switches the network boot on or off

**Parameter:**

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

### 8.38.17 DHCP\_calcPXEIP

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

**Parameter:**

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

### 8.38.18 DHCP\_writePXEcfg

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

**Parameter:**

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

### 8.38.19 DHCP\_removePXEcfg

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

**Parameter:**

- clientName: name of the client

### 8.38.20 DHCP\_isNetworkBoottingActive

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

**Parameter:**

- clientName: name of the client

**Returns:**

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

## 8.39 `./inc/distr/debian/clientConfigCommon.php`

Author: Hauke Goos-Habermann (HHabermann@pc-kiel.de)

Description: Common management funtions shared by Linux distributions.

### 8.39.1 `CLCFG_setSysTime`

**Description:** Sets the system time of the client to about the same time of the m23 server.

### 8.39.2 `CLCFG_disableAptSystemdDaily`

**Description:** Deactivates systemd apt-daily.timer.

### 8.39.3 `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.

### 8.39.4 `CLCFG_disableSudoRootLogin`

**Description:** Disables getting root rights of normal users via sudo.

### 8.39.5 `CLCFG_installFirmware`

**Description:** Installs available firmware packages.

### 8.39.6 `CLCFG_monoRemove`

**Description:** Removes mono packages and installs gnote when tomboy was installed before.

### 8.39.7 `CLCFG_setDebConfDM`

**Description:** Sets the display manager in debconf.

**Parameter:**

- `dm`: Name of the display manager (kdm, lightdm, mdm, ...).

### 8.39.8 `CLCFG_installXDM`

**Description:** Installs the XDM display manager.

### 8.39.9 `CLCFG_installSDDM`

**Description:** Installs the SDDM display manager.

### 8.39.10 CLCFG\_installMintDM

**Description:** Installs the Linux Mint DM display manager.

### 8.39.11 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).

### 8.39.12 CLCFG\_activateBOOT\_DEGRADED

**Description:** Activates BOOT\_DEGRADED on Ubuntu 12.04 to allow booting from degraded RAIDs.

### 8.39.13 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.

### 8.39.14 CLCFG\_setDebConfDirect

**Description:** Sets debconf settings.

**Parameter:**

- debconf: Debconf settings to add.

### 8.39.15 TRINITY\_installLoginManager

**Description:** Installs the Trinity login manager KDM.

**Parameter:**

- lang: short language

### 8.39.16 TRINITY\_install

**Description:** Installs a minimalistic Trinity desktop.

**Parameter:**

- lang: short language

### 8.39.17 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.

### 8.39.18 CLCFG\_disablePlymouth

**Description:** Disables the plymouth.



### 8.39.19 CLCFG\_disableAvahiDaemon

**Description:** Disables the avahi-daemon.

### 8.39.20 CLCFG\_configUpstartForNormalUsage

**Description:** Configures upstart for normal running in an installed system.

### 8.39.21 CLCFG\_configUpstartForChroot

**Description:** Configures upstart to make it not fail installation.

### 8.39.22 CLCFG\_createScreenRC

**Description:** Creates the (under Ubuntu) needed settings for screen.

### 8.39.23 CLCFG\_addGrubPassword

**Description:** Adds a password line to the grub configuration to lock the edit line of bootmanager.

### 8.39.24 CLCFG\_addLiloPassword

**Description:** Adds a password line to the LiLo configuration to lock the edit line of bootmanager.

### 8.39.25 GNOME\_prepare

**Description:** Prepares the GNOME installation

### 8.39.26 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

### 8.39.27 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

### 8.39.28 GNOME\_install

**Description:** Installs a minimalistic GNOME desktop.

**Parameter:**

- lang: short language

### 8.39.29 GNOME\_installLoginManager

**Description:** Installs the GNOME login manager GDM.

**Parameter:**

- lang: short language

### 8.39.30 KDE\_prepare

**Description:** Prepares the KDE installation

### 8.39.31 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)

### 8.39.32 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.

### 8.39.33 CLCFG\_lilo2Grub

**Description:** Installs and runs the lilo.conf to Grub's menu.lst converter

### 8.39.34 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.

### 8.39.35 CLCFG\_setTimeZone

**Description:** Sets the timezone a a client.

**Parameter:**

- timezone: POSIX timezone as defined at /usr/share/zoneinfo/posix/

### 8.39.36 CLCFG\_writeHosts

**Description:** writes the /etc/hosts file for the client

### 8.39.37 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 unencrypted 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.

### 8.39.38 CLCFG\_createBootDeviceNode

**Description:** Makes sure the needed device nodes for the boot device are created.

### 8.39.39 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.

### 8.39.40 CLCFG\_genFstab

**Description:** generates the commands to auto detect the partitions and generate the fstab file

**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)

### 8.39.41 CLCFG\_interfaces

**Description:** generates a script that writes the /etc/network/interfaces file.

**Parameter:**

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

### 8.39.42 CLCFG\_hostname

**Description:** generates a script that writes the /etc/hostname file.

**Parameter:**

- clientName: the name of the client

### 8.39.43 CLCFG\_resolvConf

**Description:** generates a script that writes the /etc/resolv.conf file.

**Parameter:**

- DNSServers: list of DNS servers

### 8.39.44 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

### 8.39.45 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

### 8.39.46 CLCFG\_sourceslistCreateConfigFiles

**Description:** Creates config files for the package manager.

**Parameter:**

- sourceName: The name of the package source list

### 8.39.47 CLCFG\_hwdetect

**Description:** updates and runs the hardware detection

### 8.39.48 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).

### 8.39.49 CLCFG\_setRootPassword

**Description:** sets the root password of the client

**Parameter:**

- cryptedPassword: sets an already crypted root password

### 8.39.50 `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

### 8.39.51 `CLCFG_copyClientPackageStatus`

**Description:** copies the package installation status file from the client to the cache directory of the server

**Parameter:**

- clientName: the name of the client

### 8.39.52 `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 packages one by one.

**Parameter:**

- command: install or remove
- packages: white space separated list of packages to install or remove

### 8.39.53 `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.

### 8.39.54 `CLCFG_installBasePackages`

**Description:** installs needed base packages

**Parameter:**

- packagelist: the list of the packages to install
- keyring: name of the keyring package

### 8.39.55 `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

### 8.39.56 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

### 8.39.57 CLCFG\_downgradeExt

**Description:** downgrades all ext\* partitions so the fsck.ext\* of woody can understand it

### 8.39.58 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).

### 8.39.59 CLCFG\_activateDMA

**Description:** try to activate DMA to speed up installation. this short script tries to detect

### 8.39.60 CLCFG\_fetchm23BasicTools

**Description:** fetches the basic m23 tools and fsize

### 8.39.61 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

### 8.39.62 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

### 8.39.63 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

### 8.39.64 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

### 8.39.65 CLCFG\_executeAfterChroot

**Description:** prepares for pivot\_root and does it and executes afterChroot

### 8.39.66 CLCFG\_writeCrontabm23fetchjobEvery5Minutes

**Description:** Adds entries to crontab to check every 5 minutes for new jobs.

### 8.39.67 CLCFG\_writeM23fetchjob

**Description:** generates the m23fetchjob script

**Parameter:**

- release: Name of the distribution release for special handling of some releases.

### 8.39.68 CLCFG\_hideKernelWarnings

**Description:** hides the kernel warnings

### 8.39.69 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.

### 8.39.70 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

### 8.39.71 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.

### 8.39.72 CLCFG\_changeUser

**Description:** changes the settings of an useraccount on a client

**Parameter:**

- userName: the (old) username for the account
- password: the new unencrypted password for the account
- newUserName: the new username



### 8.39.73 CLCFG\_patchNsswitchForLDAP

**Description:** Patches /etc/nsswitch.conf for usage with LDAP.

### 8.39.74 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)

### 8.39.75 CLCFG\_disableNFSHome

**Description:** Disables storing of home directories on a NFS server

### 8.39.76 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.

### 8.39.77 CLCFG\_installApplicationLanguagePackages

**Description:** Installs some additional language packages for installed applications with separate language packs.

**Parameter:**

- lang: short language

### 8.39.78 CLCFG\_updateDebootstrapScripts

**Description:** Updates the debootstrap scripts for Debian or Ubuntu and returns the www path to the files

### 8.39.79 CLCFG\_updateDebootstrapScriptsDebian

**Description:** Updates the debootstrap scripts for Debian and returns the www path to the files

## 8.40 `./inc/distr/debian/clientConfig.php`

Author: Hauke Goos-Habermann (HHabermann@pc-kiel.de)

Description: Debian specific management functions.

### 8.40.1 `DEBIAN_desktopInstall`

**Description:** Installs a Debian desktop.

**Parameter:**

- desktop: Desktop constant.

### 8.40.2 `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.

### 8.40.3 `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.

### 8.40.4 `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

### 8.40.5 `CLCFG_addDistributionSpecificOptions`

**Description:** adds distribution specific settings to an option array and returns the new array

**Parameter:**

- \$options: the options array with some values

#### 8.40.6 **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.

#### 8.40.7 **CLIENT\_enableLDAP**

**Description:** enables LDAP login for a client.

**Parameter:**

- clientOptions: the client's options array

## 8.41 `./inc/distr/debian/packages.php`

Author: Hauke Goos-Habermann (HHabermann@pc-kiel.de)

Description: Debian/Ubuntu specific package handling functions.

### 8.41.1 `PKG_updatePackageSearchCacheFile`

**Description:** Updates the compressed package search file.

**Parameter:**

- `packagesource`: Name of the package sources list.

**Returns:**

- Full path to the log file.

### 8.41.2 `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.

### 8.41.3 `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.

### 8.41.4 `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.

### 8.41.5 `PKG_ncTarDebsFromClientToServer_Client`

**Description:** Client to send the Debian packages to the m23 client.

### 8.41.6 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.

### 8.41.7 PKG\_rsncDebsFromClientToServer

**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.

### 8.41.8 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.

### 8.41.9 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.

### 8.41.10 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.

### 8.41.11 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.

### 8.41.12 PKG\_searchFor

**Description:** searches for a package and returns 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.

### 8.41.13 PKG\_getDescription

**Description:** gets the description of a package

**Parameter:**

- line: line containing package name and description

### 8.41.14 PKG\_getPackageName

**Description:** gets the name of a package

**Parameter:**

- line: line containing package name and description

### 8.41.15 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.

### 8.41.16 PKG\_getPackageDescriptionSize

**Description:** Returns 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

### 8.41.17 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

### 8.41.18 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.

### 8.41.19 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.

### 8.41.20 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.

### 8.41.21 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.

### 8.41.22 **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.

### 8.41.23 **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

### 8.41.24 **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.

### 8.41.25 **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).

### 8.41.26 **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)



## 8.42 `./inc/distr/halfSister/clientConfig.php`

Author: Hauke Goos-Habermann (HHabermann@pc-kiel.de)

Description: halfSister specific management functions.

### 8.42.1 `CLCFG_copyClientPackageStatus`

**Description:** Not used, but needs to be defined here.

### 8.42.2 `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.

### 8.42.3 `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.

### 8.42.4 `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

### 8.42.5 `CLCFG_addDistributionSpecificOptions`

**Description:** adds distribution specific settings to an option array and returns the new array

**Parameter:**

- `$options`: the options array with some values

### 8.42.6 `CLCFG_showDistributionSpecificOptions`

**Description:** shows distribution specific options and returns false, if there was an error

**Parameter:**

- `options`: options array

## 8.43 `./inc/distr/halfSister/packages.php`

Author: Hauke Goos-Habermann (HHabermann@pc-kiel.de)

Description: Basic package operations (search, add,...) for halfSister distributions

### 8.43.1 `PKG_updatePackageSearchCacheFile`

**Description:** Updates the compressed package search file.

**Parameter:**

- `packagesource`: Name of the package sources list.

**Returns:**

- The name of the package cache.

### 8.43.2 `PKG_fastGetInstalledPackages`

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

**Parameter:**

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

### 8.43.3 `pkgUpdateCacheOnServer`

**Parameter:**

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

### 8.43.4 `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.

### 8.43.5 PKG\_previewInstall

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

**Parameter:**

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

### 8.43.6 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:**

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

### 8.43.7 PKG\_translateClientPackageStatus

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

**Parameter:**

- status: status code you want to translate

## 8.44 `./inc/distr/imaging/clientConfigCommon.php`

Author: Hauke Goos-Habermann (HHabermann@pc-kiel.de)

Description: Common management functions shared by Linux distributions.

### 8.44.1 `CLCFG_setSystemtime`

**Description:** Sets the system time of the client to about the same time of the m23 server.

### 8.44.2 `CLCFG_disableAptSystemdDaily`

**Description:** Deactivates systemd apt-daily.timer.

### 8.44.3 `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.

### 8.44.4 `CLCFG_disableSudoRootLogin`

**Description:** Disables getting root rights of normal users via sudo.

### 8.44.5 `CLCFG_installFirmware`

**Description:** Installs available firmware packages.

### 8.44.6 `CLCFG_monoRemove`

**Description:** Removes mono packages and installs gnote when tomboy was installed before.

### 8.44.7 `CLCFG_setDebConfDM`

**Description:** Sets the display manager in debconf.

**Parameter:**

- dm: Name of the display manager (kdm, lightdm, mdm, ...).

### 8.44.8 `CLCFG_installXDM`

**Description:** Installs the XDM display manager.

### 8.44.9 `CLCFG_installSDDM`

**Description:** Installs the SDDM display manager.

#### 8.44.10 CLCFG\_installMintDM

**Description:** Installs the Linux Mint DM display manager.

#### 8.44.11 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).

#### 8.44.12 CLCFG\_activateBOOT\_DEGRADED

**Description:** Activates BOOT\_DEGRADED on Ubuntu 12.04 to allow booting from degraded RAIDs.

#### 8.44.13 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.

#### 8.44.14 CLCFG\_setDebConfDirect

**Description:** Sets debconf settings.

**Parameter:**

- debconf: Debconf settings to add.

#### 8.44.15 TRINITY\_installLoginManager

**Description:** Installs the Trinity login manager KDM.

**Parameter:**

- lang: short language

#### 8.44.16 TRINITY\_install

**Description:** Installs a minimalistic Trinity desktop.

**Parameter:**

- lang: short language

#### 8.44.17 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.

#### 8.44.18 CLCFG\_disablePlymouth

**Description:** Disables the plymouth.

#### 8.44.19 CLCFG\_disableAvahiDaemon

**Description:** Disables the avahi-daemon.

#### 8.44.20 CLCFG\_configUpstartForNormalUsage

**Description:** Configures upstart for normal running in an installed system.

#### 8.44.21 CLCFG\_configUpstartForChroot

**Description:** Configures upstart to make it not fail installation.

#### 8.44.22 CLCFG\_createScreenRC

**Description:** Creates the (under Ubuntu) needed settings for screen.

#### 8.44.23 CLCFG\_addGrubPassword

**Description:** Adds a password line to the grub configuration to lock the edit line of bootmanager.

#### 8.44.24 CLCFG\_addLiloPassword

**Description:** Adds a password line to the LiLo configuration to lock the edit line of bootmanager.

#### 8.44.25 GNOME\_prepare

**Description:** Prepares the GNOME installation

#### 8.44.26 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

#### 8.44.27 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

#### 8.44.28 GNOME\_install

**Description:** Installs a minimalistic GNOME desktop.

**Parameter:**

- lang: short language

#### 8.44.29 GNOME\_installLoginManager

**Description:** Installs the GNOME login manager GDM.

**Parameter:**

- lang: short language

#### 8.44.30 KDE\_prepare

**Description:** Prepares the KDE installation

#### 8.44.31 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)

#### 8.44.32 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.

#### 8.44.33 CLCFG\_lilo2Grub

**Description:** Installs and runs the lilo.conf to Grub's menu.lst converter

#### 8.44.34 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.

#### 8.44.35 CLCFG\_setTimeZone

**Description:** Sets the timezone a a client.

**Parameter:**

- timezone: POSIX timezone as defined at /usr/share/zoneinfo/posix/

#### 8.44.36 CLCFG\_writeHosts

**Description:** writes the /etc/hosts file for the client

#### 8.44.37 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 unencrypted 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.

#### 8.44.38 CLCFG\_createBootDeviceNode

**Description:** Makes sure the needed device nodes for the boot device are created.

#### 8.44.39 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.

#### 8.44.40 CLCFG\_genFstab

**Description:** generates the commands to auto detect the partitions and generate the fstab file

**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)

#### 8.44.41 CLCFG\_interfaces

**Description:** generates a script that writes the /etc/network/interfaces file.

**Parameter:**

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

#### 8.44.42 CLCFG\_hostname

**Description:** generates a script that writes the /etc/hostname file.

**Parameter:**

- clientName: the name of the client



#### 8.44.43 CLCFG\_resolvConf

**Description:** generates a script that writes the /etc/resolv.conf file.

**Parameter:**

- DNSServers: list of DNS servers

#### 8.44.44 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

#### 8.44.45 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

#### 8.44.46 CLCFG\_sourceslistCreateConfigFiles

**Description:** Creates config files for the package manager.

**Parameter:**

- sourceName: The name of the package source list

#### 8.44.47 CLCFG\_hwdetect

**Description:** updates and runs the hardware detection

#### 8.44.48 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).

#### 8.44.49 CLCFG\_setRootPassword

**Description:** sets the root password of the client

**Parameter:**

- cryptedPassword: sets an already crypted root password

#### 8.44.50 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

#### 8.44.51 CLCFG\_copyClientPackageStatus

**Description:** copies the package installation status file from the client to the cache directory of the server

**Parameter:**

- clientName: the name of the client

#### 8.44.52 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 packages one by one.

**Parameter:**

- command: install or remove
- packages: white space separated list of packages to install or remove

#### 8.44.53 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.

#### 8.44.54 CLCFG\_installBasePackages

**Description:** installs needed base packages

**Parameter:**

- packagelist: the list of the packages to install
- keyring: name of the keyring package

#### 8.44.55 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

#### 8.44.56 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

#### 8.44.57 CLCFG\_downgradeExt

**Description:** downgrades all ext\* partitions so the fsck.ext\* of woody can understand it

#### 8.44.58 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).

#### 8.44.59 CLCFG\_activateDMA

**Description:** try to activate DMA to speed up installation. this short script tries to detect

#### 8.44.60 CLCFG\_fetchm23BasicTools

**Description:** fetches the basic m23 tools and fsize

#### 8.44.61 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

#### 8.44.62 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

#### 8.44.63 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

#### 8.44.64 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

#### 8.44.65 CLCFG\_executeAfterChroot

**Description:** prepares for pivot\_root and does it and executes afterChroot

**8.44.66 CLCFG\_writeCrontabm23fetchjobEvery5Minutes**

**Description:** Adds entries to crontab to check every 5 minutes for new jobs.

**8.44.67 CLCFG\_writeM23fetchjob**

**Description:** generates the m23fetchjob script

**Parameter:**

- release: Name of the distribution release for special handling of some releases.

**8.44.68 CLCFG\_hideKernelWarnings**

**Description:** hides the kernel warnings

**8.44.69 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.

**8.44.70 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

**8.44.71 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.

**8.44.72 CLCFG\_changeUser**

**Description:** changes the settings of an useraccount on a client

**Parameter:**

- userName: the (old) username for the account
- password: the new unencrypted password for the account
- newUserName: the new username

#### 8.44.73 CLCFG\_patchNsswitchForLDAP

**Description:** Patches /etc/nsswitch.conf for usage with LDAP.

#### 8.44.74 CLCFG\_enableNFSTHome

**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)

#### 8.44.75 CLCFG\_disableNFSTHome

**Description:** Disables storing of home directories on a NFS server

#### 8.44.76 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.

#### 8.44.77 CLCFG\_installApplicationLanguagePackages

**Description:** Installs some additional language packages for installed applications with separate language packs.

**Parameter:**

- lang: short language

#### 8.44.78 CLCFG\_updateDebootstrapScripts

**Description:** Updates the debootstrap scripts for Debian or Ubuntu and returns the www path to the files

#### 8.44.79 CLCFG\_updateDebootstrapScriptsDebian

**Description:** Updates the debootstrap scripts for Debian and returns the www path to the files

## 8.45 *./inc/distr/imaging/clientConfig.php*

Author: Hauke Goos-Habermann (HHabermann@pc-kiel.de)

Description: Imaging specific management functions.

### 8.45.1 *CLCFG\_addDistributionSpecificOptions*

**Description:** adds distribution specific settings to an option array and returns the new array

**Parameter:**

- \$options: the options array with some values

### 8.45.2 *CLCFG\_showDistributionSpecificOptions*

**Description:** shows distribution specific options and returns false, if there was an error

**Parameter:**

- options: options array
- client: Name of the client.

## 8.46 `./inc/distr/imaging/packages.php`

Author: Hauke Goos-Habermann (HHabermann@pc-kiel.de)

Description: Debian/Ubuntu specific package handling functions.

### 8.46.1 `PKG_updatePackageSearchCacheFile`

**Description:** Updates the compressed package search file.

**Parameter:**

- `packagesource`: Name of the package sources list.

**Returns:**

- Full path to the log file.

### 8.46.2 `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.

### 8.46.3 `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.

### 8.46.4 `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.

### 8.46.5 `PKG_ncTarDebsFromClientToServer_Client`

**Description:** Client to send the Debian packages to the m23 client.



### 8.46.6 PKG\_ncTarDebsFromClientToServer\_Server

**Description:** Starts a server to receive the Debian packages from an m3 client.

**Parameter:**

- poolDir: The directory on the server to store the Debian packages.

### 8.46.7 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.

### 8.46.8 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.

### 8.46.9 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.

### 8.46.10 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.

### 8.46.11 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.

#### 8.46.12 **PKG\_searchFor**

**Description:** searches for a package and returns 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.

#### 8.46.13 **PKG\_getDescription**

**Description:** gets the description of a package

**Parameter:**

- line: line containing package name and description

#### 8.46.14 **PKG\_getPackageName**

**Description:** gets the name of a package

**Parameter:**

- line: line containing package name and description

#### 8.46.15 **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.

#### 8.46.16 **PKG\_getPackageDescriptionSize**

**Description:** Returns 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

#### 8.46.17 **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

#### 8.46.18 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.

#### 8.46.19 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.

#### 8.46.20 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.

#### 8.46.21 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.

### 8.46.22 **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.

### 8.46.23 **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

### 8.46.24 **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.

### 8.46.25 **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).

### 8.46.26 **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)

## 8.47 *./inc/distr.php*

Author: Hauke Goos-Habermann (HHabermann@pc-kiel.de)

Description: Functions for handling distributions.

### 8.47.1 *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 separated by a separator character.

### 8.47.2 *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 separated by a separator character.

### 8.47.3 *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.

### 8.47.4 *DISTR\_listDistributions*

**Description:** returns a list of the directory names of the distributions

**Parameter:**

- `addEmpty`: set to true if an empty fake distribution should be put on top of the list

**Returns:**

- An array that contains the system names of all distributions.

### 8.47.5 *DISTR\_getDescriptionValues*

**Description:** gets the values stored in the info.txt file of the distributions and returns an associative array

**Parameter:**

- `shortName`: the short name of the distribution

### 8.47.6 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

### 8.47.7 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

### 8.47.8 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

### 8.47.9 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

### 8.47.10 DISTR\_getDesktopsCBLList

**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

### 8.47.11 DISTR\_getDesktopDescription

**Description:** Returns the description for the given desktop in the given distribution preferredly in the language of the m23 webinterface.

**Parameter:**

- distr: Name of the distribution.
- desktop: Name of the desktop.

**8.47.12 DISTR\_getSelectedDesktopsArr**

**Description:** returns an array with selected desktops from the list generated by DISTR\_getDesktopsCBLi

**8.47.13 DISTR\_getSelectedDesktopsStr**

**Description:** returns a string with selected desktops (seperated by "###") from the list generated by DISTR\_getDesktopsCBLi

## 8.48 `./inc/distr/ubuntu/clientConfigCommon.php`

Author: Hauke Goos-Habermann (HHabermann@pc-kiel.de)

Description: Common management funtions shared by Linux distributions.

### 8.48.1 `CLCFG_setSystemtime`

**Description:** Sets the system time of the client to about the same time of the m23 server.

### 8.48.2 `CLCFG_disableAptSystemdDaily`

**Description:** Deactivates systemd apt-daily.timer.

### 8.48.3 `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.

### 8.48.4 `CLCFG_disableSudoRootLogin`

**Description:** Disables getting root rights of normal users via sudo.

### 8.48.5 `CLCFG_installFirmware`

**Description:** Installs available firmware packages.

### 8.48.6 `CLCFG_monoRemove`

**Description:** Removes mono packages and installs gnote when tomboy was installed before.

### 8.48.7 `CLCFG_setDebConfDM`

**Description:** Sets the display manager in debconf.

**Parameter:**

- dm: Name of the display manager (kdm, lightdm, mdm, ...).

### 8.48.8 `CLCFG_installXDM`

**Description:** Installs the XDM display manager.

### 8.48.9 `CLCFG_installSDDM`

**Description:** Installs the SDDM display manager.



#### 8.48.10 CLCFG\_installMintDM

**Description:** Installs the Linux Mint DM display manager.

#### 8.48.11 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).

#### 8.48.12 CLCFG\_activateBOOT\_DEGRADED

**Description:** Activates BOOT\_DEGRADED on Ubuntu 12.04 to allow booting from degraded RAIDs.

#### 8.48.13 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.

#### 8.48.14 CLCFG\_setDebConfDirect

**Description:** Sets debconf settings.

**Parameter:**

- debconf: Debconf settings to add.

#### 8.48.15 TRINITY\_installLoginManager

**Description:** Installs the Trinity login manager KDM.

**Parameter:**

- lang: short language

#### 8.48.16 TRINITY\_install

**Description:** Installs a minimalistic Trinity desktop.

**Parameter:**

- lang: short language

#### 8.48.17 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.

#### 8.48.18 CLCFG\_disablePlymouth

**Description:** Disables the plymouth.

#### 8.48.19 CLCFG\_disableAvahiDaemon

**Description:** Disables the avahi-daemon.

#### 8.48.20 CLCFG\_configUpstartForNormalUsage

**Description:** Configures upstart for normal running in an installed system.

#### 8.48.21 CLCFG\_configUpstartForChroot

**Description:** Configures upstart to make it not fail installation.

#### 8.48.22 CLCFG\_createScreenRC

**Description:** Creates the (under Ubuntu) needed settings for screen.

#### 8.48.23 CLCFG\_addGrubPassword

**Description:** Adds a password line to the grub configuration to lock the edit line of bootmanager.

#### 8.48.24 CLCFG\_addLiloPassword

**Description:** Adds a password line to the LiLo configuration to lock the edit line of bootmanager.

#### 8.48.25 GNOME\_prepare

**Description:** Prepares the GNOME installation

#### 8.48.26 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

#### 8.48.27 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

### 8.48.28 GNOME\_install

**Description:** Installs a minimalistic GNOME desktop.

**Parameter:**

- lang: short language

### 8.48.29 GNOME\_installLoginManager

**Description:** Installs the GNOME login manager GDM.

**Parameter:**

- lang: short language

### 8.48.30 KDE\_prepare

**Description:** Prepares the KDE installation

### 8.48.31 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)

### 8.48.32 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.

### 8.48.33 CLCFG\_lilo2Grub

**Description:** Installs and runs the lilo.conf to Grub's menu.lst converter

### 8.48.34 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.

### 8.48.35 CLCFG\_setTimeZone

**Description:** Sets the timezone a a client.

**Parameter:**

- timezone: POSIX timezone as defined at /usr/share/zoneinfo/posix/

### 8.48.36 CLCFG\_writeHosts

**Description:** writes the /etc/hosts file for the client

### 8.48.37 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 unencrypted 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.

### 8.48.38 CLCFG\_createBootDeviceNode

**Description:** Makes sure the needed device nodes for the boot device are created.

### 8.48.39 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.

### 8.48.40 CLCFG\_genFstab

**Description:** generates the commands to auto detect the partitions and generate the fstab file

**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)

### 8.48.41 CLCFG\_interfaces

**Description:** generates a script that writes the /etc/network/interfaces file.

**Parameter:**

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

### 8.48.42 CLCFG\_hostname

**Description:** generates a script that writes the /etc/hostname file.

**Parameter:**

- clientName: the name of the client

#### 8.48.43 CLCFG\_resolvConf

**Description:** generates a script that writes the /etc/resolv.conf file.

**Parameter:**

- DNSServers: list of DNS servers

#### 8.48.44 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

#### 8.48.45 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

#### 8.48.46 CLCFG\_sourceslistCreateConfigFiles

**Description:** Creates config files for the package manager.

**Parameter:**

- sourceName: The name of the package source list

#### 8.48.47 CLCFG\_hwdetect

**Description:** updates and runs the hardware detection

#### 8.48.48 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).

#### 8.48.49 CLCFG\_setRootPassword

**Description:** sets the root password of the client

**Parameter:**

- cryptedPassword: sets an already crypted root password

#### 8.48.50 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

#### 8.48.51 CLCFG\_copyClientPackageStatus

**Description:** copies the package installation status file from the client to the cache directory of the server

**Parameter:**

- clientName: the name of the client

#### 8.48.52 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 packages one by one.

**Parameter:**

- command: install or remove
- packages: white space separated list of packages to install or remove

#### 8.48.53 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.

#### 8.48.54 CLCFG\_installBasePackages

**Description:** installs needed base packages

**Parameter:**

- packagelist: the list of the packages to install
- keyring: name of the keyring package

#### 8.48.55 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

### 8.48.56 `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-separated list)
- include: additional packages to include
- isCritical: selects if debootstrap errors should be critical
- additionalPackages: space separated list of additional packages that should be installed during bootstrapping

### 8.48.57 `CLCFG_downgradeExt`

**Description:** downgrades all ext\* partitions so the fsck.ext\* of woody can understand it

### 8.48.58 `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).

### 8.48.59 `CLCFG_activateDMA`

**Description:** try to activate DMA to speed up installation. this short script tries to detect

### 8.48.60 `CLCFG_fetchm23BasicTools`

**Description:** fetches the basic m23 tools and fsize

### 8.48.61 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

### 8.48.62 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

### 8.48.63 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

### 8.48.64 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

### 8.48.65 CLCFG\_executeAfterChroot

**Description:** prepares for pivot\_root and does it and executes afterChroot



**8.48.66 CLCFG\_writeCrontabm23fetchjobEvery5Minutes**

**Description:** Adds entries to crontab to check every 5 minutes for new jobs.

**8.48.67 CLCFG\_writeM23fetchjob**

**Description:** generates the m23fetchjob script

**Parameter:**

- release: Name of the distribution release for special handling of some releases.

**8.48.68 CLCFG\_hideKernelWarnings**

**Description:** hides the kernel warnings

**8.48.69 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.

**8.48.70 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

**8.48.71 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.

**8.48.72 CLCFG\_changeUser**

**Description:** changes the settings of an useraccount on a client

**Parameter:**

- userName: the (old) username for the account
- password: the new unencrypted password for the account
- newUserName: the new username

#### 8.48.73 CLCFG\_patchNsswitchForLDAP

**Description:** Patches /etc/nsswitch.conf for usage with LDAP.

#### 8.48.74 CLCFG\_enableNFSTHome

**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)

#### 8.48.75 CLCFG\_disableNFSTHome

**Description:** Disables storing of home directories on a NFS server

#### 8.48.76 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.

#### 8.48.77 CLCFG\_installApplicationLanguagePackages

**Description:** Installs some additional language packages for installed applications with separate language packs.

**Parameter:**

- lang: short language

#### 8.48.78 CLCFG\_updateDebootstrapScripts

**Description:** Updates the debootstrap scripts for Debian or Ubuntu and returns the www path to the files

#### 8.48.79 CLCFG\_updateDebootstrapScriptsDebian

**Description:** Updates the debootstrap scripts for Debian and returns the www path to the files

## 8.49 *./inc/distr/ubuntu/clientConfig.php*

Author: Hauke Goos-Habermann (HHabermann@pc-kiel.de)

Description: Ubuntu specific management functions.

### 8.49.1 *CLCFG\_showDistributionSpecificOptionsUbuntu*

**Description:** Shows additional distribution specific options for Ubuntu.

**Parameter:**

- options: options array

**Returns:**

- Options array or false, if there was an error

### 8.49.2 *UBUNTU\_installLanguagePacks*

**Description:** Installs the language packs on Ubuntu.

**Parameter:**

- lang: short language

### 8.49.3 *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 used 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.

### 8.49.4 *UBUNTU\_fixBeforeBaseInstall*

**Description:** Corrects the settings for Ubuntu before the base packages are installed.

**Parameter:**

- release: Name of the Ubuntu release.

### 8.49.5 *UBUNTU\_fixAfterBaseInstall*

**Description:** Corrects the settings for Ubuntu after the base packages are installed.

**Parameter:**

- release: Name of the Ubuntu release.

### 8.49.6 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

### 8.49.7 CLCFG\_updateDebootstrapScriptsUbuntu

**Description:** Updates the debootstrap scripts for Ubuntu and returns the www path to the files

### 8.49.8 CLIENT\_enableLDAP

**Description:** enables LDAP login for a client.

**Parameter:**

- clientOptions: the client's options array

## 8.50 *./inc/distr/ubuntu/packages.php*

Author: Hauke Goos-Habermann (HHabermann@pc-kiel.de)

Description: Debian/Ubuntu specific package handling functions.

### 8.50.1 *PKG\_updatePackageSearchCacheFile*

**Description:** Updates the compressed package search file.

**Parameter:**

- *packagesource*: Name of the package sources list.

**Returns:**

- Full path to the log file.

### 8.50.2 *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.

### 8.50.3 *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.

### 8.50.4 *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.

### 8.50.5 *PKG\_ncTarDebsFromClientToServer\_Client*

**Description:** Client to send the Debian packages to the m23 client.

### 8.50.6 `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.

### 8.50.7 `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.

### 8.50.8 `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.

### 8.50.9 `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.

### 8.50.10 `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.

### 8.50.11 `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.

### 8.50.12 PKG\_searchFor

**Description:** searches for a package and returns 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.

### 8.50.13 PKG\_getDescription

**Description:** gets the description of a package

**Parameter:**

- line: line containing package name and description

### 8.50.14 PKG\_getPackageName

**Description:** gets the name of a package

**Parameter:**

- line: line containing package name and description

### 8.50.15 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.

### 8.50.16 PKG\_getPackageDescriptionSize

**Description:** Returns 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

### 8.50.17 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

### 8.50.18 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.

### 8.50.19 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.

### 8.50.20 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.

### 8.50.21 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.



### 8.50.22 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.

### 8.50.23 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

### 8.50.24 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.

### 8.50.25 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).

### 8.50.26 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)

## 8.51 `./inc/edit.php`

Author: Hauke Goos-Habermann (HHabermann@pc-kiel.de)

Description: routines for editing files with sed

### 8.51.1 `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.

### 8.51.2 `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.

### 8.51.3 `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.

### 8.51.4 `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

### 8.51.5 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)

### 8.51.6 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)

### 8.51.7 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

### 8.51.8 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")

### 8.51.9 EDIT\_uncomment

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

**Parameter:**

- file: the name of the file

### 8.51.10 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:

### 8.51.11 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.

### 8.51.12 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.

### 8.51.13 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.

### 8.51.14 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

### 8.51.15 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

### 8.51.16 EDIT\_searchNextLineNumber

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

**Parameter:**

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

### 8.51.17 EDIT\_replace

**Description:** replaces \$searchLine with \$replaceText

**Parameter:**

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

### 8.51.18 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

### 8.51.19 EDIT\_savePerms

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

**Parameter:**

- file: the name to the file

### 8.51.20 EDIT\_restorePerms

**Description:** restores previously saved file permissions and owner

### 8.51.21 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

### 8.51.22 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

### 8.51.23 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

### 8.51.24 EDIT\_deleteMatching

**Description:** Deletes all lines matching the regular expression

**Parameter:**

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

### 8.51.25 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

### 8.51.26 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.

## 8.52 *./inc/fdisk.php*

Author: Hauke Goos-Habermann (HHabermann@pc-kiel.de)

Description: functions for partition and format of the client. print partition information and do the format itself.

### 8.52.1 *FDISK\_showFdiskCombinedGUIFunctions*

**Description:** Shows the menu bar with integrated logic for *FDISK\_showCombinedFdiskGUIDialog*.

### 8.52.2 *FDISK\_showCombinedFdiskGUIDialog*

**Description:** Shows the new partition and formatting screen.

### 8.52.3 *FDISK\_mdToEndOfArray*

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

**Parameter:**

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

**Returns:**

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

### 8.52.4 *FDISK\_getFstabArray*

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

**Parameter:**

- client: Name of the client.

**Returns:**

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

### 8.52.5 *FDISK\_findFstabMountPointByDev*

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

**Parameter:**

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

**Returns:**

- Mount point for the device.

### 8.52.6 *FDISK\_swapFilesystems*

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

### 8.52.7 `FDISK_formatInstallAndSwappart`

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

### 8.52.8 `FDISK_finalChecksAndRealPartitionAndFormatStart`

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

### 8.52.9 `FDISK_installFilesystems`

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

### 8.52.10 `FDISK_getUnusedMDs`

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

**Parameter:**

- param: parameter string containing status informations about the harddisks

**Returns:**

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

### 8.52.11 `FDISK_listDrivesAndPartitions2`

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

**Parameter:**

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

**Returns:**

- String with the HTML selection.

### 8.52.12 `FDISK_printAllBars2`

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

### 8.52.13 `FDISK_showAllPartTables`

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



#### 8.52.14 FDISK\_fdiskSessionPartJobs

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

**Parameter:**

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

**Returns:**

- The current partition jobs.

#### 8.52.15 FDISK\_fdiskSessionInstPart

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

**Parameter:**

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

**Returns:**

- The current installation partition.

#### 8.52.16 FDISK\_fdiskSessionSwapPart

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

**Parameter:**

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

**Returns:**

- The current swap partition.

#### 8.52.17 FDISK\_fdiskSessionSetter

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

**Parameter:**

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

**Returns:**

- The current value.

#### 8.52.18 FDISK\_fdiskSessionClient

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

**Returns:**

- The client name to partition and format.

#### 8.52.19 FDISK\_fdiskSessionParam

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

**Parameter:**

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

**Returns:**

- The current partition parameters.

### 8.52.20 FDISK\_fdiskSessionInstallDrive

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

**Parameter:**

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

**Returns:**

- The current installation drive.

### 8.52.21 FDISK\_fdiskSessionvDevInstall

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

**Parameter:**

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

**Returns:**

- The current internal virtual installation drive number.

### 8.52.22 FDISK\_fdiskSessionFreeSpaces

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

**Parameter:**

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

**Returns:**

- The current free spaces.

### 8.52.23 FDISK\_fdiskSessionReset

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

**Parameter:**

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

### 8.52.24 FDISK\_fdiskSessionPartMethod

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

**Parameter:**

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

**Returns:**

- The current partitioning method.

### 8.52.25 FDISK\_fdiskSessionPage

**Description:** Stores the page in the session.

**Parameter:**

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

**Returns:**

- The current page.

### 8.52.26 FDISK\_fdiskSessionHelpPage

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

**Parameter:**

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

**Returns:**

- The current help page.

### 8.52.27 FDISK\_fdiskSessionTitle

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

**Parameter:**

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

**Returns:**

- The current partitioning title.

### 8.52.28 FDISK\_fdiskSessionFstab

**Description:** Stores the fstab in the session.

**Parameter:**

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

**Returns:**

- The current fstab.

### 8.52.29 FDISK\_getPartitionByType

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

**Parameter:**

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

**Returns:**

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

### 8.52.30 FDISK\_getDrivesAndPartitions

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

**Parameter:**

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

**Returns:**

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

### 8.52.31 FDISK\_listDrivesAndPartitions

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

**Parameter:**

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

**Returns:**

- String with the HTML selection.

### 8.52.32 FDISK\_selectDrives

**Description:** creates a selection list of all drives

**Parameter:**

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

### 8.52.33 FDISK\_printAllBars

**Description:** shows the partitions bars of all available drives

**Parameter:**

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

### 8.52.34 FDISK\_getFirstDrive

**Description:** return the first drive as installation drive

**Parameter:**

- param: parameter string containing status informations about the harddisks

### 8.52.35 FDISK\_formatPart

**Description:** formats a partition

**Parameter:**

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

### 8.52.36 FDISK\_getBiggestValueOf

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

**Parameter:**

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

### 8.52.37 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 number you want to check

### 8.52.38 FDISK\_nextLogicalDevNr

**Description:** gets the next free logical device number

**Parameter:**

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

### 8.52.39 FDISK\_nextPrimaryDevNr

**Description:** gets the next free primary device number

**Parameter:**

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

### 8.52.40 FDISK\_correctLogical

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

**Parameter:**

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

### 8.52.41 FDISK\_findDevNrPosition

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

**Parameter:**

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

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

#### 8.52.42 FDISK\_partCreationSelect

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

**Parameter:**

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

#### 8.52.43 FDISK\_canPartTypeBeCreated

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

**Parameter:**

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

#### 8.52.44 FDISK\_checkFreeSpace

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

**Parameter:**

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

#### 8.52.45 FDISK\_installExistingDialog

**Description:** shows the dialog for installation on existing partitions

**Parameter:**

- param: parameter string containing status informations about the harddisks

#### 8.52.46 FDISK\_addFstab

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

**Parameter:**

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

### 8.52.47 FDISK\_delFstab

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

**Parameter:**

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

### 8.52.48 FDISK\_listFstab

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

**Parameter:**

- param: parameter string containing status informations about the harddisks

**Returns:**

- HTML table with the fstab.

### 8.52.49 FDISK\_fstabAddDialog2

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

### 8.52.50 FDISK\_fstabAddDialog

**Description:** Dialog for adding fstab entries.

**Parameter:**

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

### 8.52.51 FDISK\_adjustFstabParam

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

**Parameter:**

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

**Returns:**

- Adjust the parameter block of a fstab line

### 8.52.52 FDISK\_genManualFstab

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

**Parameter:**

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

### 8.52.53 FDISK\_getBelongingRaidDev

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

**Parameter:**

- dev: The physical partition (e.g. /dev/hda4) that belongs to a RAID.
- param: parameter string containing status informations about the harddisks

**Returns:**

- The RAID device (e.g. /dev/md0) the physical partition belongs to or false, if no belonging RAID was found.

### 8.52.54 FDISK\_delPart

**Description:** deletes a partition from the param string and generates the parted commands

**Parameter:**

- dev: the partition to delete (e.g. /dev/hda4)
- param: parameter string containing status informations about the harddisks
- partJobs: parted commands
- deleteBelongingRaid: If set to true, the RAID, the partition belongs to will be destroyed.

**Returns:**

- Changed param string.

### 8.52.55 FDISK\_addPart

**Description:** adds a partition to the param string and generates the parted commands

**Parameter:**

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

**Returns:**

- : Changed param string.

### 8.52.56 FDISK\_listPartitions

**Description:** lists the partitions (/dev/hda1, /dev/hda2, ...) of a device and generates a selection

**Parameter:**

- param: parameter string containing status informations about the harddisks
- vDev: Virtual (internally used) device number or -1, if all partitions on all devices should be listed.
- selName: name of the selection
- excludeType: type of partitions, not to show in the selection

**Returns:**

- HTML code for the selection.



### 8.52.57 FDISK\_definePartitionSelection

**Description:** Defines a HTML selection with the partitions (/dev/hda1, /dev/hda2, ...) of a device

**Parameter:**

- param: parameter string containing status informations about the harddisks
- vDev: Virtual (internally used) device number or -1, if all partitions on all devices should be listed.
- selName: name of the selection
- excludeType: type of partitions, not to show in the selection

**Returns:**

- The selected partition.

### 8.52.58 FDISK\_getPartitionsFromParam

**Description:** Returns an array with the partitions (/dev/hda1, /dev/hda2, ...) of a device

**Parameter:**

- param: parameter string containing status informations about the harddisks
- vDev: Virtual (internally used) device number or -1, if all partitions on all devices should be listed.
- selName: name of the selection
- excludeType: type of partitions, not to show in the selection

**Returns:**

- Selected partition.

### 8.52.59 FDISK\_getAllDrives

**Description:** gets all drives of the client

**Parameter:**

- partitions: associative array containing status information about the harddisks

### 8.52.60 FDISK\_colorFS

**Description:** get color for a selected filesystem

**Parameter:**

- fsName: name of the file system: ext3, ext2, linux-swap,...

### 8.52.61 FDISK\_getPartitionPercent

**Description:** calculates the percent of a selected partition

**Parameter:**

- param: parameter string containing status informations about the harddisks
- vDev: Virtual (internally used) device number.
- vPart: Virtual (internally used) partition number. This is normally another number than the physical number (e.g. 1 on /dev/hda1)

### 8.52.62 FDISK\_getAfterPartition

**Description:** calculates ??? of free size after a selected partition

**Parameter:**

- param: parameter string containing status informations about the harddisks
- vDev: Virtual (internally used) device number.
- vPart: Virtual (internally used) partition number. This is normally another number than the physical number (e.g. 1 on /dev/hda1)
- factor: the factor to multiplay percent amount of free space

### 8.52.63 FDISK\_getBeforeFristPartition

**Description:** gets the free space before the first partition

**Parameter:**

- param: parameter string containing status informations about the harddisks
- vDev: virtual device number to access the drive
- factor: the factor to multiplay percent amount of free space

### 8.52.64 FDISK\_getPartitions

**Description:** get the partition info for the client from db

**Parameter:**

- client: name of the client

### 8.52.65 FDISK\_getPartInfoString

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

**Parameter:**

- vDev: Virtual (internally used) device number.
- vPart: Virtual (internally used) partition number. This is normally another number than the physical number (e.g. 1 on /dev/hda1)
- param: parameter string containing status informations about the harddisks
- fstabA: Associative array with fstab information.

**Returns:**

- Info string.

### 8.52.66 FDISK\_getDriveInfoString

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

**Parameter:**

- vDev: Virtual (internally used) device number.
- param: parameter string containing status informations about the harddisks
- fstabA: Associative array with fstab information.

**Returns:**

- Info string.

**8.52.67 FDISK\_getDriveInfoIcon****Description:** Generates HTML code for showing an icon with status information about a drive.**Parameter:**

- vDev: Virtual (internally used) device number.
- param: parameter string containing status informations about the harddisks
- fstabA: Associative array with fstab information.

**Returns:**

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

**8.52.68 FDISK\_getPartInfoIcon****Description:** Generates HTML code for showing an icon with status information about a drive or partition.**Parameter:**

- vDev: Virtual (internally used) device number.
- vPart: Virtual (internally used) partition number. This is normally another number than the physical number (e.g. 1 on /dev/hda1) and if set to false, the icon and the status information will be generated for a drive and not for a partition.
- param: parameter string containing status informations about the harddisks
- fstabA: Associative array with fstab information.

**Returns:**

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

**8.52.69 FDISK\_printBars****Description:** prints the partitions as colored table**Parameter:**

- param: parameter string containing status informations about the harddisks
- dev: selected device (e.g. /dev/hda)
- addJavaScript: Set to true to add JavaScript code that calls the JS function emptySpace(), if empty parts of the drive are clicked, selectPartition(), if a partition is clicked and showPartTable(), if the mouse is over the bar.
- fstabA: Associative array with fstab information.

**8.52.70 FDISK\_getSupportedFS****Description:** Generates and returns an array with the list of supported file systems.**Returns:**

- Array with the list of supported file systems.

**8.52.71 FDISK\_listSupportedFS****Description:** lists all supported fileSystems for the menu.**Parameter:**

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

**Returns:**

- The currently choosen file system.

### 8.52.72 **FDISK\_deletePartitionFromParam**

**Description:** Deletes all partition parameters of a partition from param without correcting the other partitions.

**Parameter:**

- vDev: Virtual (internally used) device number.
- vPart: Virtual (internally used) partition number. This is normally another number than the physical number (e.g. 1 on /dev/hda1)
- param: parameter string containing status informations about the harddisks.

**Returns:**

- Changed param without the partition.

### 8.52.73 **FDISK\_virtualDeletePartition**

**Description:** deletes partition from param assigned thru \$vDev and \$vPart.

**Parameter:**

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

### 8.52.74 **FDISK\_virtualAddPartition**

**Description:** adds a partition to the param param

**Parameter:**

- vDev: virtual 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: returns the device number

### 8.52.75 **FDISK\_listPartTable**

**Description:** lists the partition information as table

**Parameter:**

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

### 8.52.76 **FDISK\_listInstPartSelector**

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

**Parameter:**

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

### 8.52.77 **FDISK\_formatExisting**

**Parameter:**

- instPart: partition to put the operation system on (e.g. /dev/hda1)
- swapPart: partition to put the swap file system on (e.g. /dev/hdb2)
- command: parted commands to do the installation
- param: parameter string containing status informations about the harddisks

### 8.52.78 **FDISK\_getvPart**

**Description:** returns vPart with the real device number.

**Parameter:**

- param: parameter string containing status informations about the harddisks
- dev: selected device (e.g. hda)
- devNr: number of partition

### 8.52.79 **FDISK\_rmJob**

**Description:** generates a partition remove job

**Parameter:**

- dev: selected device (e.g. hda)
- devNr: number of partition, minor number in parted
- partJobs: associative array with partition jobs

### 8.52.80 **FDISK\_addJob**

**Description:** generates a partition add job

**Parameter:**

- path: selected device (e.g. /dev/hda)
- start: start point fo the partition
- end: end point fo the partition
- type: type of the partition (primary, logical)
- partJobs: associative array with partition jobs
- fullPath: full path for the partition
- devNr: number of the device (e.g. 1 with /dev/hda1)

### 8.52.81 **FDISK\_bootflagJob**

**Description:** enables the booting flag on a partition

**Parameter:**

- path: device to activate booting on (e.g. /dev/hda1)
- devNr: number of partition, minor number in parted
- partJobs: associative array with partition jobs

### 8.52.82 FDISK\_formatJob

**Description:** generates a partition format job

**Parameter:**

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

### 8.52.83 FDISK\_countPartitions

**Description:** count all partitions of a selected type

**Parameter:**

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

### 8.52.84 FDISK\_getFreeSpaces

**Description:** get free spaces as array

**Parameter:**

- param: parameter string containing status informations about the harddisks
- dev: selected device (e.g. hda)

### 8.52.85 FDISK\_autoPart

**Description:** generation of param string and command list for automatic partition

**Parameter:**

- clientName: name of the client
- command: parted commands are written to \$command
- dev: selected device (e.g. /dev/hda)
- param: parameter string containing status informations about the harddisks
- instPart: the variable the installation device name is written to
- swapPart: the variable the swap device name is written to
- minSwap: Minimal size of the swap partition in MB.
- maxSwap: Maximal size of the swap partition in MB.

### 8.52.86 FDISK\_printColorDefinitions

**Description:** prints the color definitions for the filesystems

### 8.52.87 FDISK\_showDiskDefine

**Description:** shows a dialog for defining the type and size of the fake drive for the clientBuilder

**Parameter:**

- client: client name

### 8.52.88 FDISK\_defineDrive

**Description:** defines drive information for the clientBuilder

**Parameter:**

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

### 8.52.89 FDISK\_dev2LDevLPart

**Description:** searches a special device (e.g. /dev/hda2) and writes the virtual device and partition numbers to the variables. These values can be used to access the file system via \$param["dev\$vDev"."part\$vPart"."\_fs"]

**Parameter:**

- param: the associative array containing all values describing the drives of the client
- dev: the device (e.g. /dev/hda2)
- vDev: the virtual device number, that is used to build the variable name to access the associative array.
- vPart: the virtual partition number, that is used to build the variable name to access the associative array. This number has not to be qual to the partition number of the real drive (e.g. /dev/hda5 can be \$vPart == 3). If it is set to "empty", only vDev is calculated.

**Returns:**

- true if the search worked otherwise false.

### 8.52.90 FDISK\_rereadPartTable

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

**Parameter:**

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

### 8.52.91 FDISK\_genPartedCommands

**Description:** returns the partition and formation commands that are generated from partJobs.

**Parameter:**

- partJobs: string with information about all created partition jobs
- mkfsExtOptions: Extra parameter for mkfs.extX .

### 8.52.92 FDISK\_listPartJobs

**Description:** print all part jobs in the table

**Parameter:**

- partJobs: string with information about all created partition jobs

### 8.52.93 FDISK\_getDiskType

**Description:** returns the type of the drive (DISK\_TYPE\_IDE, DISK\_TYPE\_SCSI)

**Parameter:**

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

### 8.52.94 FDISK\_getDriveAndNr

**Description:** splits a path (e.g. /dev/hda1) in the device (/dev/hda) and the device number (1). The device is returned as element 0 and the number as element 1 in an array.

**Parameter:**

- path: the path to partition (e.g. /dev/hde1)

**Returns:**

- Array with two parts. \$out[0]=drive (e.g. /dev/hda), \$out[1]=the device number

### 8.52.95 FDISK\_getNextFdiskFormatJobNr

**Description:** returns the next free job number for the parameters of a m23fdiskFormat job. (e.g. there are used the following parameters: command0 = rm, command1= add. Then the next command number to use will be command2 => return value is 2)

**Parameter:**

- fdiskParams: the parameters of the m23fdiskFormat job

### 8.52.96 FDISK\_AFPselectDrive

**Description:** selects a drive from the settings in "options" and from available drives.

**Parameter:**

- drives: all drives available on the client
- options: options array of the client

### 8.52.97 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 modified format parameters
- formatarr: array that contains the original format parameters



**8.52.98 FDISK\_AFPgetPartSizes****Description:** writes the sizes of the installation, swap and other partitions to the variables.**Parameter:**

- formatArr: array that contains the oformat parameters
- options: options array of the client
- instPartSize: stores the size of the installation partition
- instSwapSize: stores the size of the swap partition
- otherSize: stores the size of the other partition(s)

**8.52.99 FDISK\_adjustFdiskParams****Description:** adjusts the installation and swap drive for a derived client, based on the defined client settings**Parameter:**

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

**8.52.100 FDISK\_virtualAddDrive****Description:** Adds a new drive to the param array.**Parameter:**

- param: parameter string containing status informations about the harddisks
- path: Device name of the new drive (e.g. /dev/md0)
- size: Size in MB of the new drive.

**8.52.101 FDISK\_getDrivePartitionSize****Description:** Calculates the size of a drive or partition.**Parameter:**

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

**Returns:**

- Size of the drive or partition in MB.

**8.52.102 FDISK\_listRaidTable****Description:** Get informations about the assigned real drives/partitions of a RAID.**Parameter:**

- raidDev: Device name of the new drive (e.g. /dev/md0)
- param: parameter string containing status information about the harddisks

**Returns:**

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

### 8.52.103 **FDISK\_addDrivePartitionToRaid**

**Description:** Adds a drive or partition to a RAID.

**Parameter:**

- raidDev: Device name of the new drive (e.g. /dev/md0)
- raidType: RAID level (this can be 0,1,4,5,6,10)
- partitionDrive: Partition or drive to add (e.g. /dev/hdc)
- param: parameter string containing status information about the harddisks
- raidMode: Raid mode (e.g. 1 for RAID-1, 5 for RAID-5)

**Returns:**

- true if the RAID has the minimum amount of assigned drives/partitions and otherwise false.

### 8.52.104 **FDISK\_buildRaidDialog**

**Description:** Shows a dialog for creating RAIDs of different types and assign real drives or partitions.

**Parameter:**

- param: parameter string containing status information about the harddisks
- partJobs: associative array with partition jobs
- currentDrive: the current drive to work on (e.g. hda)
- helpPage: Name of the help page to show.
- partitionDrive: Partition or drive to add (e.g. /dev/hdc)
- partMethod: partition method (used for the partition/format dialog). The next step will depend on this value.

**Returns:**

- true if the RAID has the minimum amount of assigned drives/partitions and otherwise false.

### 8.52.105 **FDISK\_addRaidJobs**

**Description:** Generates jobs to create all RAIDs

**Parameter:**

- partJobs: associative array with partition jobs
- param: parameter string containing status information about the harddisks

### 8.52.106 **FDISK\_addRaidBeforeFormat**

**Description:** Generates and places a job to create a RAID on given drives/partitions before the formatting of the RAID device.

**Parameter:**

- raidDev: RAID device (e.g. /dev/md0)
- devList: Space separated list of devices to create the RAID on top (e.g. /dev/sda1 /dev/hda /dev/sbd5).
- partJobs: associative array with partition jobs.
- raidMode: The type of the RAID (0,1,5, ...)

**8.52.107 FDISK\_raidJob**

**Description:** Generates a job to create a RAID on given drives/partitions.

**Parameter:**

- raidDev: RAID device (e.g. /dev/md0)
- devList: Space separated list of devices to create the RAID on top (e.g. /dev/sda1 /dev/hda /dev/sbd5).
- partJobs: associative array with partition jobs

**8.52.108 FDISK\_virtualDeleteDrive**

**Description:** Deletes a (RAID) drive from param assigned thru \$vDev.

**Parameter:**

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

**8.52.109 FDISK\_deleteDriveFromParam**

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

**Parameter:**

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

**Returns:**

- Changed param without the drive.

## 8.53 `./inc/groups.php`

Author: Hauke Goos-Habermann (HHabermann@pc-kiel.de)

Description: group management funtions

### 8.53.1 `GRP_exists`

**Description:** checks, if a group exists

**Parameter:**

- `groupName`: name of the group to check

### 8.53.2 `GRP_add`

**Description:** Adds a group with description.

**Parameter:**

- `groupName`: name of the group to add
- `groupDescription`: description of the group to add

### 8.53.3 `GRP_getIdByName`

**Description:** gets the Id of a groupname

**Parameter:**

- `groupName`: name of the group

**Returns:**

- Group number or false, if no matching group is found.

### 8.53.4 `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.

### 8.53.5 `GRP_del`

**Description:** deletes all clients from the group and the group itself

**Parameter:**

- `groupName`: name of the group

### 8.53.6 `GRP_isClientInGroup`

**Description:** returns true, if a client is in the selected group, otherwise false

**Parameter:**

- `clientName`: name of the client
- `groupName`: name of the group

### 8.53.7 GRP\_addClientToGroup

**Description:** adds a client to a group

**Parameter:**

- clientName: name of the client
- groupName: name of the group

### 8.53.8 GRP\_delClientFromGroup

**Description:** removes a client from a group

**Parameter:**

- clientName: name of the client
- groupName: name of the group

### 8.53.9 GRP\_setDescrGroup

**Description:** Sets the description of a client group.

**Parameter:**

- groupName: Name of the group
- groupDescription: New description of the group

### 8.53.10 GRP\_getDescrGroup

**Description:** Gets the description of a client group.

**Returns:**

- Description of the group.

### 8.53.11 GRP\_listGroupsAndCount

**Description:** returns a associative array with all groupnames and the amount of clients in each group

### 8.53.12 GRP\_showGroupsAndCount

**Description:** generates a table with all groupnames and the amount of clients in each group

### 8.53.13 GRP\_ren

**Description:** renames a group

**Parameter:**

- groupName: name of the group
- newGroupName: name of the new group

### 8.53.14 GRP\_HTMLBackToDetails

**Description:** generates HTML code to return to the group details page

**Parameter:**

- groupName: name of the group
- section: name of the section to jump to

### 8.53.15 GRP\_countClients

**Description:** returns the amount of client of a certain group

**Parameter:**

- groupName: name of the group

### 8.53.16 GRP\_showGeneralInfo

**Description:** shows a table with general information about the group

**Parameter:**

- groupName: name of the group

### 8.53.17 GRP\_showRenDialog

**Description:** shows a dialog to rename a group

**Parameter:**

- groupName: name of the group

### 8.53.18 GRP\_showChangeDescriptionDialog

**Description:** shows a dialog to change the group description

**Parameter:**

- groupName: name of the group

### 8.53.19 GRP\_moveClientToGroup

**Description:** moves a client from one group to another

**Parameter:**

- clientName: client to move
- oldGroup: name of the old group
- newGroup: name of the new group

### 8.53.20 GRP\_listGroups

**Description:** returns all groups in an array

### 8.53.21 GRP\_groupSelection

**Description:** generates a HTML selection with all groups as options

**Parameter:**

- selName: name of the selection
- first: the group that should be shown first

### 8.53.22 GRP\_showDelDialog

**Description:** shows a dialog for deleting a group

**Parameter:**

- groupName: name of the group

### 8.53.23 GRP\_doClientMoreGroups

**Description:** dialog and logic for adding and removing the client to and from multiple groups

**Parameter:**

- clientName: name of the client
- type: type of the action ("add" for adding,"del" for removing)

### 8.53.24 GRP\_listClientGroups

**Description:** returns an array containing all groups a client is in

**Parameter:**

- clientName: name of the client

### 8.53.25 GRP\_showClientGroups

**Description:** Shows a list containing all groups a client is in

**Parameter:**

- clientName: name of the client
- link: if there should be links to the group page
- output: If set to true, the list will be shown, if set to false returned instead.

**Returns:**

- Nothing or the list containing all groups a client is in.

### 8.53.26 GRP\_listAllClientsInGroup

**Description:** returns an array that consists of all client names that are in a group

**Parameter:**

- groupName: name of the group to check

### 8.53.27 GRP\_getDistrsAndSourcesLists

**Description:** writes the differnt distributions and package sources of the clients in a group as array to the both variables

**Parameter:**

- distrs: variable that should store the distributions
- sourceslists: variable that should store the sourceslist names

### 8.53.28 GRP\_showSelDistrSources

**Description:** shows a dialog for selection of distribution and package source name. The choices are taken from 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

### 8.53.29 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.

### 8.53.30 GRP\_HTMLBackToOverview

**Description:** generates HTML code to return to the group overview page

### 8.53.31 GRP\_getAllPackages

**Description:** shows a list of all packages on all clients in the selected groups. the packages can be selected by checkboxes

**Parameter:**

- groupNames: group names stores in an array
- key: keyword for searching for packages
- withActions: you can select to draw te action

### 8.53.32 GRP\_desasterRecovery

**Description:** recovers all selected clients and shows a message afterwards

**Parameter:**

- clients: an array containing all clients that should be recovered



## 8.54 *./inc/halfSister.php*

Author: Hauke Goos-Habermann (HHabermann@pc-kiel.de)

Description: Functions for halfSister.

### 8.54.1 *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.

### 8.54.2 *HS\_getClientSettingsCommand*

**Description:** Generates the commands to send the client infos to the server (This has the same functionality as *MSR\_getClientSettingsCommand*).

### 8.54.3 *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.

### 8.54.4 *HS\_ASSI\_prepareClient*

**Description:** Prepares a halfSister client for assimilation.

### 8.54.5 *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.

### 8.54.6 *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.

### 8.54.7 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.

### 8.54.8 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

### 8.54.9 HS\_fetchm23HSAdminAndm23hwscannerByOS

**Description:** Fetches the m23HSAdmin tool and m23hwscanner by detection of the OS.

### 8.54.10 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).

### 8.54.11 HS\_netConfig

**Description:** Sets IP, gateway, netmask, DNS and hostname.

**Parameter:**

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

### 8.54.12 HS\_setPackageProxy

**Description:** Sets the proxy for the package management tool.

**Parameter:**

- clientOptions: Associated array with the options of the client.

#### 8.54.13 HS\_setSourcesList

**Description:** Writes the package sources list for the client's package manager.

**Parameter:**

- clientName: name of the client

#### 8.54.14 HS\_normalUpdate

**Description:** Performs a normal update of the installed packages.

#### 8.54.15 HS\_pkgFullUpdate

**Description:** Performs a full update of the installed packages.

#### 8.54.16 HS\_sysSetm23ClientID

**Description:** Sets the m23 client ID.

**Parameter:**

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

#### 8.54.17 HS\_netEnableNTP

**Description:** Enable getting the system time by NTP.

#### 8.54.18 HS\_netDisableNTP

**Description:** Disable getting the system time by NTP.

#### 8.54.19 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.

#### 8.54.20 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.

#### 8.54.21 HS\_pkgInstallBasePackages

**Description:** Installs basic packages

#### 8.54.22 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).

#### 8.54.23 HS\_sysSetLanguage

**Description:** Sets the system language.

**Parameter:**

- lang: Two-character code of the language to set (de, en, fr).

#### 8.54.24 HS\_sysSetRootPW

**Description:** Sets the root password.

**Parameter:**

- Password: The crypted password of the user root.

#### 8.54.25 HS\_sysSetTimeZone

**Description:** Sets the time zone.

**Parameter:**

- timezone: Timezone in POSIX notation (e.g. Europe/Berlin)

#### 8.54.26 HS\_sysHWsetup

**Description:** Detects and configures new hardware

#### 8.54.27 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.

#### 8.54.28 HS\_sysChangeUser

**Description:** changes the settings of an useraccount on a client

**Parameter:**

- userName: the (old) username for the account
- password: the new unencrypted password for the account
- newUserName: the new username

#### 8.54.29 HS\_netEnableSSHDAndImportKey

**Description:** Enables the SSH daemon and adds a SSH key to let the m23 server log into the machine.

### 8.54.30 HS\_writeHosts

**Description:** Writes the /etc/hosts file for the client

### 8.54.31 HS\_sysWriteM23fetchjob

**Description:** Generates the m23fetchjob script and adds it to the init levels.

### 8.54.32 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.

### 8.54.33 HS\_sysInstallKernel

**Description:** Installs a matching kernel.

**Parameter:**

- kernelPkg: Name of the kernel package to install.

### 8.54.34 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)

### 8.54.35 HS\_netEnableLDAP

**Description:** Enables LDAP login for a client.

**Parameter:**

- clientOptions: the client's options array

### 8.54.36 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 associative array.

### 8.54.37 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.

### 8.54.38 HS\_sysConfigurePrinter

**Description:** Detects and configures printer(s).

#### 8.54.39 HS\_sysInstallPrinter

**Description:** Installs the printer software.

#### 8.54.40 HS\_pkgInstallKDE

**Description:** Installs KDE

#### 8.54.41 HS\_pkgInstallX

**Description:** Installs XOrg or another shipped X11 server.

#### 8.54.42 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.

#### 8.54.43 HS\_pkgInstall

**Description:** Installs one or more packages

**Parameter:**

- packages: List of packages to install seperated by spaces.

#### 8.54.44 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.

#### 8.54.45 HS\_pkgSearch

**Description:** Searches for available packages matching all keywords.

**Parameter:**

- searchTerms: Search terms seperated spaces.

#### 8.54.46 HS\_pkgInstallPreview

**Description:** Generates commands for getting a installation preview on the client.

**Parameter:**

- packages: List of packages to install seperated by spaces

#### 8.54.47 HS\_pkgDeinstallPreview

**Description:** Generates commands for getting a deinstallation preview on the client.

**Parameter:**

- packages: List of packages to deinstall seperated by spaces

#### 8.54.48 HS\_statusFileCommand

**Description:** Generates the commands to send the package infos to the server (This has the same functionality as MSR\_statusFileCommand).

#### 8.54.49 HS\_pkgUpdateCache

**Description:** Updates the list of available packages.

#### 8.54.50 HS\_pkgInstallGnome

**Description:** Installs Gnome.

#### 8.54.51 HS\_pkgInstallLXDE

**Description:** Installs LXDE.

#### 8.54.52 HS\_pkgInstallXFce

**Description:** Installs XFce.

#### 8.54.53 HS\_pkgDeinstall

**Description:** Deinstalls one or more packages.

**Parameter:**

- packages: List of packages to deinstall seperated by spaces.

#### 8.54.54 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.

#### 8.54.55 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.

#### 8.54.56 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

## 8.55 ./inc/helper.php

Author: Hauke Goos-Habermann (HHabermann@pc-kiel.de)

Description: Helper functions that did not fit into another include file.

### 8.55.1 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 numeric, the text will shown with the given HTML heading level.

### 8.55.2 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.

### 8.55.3 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

### 8.55.4 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.

### 8.55.5 HELPER\_URIencode

**Description:** Encodes a string like the JavaScript function URIencode would do it.

**Parameter:**

- in: Input string to be encoded.

**Returns:**

- Encoded version of the string



### 8.55.6 HELPER\_isUpper

**Description:** Checks, if a character is upper case

**Parameter:**

- char: Character to check.

**Returns:**

- true, when upper case otherwise false

### 8.55.7 HELPER\_filesize

**Description:** Gets the correct file size of a file, even if it is bigger than 2 GB.

**Parameter:**

- fileName: Name of the file with full path.

**Returns:**

- The file size of the file in bytes.

### 8.55.8 HELPER\_isExecutedInCLI

**Description:** Checks, if it is run in CLI.

**Returns:**

- True, when run in CLI otherwise false.

### 8.55.9 HELPER\_isExecutedOnUCS

**Description:** Checks, if it is run on UCS.

**Returns:**

- True, when run on UCS otherwise false.

### 8.55.10 HELPER\_getContentFromURL

**Description:** Downloads an URL via curl and gives back the site code.

**Parameter:**

- url: The URL to download.
- range: If set, a part of the file will be downloaded. (e.g. 0-500 will download the first 500 kb)

**Returns:**

- The downloaded site code or false in case of an error.

### 8.55.11 HELPER\_trimValue

**Description:** Runs trim on the input parameter and writes the result back.

**Parameter:**

- value: Value to trim.

### 8.55.12 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 indenting to start with.

**Returns:**

- Recursive calls of the BASH command with error checking.

### 8.55.13 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.

### 8.55.14 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 filtered version.

**Returns:**

- UTF8-encoded new lines of the log file.

### 8.55.15 HELPER\_rmRecursive

**Description:** Removes a directory with sub-directories and contained files.

**Parameter:**

- : dir: Full path to the directory.

### 8.55.16 HELPER\_showBAfH

**Description:** Shows the German BAFH excuse of the day.

### 8.55.17 HELPER\_ucrc32

**Description:** Returns the unsigned crc32 sum of an input value.

**Parameter:**

- : in: Input to crc.

**Returns:**

- Unsigned crc32 sum of an input value.

### 8.55.18 HELPER\_md5x5

**Description:** Hashes an input value 5 times with MD5.

**Parameter:**

- : in: Input to hash.

**Returns:**

- Hashed value.

### 8.55.19 HELPER\_netmaskAmountOfSetBits

**Description:** Calculates the amount of set bits in a network mask (as used in the short form of netmasks).

**Parameter:**

- : nm: The netmask in decimal notation.

**Returns:**

- Amount of set bits in the network mask.

### 8.55.20 HELPER\_networkCalculator

**Description:** Calculates the network IP by a given IP and the netmask.

**Parameter:**

- : ip: The IP.
- : nm: The netmask.

**Returns:**

- Network IPs

### 8.55.21 HELPER\_netmaskCalculator

**Description:** Converts a short netmask (e.g. 24 for 255.255.255.0) into the decimal notation.

**Parameter:**

- : nm: The netmask to convert. If a netmask in decimal notation is given, no conversation is done.

**Returns:**

- Netmask in decimal notation.

### 8.55.22 HELPER\_importAllIntoPOST

**Description:** Imports all values into the \$\_POST array in case that there are too much array keys for the normal processing.

### 8.55.23 HELPER\_randomUsername

**Description:** Generates a random username with a given length.

**Parameter:**

- length: Length of the username to create.

**Returns:**

- The random username.

#### 8.55.24 HELPER\_createSelfSignedCAAndServerCertificate

**Description:** Creates a selfsigned CA and a server certificate.

**Parameter:**

- CADn: Information about the CA.
- serverDn: Information about the server.
- password: Password for the private server key.
- expirationDate: Expiration of the certificates in days.

**Returns:**

- Associative array with the certificate of the CA, the certificate and private key of the server.

#### 8.55.25 HELPER\_arrayReOrderKeynumbers

**Description:** Changes all keys of the input array to simple ascending numbers, if the key of the input array is a number (if not, the key will be left unchanged). The order of the keys is preserved.

**Parameter:**

- inArray: The input array.

**Returns:**

- New array with the ascending key numbers.

#### 8.55.26 HELPER\_arrayInsertBeforeKeynumber

**Description:** Inserts a value into an array (that has simple numbers as keys) before a given key.

**Parameter:**

- inArray: The input array.
- beforeKeynumber: The key number the value should be inserted before.
- val: The value that should be inserted.

**Returns:**

- New array with the new value inserted.

#### 8.55.27 HELPER\_arrayInsertAfterKeynumber

**Description:** Inserts a value into an array (that has simple numbers as keys) after a given key.

**Parameter:**

- inArray: The input array.
- afterKeynumber: The key number the value should be inserted after.
- val: The value that should be inserted.

**Returns:**

- New array with the new value inserted.

**8.55.28 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 easily. (e.g [0] => Array([command] => format, [path] => /dev/md0, [fs] => ext4 ))

**8.55.29 HELPER\_array2m23Array****Description:** Converts a normal array to an m23 array.**Parameter:**

- array: A normal array, that may be edited more easily. (e.g [0] => Array([command] => format, [path] => /dev/md0, [fs] => ext4 ))

**Returns:**

- The m23 array, that is a 2D array, that consists of keys build of a parameter names combined with a parameter number. Parameter names with the same parameter number belong together. (e.g. [command0] => format, [path0] => /dev/md0, [fs0] => ext4, ...)

**8.55.30 print\_r2****Description:** Function like print\_r, but sorts the entries, if the input is an array and converts newlines to HTML breaks.**Parameter:**

- in: Value to print.

**8.55.31 HELPER\_debugBacktraceToFile****Description:** Writes/Appends debug information about all calling functions and parameters into a file.**Parameter:**

- file: File name with full path, where the debug information should be stored.

**8.55.32 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 empty.
- noProxy: Set to true, if the system proxy should not be used.

**Returns:**

- The contents of the files from cache or from download or false if no file could be found.

### 8.55.33 HELPER\_passGenerator

**Description:** Generates semi-random passwords via pwgen or DB\_genPassword.

**Parameter:**

- length: The length of the passwords.
- amount: The amount of passwords to generate.

**Returns:**

- Array with the generated passwords if \$amount > 1 or the password string directly if \$amount = 1.

### 8.55.34 HELPER\_array2AssociativeArray

**Description:** Copies the values of an array as keys AND values to a new associative array.

**Parameter:**

- in: Input array.

**Returns:**

- Associative array with equal keys and values.

### 8.55.35 HELPER\_randomMAC

**Description:** Generates a random MAC address.

**Returns:**

- Random MAC address in the format XX:XX:XX:XX:XX:XX (e.g. 70:c4:d4:49:6e:27).

### 8.55.36 HELPER\_generateSalt

**Description:** Generates a random salt string.

**Parameter:**

- length: Length of the salt.

**Returns:**

- Random salt of given length.

### 8.55.37 HELPER\_grubMd5Crypt

**Description:** Encrypts a password to the MD5 hash as expected by grub.

**Parameter:**

- password: Password to encrypt.
- length: Length of the salt.

**Returns:**

- Encrypted password in grub style or false if MD5 hash function isn't available.

### 8.55.38 HELPER\_listFilesInDir

**Description:** Lists all files in a directory and returns an array with all file names.

**Parameter:**

- dirname: Name of the directory.

**Returns:**

- Array with all file names.

### 8.55.39 HELPER\_getBootLoaders

**Description:** Returns a list of available bootloaders.

**Returns:**

- Array with available bootloaders.

### 8.55.40 HELPER\_getTimeZones

**Description:** Searches for all time zones.

**Parameter:**

- country: two letter country name that is used to select a timezone if none is set with \$first.

**Returns:**

- Array with all time zones.

### 8.55.41 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 truncated

### 8.55.42 HELPER\_grep

**Description:** returns all lines from \$string seperated by \$cut that contain \$search

**Parameter:**

- string: the text, that should be searched
- search: the string to be searched
- cut: seperator for the input and output lines

**Returns:**

- The found lines as string separated by \$cut.

### 8.55.43 HELPER\_grepNot

**Description:** Returns all lines from \$string seperated by \$cut that do NOT contain \$search.

**Parameter:**

- string: the text, that should be searched
- search: the string to be searched
- cut: seperator for the input and output lines

**Returns:**

- The found lines as string separated by \$cut.

#### 8.55.44 HELPER\_grepCount

**Description:** Counts the lines from \$string separated by \$cut that contain \$search.

**Parameter:**

- string: the text, that should be searched
- search: the string to be searched
- cut: separator for the input and output lines

**Returns:**

- Amount of lines that match the \$search.

#### 8.55.45 HELPER\_getFdiskMountPoints

**Description:** Returns an array with all mount points listed in /etc/fstab

**Parameter:**

- excludeExtra: set to true, if you want to exclude /proc and /sys from the array

**Returns:**

- Found mount points as array keys and values.

#### 8.55.46 HELPER\_getApacheUser

**Description:** returns the name of the Apache user

#### 8.55.47 HELPER\_getApacheGroup

**Description:** returns the group of the Apache user

#### 8.55.48 HELPER\_putFileContents

**Description:** Writes data to a file.

**Parameter:**

- fileName: name of the file to write
- contents: Text or data that should be written to the file.

**Returns:**

- Error code from fwrite.

#### 8.55.49 HELPER\_getFileContents

**Description:** returns the contents of a file (the file is read to a maximum of 5 MB)

**Parameter:**

- fileName: name of the file to read

#### 8.55.50 HELPER\_showFileContents

**Description:** Shows the contents of a file (the file is read to a maximum of 5 MB)

**Parameter:**

- fileName: name of the file to read



**8.55.51 HELPER\_maxPhpUploadSize**

**Description:** Returns the maximum file upload size allowed by php.ini.

**8.55.52 HELPER\_compareLengthAbc**

**Description:** Compares the length of two strings and then by alphabet

**Parameter:**

- \$a: string of a certain length
- \$b: string of a certain length

**8.55.53 HELPER\_sortByLength**

**Description:** Sorts an array by length of its values, shortest value first, keeping key-value pairs

**Parameter:**

- \$array: The array you want to sort by length

**8.55.54 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.

## 8.56 `./inc/help.php`

Author: Hauke Goos-Habermann (HHabermann@pc-kiel.de)

Description: function to show the help box in the correct language

### 8.56.1 `HELP_showHelp`

**Description:** shows the help block for the online help

**Parameter:**

- topic: the name of the help file
- language: two character language description (e.g. de, en, fr,...)

### 8.56.2 `HELP_getHelp`

**Description:** Returns the help block for the online help

**Parameter:**

- topic: the name of the help file or name of a man page starting with "man://" e.g. man://tar
- language: two character language description (e.g. de, en, fr,...)
- fileName: full path to a help file in a directory with language short name

**Returns:**

- help block string

### 8.56.3 `HELP_getHelpString`

**Description:** returns the help block for the online help

**Parameter:**

- topic: the name of the help file
- language: two character language description (e.g. de, en, fr,...)

### 8.56.4 `HELP_showHelpTex`

**Description:** shows the help file converted to LaTeX code

**Parameter:**

- fileName: name of the help file

## 8.57 *./inc/html.php*

Author: Hauke Goos-Habermann (HHabermann@pc-kiel.de)

Description: functions for generating often used HTML code

### 8.57.1 *HTML\_waitAnimation*

**Description:** Defines HTML code for showing a waiting animation and onClick code for inserting 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.

### 8.57.2 *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 if its state is "on"
- \$off\_text: Text to be displayed if state is "off"
- \$on\_text: Text to be displayed if state is "on"
- \$separator: Anything which shall be displayed between the picture (clickable) and the text (not clickable)
- \$default: State of the image input element on first load of page ("on" or "off")
- \$outState: Current state of element (true for "on" or false for "off").

**Returns:**

- true, if the button was clicked otherwise false.

### 8.57.3 *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.

### 8.57.4 HTML\_uploadFile

**Description:** Shows a dialog for uploading image files.

**Parameter:**

- `htmlName`: Name of the HTML element
- `label`: The visual naming of the HTML element.
- `maxFileSize`: The maximum allowed filesize in bytes.

**Returns:**

- The full path to the uploaded file or false in case of an error.

### 8.57.5 HTML\_urlButton

**Description:** Defines a link that appears like a button.

**Parameter:**

- `htmlNames`: Name of the constant.
- `label`: Label of the button.
- `url`: The URL where the link button should point to.

### 8.57.6 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:**

- Chosen weekday and hour/minute as DHHMM string.

### 8.57.7 HTML\_sourceViewer

**Description:** Creates a source code viewer area with syntax highlighting.

**Parameter:**

- `htmlNames`: Name of the source viewer.
- `code`: The source code to show.
- `highlighting`: The GeSHi language of the source code (e.g. bash).

### 8.57.8 HTML\_getOutputBuffer

**Description:** Gets the complete (previously rendered) HTML output buffer.

**Returns:**

- The complete contents of the (previously rendered) HTML output buffer.

### 8.57.9 HTML\_setOutputBuffer

**Description:** Sets (replaces) the complete (previously rendered) HTML output buffer, that will be sent to the webbrowser.

**Parameter:**

- `HTMLOutputBuffer`: New HTML codes that should replace the complete current output buffer.

### 8.57.10 HTML\_AJAXAutoSubmit

**Description:** Defines AJAX code that clicks a submit button when the given URL returns 'submit'. The defined constant should be used as LAST part of the \$extra parameter in the HTML\_submit function.

**Parameter:**

- htmlNames: Name of the HTML submit element (button).
- url: The HTTP URL to poll.
- timeout: Time in milliseconds to poll the URL for new status.

**Returns:**

- Constant name to insert into \$extra.

### 8.57.11 HTML\_liveSpan

**Description:** Creates a span that updates itself via AJAX by polling from a given URL.

**Parameter:**

- htmlNames: Name of the span.
- url: The HTTP URL to poll.
- staticValue: Value that should be shown, when AJAX is not available (e.g. when JavaScript is disabled)
- timeout: Time in milliseconds to poll the URL for new status.

### 8.57.12 HTML\_manipulateOutputBuffer

**Description:** Manipulates the output buffer with already generated HTML code and replaces all occurrences the search term with the replace term.

**Parameter:**

- search: The search term.
- replace: The replace text.

**Returns:**

- true, if the search term was found.

### 8.57.13 HTML\_showTitle

**Description:** Shows a title.

**Parameter:**

- title: Text of the title.

### 8.57.14 HTML\_showSmallTitle

**Description:** Shows a title of the second tier.

**Parameter:**

- title: Text of the small title.

### 8.57.15 HTML\_hiddenVar

**Description:** Create a hidden HTML variable to store values in an HTML form.

**Parameter:**

- var: Name of the hidden variable.
- val: Its value.

### 8.57.16 HTML\_liveLogArea

**Description:** Creates a log area that updates itself via AJAX by polling from a given URL.

**Parameter:**

- htmlNames: Name of the log area.
- width: The width in characters of the log area.
- height: The height in characters of the log area.
- url: The HTTP URL to poll.
- timeout: Time in milliseconds to poll the URL for new status.
- maxLines: Maximum amount of lines to show in the log area (older lines are removed, when there are too many).

### 8.57.17 HTML\_checkboxChangerButtons

**Description:** Defines buttons for changing all check boxes.

**Parameter:**

- htmlNames: Name of the HTML elements.

### 8.57.18 HTML\_jsCheckboxChanger

**Description:** Generates JavaScript code for changing all check boxes.

**Parameter:**

- jsBlockName: Name of the JS block constant.

### 8.57.19 HTML\_logArea

**Description:** Shows a text area for log information (readonly).

**Parameter:**

- htmlName: Name of the HTML element.
- cols: Number of columns.
- rows: Number of rows to show.
- text: The log information to show.

### 8.57.20 HTML\_getQuestionnaireURL

**Description:** Returns the complete URL to the m23 questionnaire in the language of the webinterface.

**Returns:**

- Complete URL to the m23 questionnaire in the language of the webinterface.

### 8.57.21 HTML\_questionnaire

**Description:** Shows the questionnaire window.

**Parameter:**

- disable: Set to true to disable showing of the window again.

### 8.57.22 HTML\_esel

**Description:** Shows an dog-ear that can be opened to show "goos-habermann.de/m23ad".

### 8.57.23 HTML\_JSMenuCloseAllEntries

**Description:** Closes all menu entries for a menu. This should be called at the end of a page to get it executed after loading.

### 8.57.24 HTML\_JSMenuOpener

**Description:** Opens a menu entry when moving the mouse over the title and closes all other entries of the same menu.

**Parameter:**

- menuName: Name of the complete menu. This name must be the same on all entries belonging to the same menu.
- entryName: Name of the entry. This name must be unique.
- title: Title for the menu entry.
- html: HTML code of the menu entry. Here can stand all that is expressable with HTML. This part is shown and hidden.
- titleCSS: CSS class for marking the title.

**Returns:**

- The HTML code for displaying the menu entry.

### 8.57.25 HTML\_jQueryMenu

**Description:** Creates an entry for the jQuery accordion menu

**Parameter:**

- title: Title for the menu entry.
- html: HTML code of the menu entry. Here can stand all that is expressable with HTML. This part is shown and hidden.
- titleCSS: CSS class for marking the title.

**Returns:**

- The jQuery code for the menu entry.

### 8.57.26 HTML\_jQueryMenuHeader

**Description:** Generate code for beginning a the jQuery accordion menu.

**Parameter:**

- menuName: Name of the menu.

**Returns:**

- The jQuery code for beginning the menu.

### 8.57.27 HTML\_jQueryReStoreYWindowPosition

**Description:** Generates jQuery code for storing the Y scroll position of the window and to restore the position after a submit.

**Parameter:**

- variablePrefix: Prefix for the hidden variable with the Y position.
- hiddenPosCode: Variable where the hidden variable HTML code is written to.

**Returns:**

- The jQuery code for storing the Y scroll position of the window and to restore the position after a submit.

### 8.57.28 HTML\_jQueryMenuEnd

**Description:** Generates code for ending a the jQuery accordion menu.

**Parameter:**

- menuName: Name of the menu.

**Returns:**

- The jQuery code for ending the menu.

### 8.57.29 HTML\_incStatusBarPercentByName

**Description:** Increments the status bar percent by a given amount.

**Parameter:**

- statusBarName: The name of the status bar.
- client: The name of the client, the status bar belongs to (or other values for identifying the object the status bar belongs to)
- percent: Percent value of the current job.

### 8.57.30 HTML\_setStatusBarPercentPointByName

**Description:** Calculates the value of a percent point according to the amount of waiting packages and stores the result in the DB.

**Parameter:**

- statusBarName: The name of the status bar.
- client: The name of the client, the status bar belongs to (or other values for identifying the object the status bar belongs to)
- recalculate: true, if the remaining percent value of the status bar should be used to calculate a new (better fitting) percentpoint.

**Returns:**

- false on errors, otherwise true.



### 8.57.31 HTML\_setStatusBarStatusByName

**Description:** Sets new percent value and/or new status text by clientname AND status bar name.

**Parameter:**

- statusBarName: The name of the status bar.
- client: The name of the client, the status bar belongs to (or other values for identifying the object the status bar belongs to)
- percent: Percent value to write into the DB (may be false, if it should not be changed).
- statustext: A text message that should be shown under the status bar and written to the DB (may be false, if it should not be changed).

**Returns:**

- : false on parameter error.

### 8.57.32 HTML\_setStatusBarStatusByID

**Description:** Sets new percent value and/or new status text by status bar ID.

**Parameter:**

- id: ID of the status bar
- percent: Percent value to write into the DB (may be false, if it should not be changed).
- statustext: A text message that should be shown under the status bar and written to the DB (may be false, if it should not be changed).

**Returns:**

- : false on parameter error.

### 8.57.33 HTML\_setStatusBarStatus

**Description:** Sets new percent value and/or new status text by status bar ID or clientname AND status bar name.

**Parameter:**

- id: ID of the status bar (Optional parameter to set values of status bar with given ID).
- percent: Percent value to write into the DB (may be false, if it should not be changed).
- statustext: A text message that should be shown under the status bar and written to the DB (may be false, if it should not be changed).
- statusBarName: The name of the status bar.
- client: The name of the client, the status bar belongs to (or other values for identifying the object the status bar belongs to)

**Returns:**

- : false on parameter error.

### 8.57.34 HTML\_getStatusBarID

**Description:** Returns the status bar ID of the searched status bar.

**Parameter:**

- name: The name of the status bar.
- client: The name of the client, the status bar belongs to (or other values for identifying the object the status bar belongs to)

**Returns:**

- : The status bar ID of the searched status bar or false if none could be found.

### 8.57.35 HTML\_newStatusBar

**Description:** Shows the iframe for a status bar. This actually displays the status bar.

**Parameter:**

- name: The name of the status bar.
- client: The name of the client, the status bar belongs to (or other values for identifying the object the status bar belongs to)
- type: The method of calculating/getting the percentage to display in the status bar.
- cmd: BASH command, if type is STATUSBAR\_TYPE\_bash.
- refreshtime: Time (in seconds) between refreshes of the status bar.
- statustext: A text message that should be shown under the status bar.
- percent: Percent value to write into the DB.

**Returns:**

- : The status bar ID of the just created status bar or false, if it could not be created.

### 8.57.36 HTML\_showStatusBar

**Description:** Shows the iframe for a status bar. This actually displays the status bar.

**Parameter:**

- id: ID of the status bar.
- width: The width of the status bar iframe.
- height: The height of the status bar iframe.

### 8.57.37 HTML\_showStatusBarHTML

**Description:** Shows the status bar, that is drawn in the iframe (this function is only called by statusBar.php).

**Parameter:**

- id: ID of the status bar.

### 8.57.38 HTML\_multiCheckBox

**Description:** Defines a list of checkboxes, that represent a value each. The values of checked checkboxes are stored in an array and returned.

**Parameter:**

- `htmlName`: Name of the HTML element.
- `valuesLabels`: Array with the values and labels for the checkboxes.
- `defaultChecked`: Array with values that are checked by default.
- `forceReload`: Set to true if the check box should be set to the state of `$defaultCheck` in any case.

**Returns:**

- : Array with the values of all checked checkboxes.

### 8.57.39 HTML\_multiCheckBoxShow

**Description:** Shows a list of checkboxes, that represent a value each. The values of checked checkboxes are stored in an array and returned.

**Parameter:**

- `valuesLabels`: Array with the values and labels for the checkboxes.
- `defaultChecked`: Array with values that are checked by default.
- `forceReload`: Set to true if the check box should be set to the state of `$defaultCheck` in any case.

**Returns:**

- : Array with the values of all checked checkboxes.

### 8.57.40 HTML\_setPage

**Description:** Sets the m23 page as hidden value.

**Parameter:**

- `page`: Name of the page.

### 8.57.41 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.

### 8.57.42 HTML\_storableSelection

**Description:** Shows a list of radio buttons or a selection with loading and storing the checking state to and from the session.

**Parameter:**

- **htmlName:** Name of the HTML element.
- **prefKey:** Variable name of the preference the dialog element stands for.
- **array:** An array that hold the returned values (array keys) the naming for the elements (array values).
- **type:** SELTYPE\_selection for a selection or SELTYPE\_radio for radio buttons.
- **vertical:** Set to true if the radio buttons should be aligned vertically or to false for horizontal aligning. This parameter is ignored by selections.
- **defaultSelection:** The value of the item to select by default.
- **storePointer:** Additional pointer to the variable where to store the entered value.
- **js:** Here can JavaScript or other parameters be added.

**Returns:**

- true if the check box is checked.

### 8.57.43 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.

#### 8.57.44 HTML\_storableCheckBox

**Description:** Shows a check box with label with loading and storing the checking state to and from the session.

**Parameter:**

- **htmlName:** Name of the HTML element.
- **label:** Label of the element.
- **prefKey:** Variable name of the preference the dialog element stands for.
- **defaultCheck:** Set to true if the check box should be checked if no HTML value is given.
- **storePointer:** Additional pointer to the variable where to store the entered value.
- **checkedValue:** The value that should be stored into \$storePointer if the check box is checked.
- **uncheckedValue:** The value that should be stored into \$storePointer if the check box is NOT checked.

**Returns:**

- true if the check box is checked.

#### 8.57.45 HTML\_getElementValue

**Description:** Gets the value for a HTML element by the session data or POST value.

**Parameter:**

- **htmlName:** Name of the HTML element.
- **prefKey:** Variable name of the preference the dialog element stands for.
- **initValue:** The initial value if the element is shown first.

**Returns:**

- Returns the default value, the session value or false.

#### 8.57.46 HTML\_listSelection

**Description:** shows a selection with options stored in an array

**Parameter:**

- **selName:** name of the selection
- **list:** array with the entries. The array can be a simple numeric array or an associative array with discrete entries for the shown name and the value. e.g. : \$l[name0]="public"; \$l[val0]="internal"; \$l[name1]="public1"; \$l[val1]="internal1"; public and public1 will be shown the user in the web-browser, while internal and internal1 are the values that are transfered to the server.
- **first:** entry that should be shown first (this is the internal value and NOT the name shown to the user). the first value from the list will be written to \$first. set first to "false" to disable writing the first entry.
- **firstName:** if you want to use the associative array variant and a first value, you need to set the name that should be shown to the user. This name is stored in firstName

#### 8.57.47 HTML\_showTableHeader

**Description:** prints the header of a shadowed table

**Parameter:**

- **centerTable:** set to true if the table should be centered vertically
- **tableStyle:** CSS class of the inner table.
- **width:** Width of the table.

### 8.57.48 HTML\_showTableEnd

**Description:** prints the end of a shadowed table

**Parameter:**

- centerTable: set to true if the table should be centered vertically

### 8.57.49 HTML\_showFormHeader

**Description:** Shows the header of a formular

**Parameter:**

- addAction: set it, if additional parameters to index.php should be used
- method: default is POST, but you can set it to GET

### 8.57.50 HTML\_showFormEnd

**Description:** Shows the end of a formular

### 8.57.51 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.

### 8.57.52 HTML\_submitImg

**Description:** Defines a graphical submit button.

**Parameter:**

- htmlName: Name of the HTML element.
- img: Name of the image to show.
- alt: Alternative text to show when no images can be shown.
- extra: Extra options for the HTML input tag.

**Returns:**

- True if it was clicked otherwise false.

### 8.57.53 **HTML\_input**

**Description:** HTML text or password edit line.

**Parameter:**

- **htmlName:** Name of the HTML element.
- **htmlValue:** The default text to show in the edit line if nothing was submitted.
- **size:** Size (in characters) of the input line.
- **maxlength:** The maximum length of the entered text.
- **type:** Type of the edit line (INPUT\_TYPE\_text or INPUT\_TYPE\_password)
- Returns the entered value, the default value or false.

### 8.57.54 **array\_makeFirst**

**Description:** special sort function that makes a special element the first element and leaves the other elements in its previous order.

**Parameter:**

- **arr:** Array to sort
- **first:** Value of the element that should be put on top

### 8.57.55 **HTML\_getValidSelected**

**Description:** Checks for a valid selected value from a list of possible values. In case the value could not be found, a default value is taken.

**Parameter:**

- **selected:** Array or single value to check, if it is on the list aof array keys.
- **arrayKeys:** An array that holds the possible returned values (array keys).
- **defaultSelection:** The value of the item to select by default.

**Returns:**

- A valid value from a list of possible values.

### 8.57.56 **HTML\_selection**

**Description:** Shows a list of radio buttons or a selection.

**Parameter:**

- **htmlName:** Name of the HTML element.
- **array:** An array that hold the returned values (array keys) the naming for the elements (array values).
- **type:** SELTYPE\_selection for a selection or SELTYPE\_radio for radio buttons.
- **vertical:** Set to true if the radio buttons should be aligned vertically or to false for horizontal aligning. This parameter is ignored by selections.
- **defaultSelection:** The value of the item to select by default.
- **prefKey:** Variable name of the preference the dialog element stands for.
- **js:** Here can JavaScript or other parameters be added.
- **multipleSize:** If set to a number (and not to false) a multi selection is generated, where the user can select multiple entries. The number sets the amount of entries to show the user.

**Returns:**

- The value of the selected element or false if nothing was selected.

### 8.57.57 HTML\_checkBox

**Description:** Shows a check box with label.

**Parameter:**

- `htmlName`: Name of the HTML element.
- `label`: Label of the element.
- `defaultCheck`: Set to true if the check box should be checked if no HTML value is given.
- `prefKey`: Variable name of the preference the dialog element stands for.
- `htmlValue`: Value of the checkbox if clicked.
- `forceReload`: Set to true if the check box should be set to the state of `$defaultCheck` in any case.

**Returns:**

- True if the check box is checked.

### 8.57.58 HTML\_checkBoxCheckAll

**Description:** Generates an array with all checked checkboxes. It assumes that value of a checked checkbox is 1.

**Parameter:**

- `filter`: Filter to get only the POST elements which begin with the filter string.

**Returns:**

- Array with all checked checkboxes.

### 8.57.59 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.

### 8.57.60 HTML\_submitCheck

**Description:** Checks if a previously defined button was clicked.

**Parameter:**

- `htmlName`: Name of the HTML element.

**Returns:**

- True if the button was clicked.

### 8.57.61 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 parameters in each call.

**Parameter:**

- Arbitrary amount of cells to show in a table.



### 8.57.62 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 parameters in each call.

**Parameter:**

- Arbitrary amount of cells to show in a table.

### 8.57.63 HTML\_textArea

**Description:** Shows a text area to insert text.

**Parameter:**

- htmlName: Name of the HTML element.
- cols: Number of columns.
- rows: Number of rows to show.
- default: Text to show by default.

**Returns:**

- : The entered text.

### 8.57.64 HTML\_showPagePrintButton

**Description:** Shows a print button that allows easy printing of the current m23 administration interface.

## 8.58 ./inc/hwinfo.php

Author: Daniel Kasten (DKasten@pc-kiel.de) ,Hauke Goos-Habermann (HHabermann@pc-kiel.de)

Description: functions to view the hardware information of the client.

### 8.58.1 HWINFO\_getParam

**Description:** get hardware infos

**Parameter:**

- paramName: parameter to read from the hardware info (e.g. cpu, mem, ...)
- clientName: name of the client

### 8.58.2 HWINFO\_getMemory

**Description:** returns the size of memory

**Parameter:**

- clientName: name of the client

### 8.58.3 HWINFO\_getHDSize

**Description:** Returns the sizes of all harddisks in a string, sperated by html breaks

**Parameter:**

- clientName: name of the client

### 8.58.4 HWINFO\_printPartitions

**Description:** prints the partition information

**Parameter:**

- clientName: name of the client

### 8.58.5 DMI\_getAllTextBox

**Description:** Get all DMI info in a text box.

**Parameter:**

- clientName: name of the client

**Returns:**

- All DMI info in a text box.

### 8.58.6 DMI\_getParam

**Description:** get dmi info for a special parameter

**Parameter:**

- paramName: name of dmi setting
- clientName: name of the client

### 8.58.7 DMI\_getBoard

**Description:** get the dmi board informations

**Parameter:**

- clientName: name of the client

### 8.58.8 DMI\_getMemory

**Description:** get the dmi memory informations

**Parameter:**

- clientName: name of the client

### 8.58.9 DMI\_getCPU

**Description:** get the dmi cpu informations

**Parameter:**

- clientName: name of the client

### 8.58.10 DMI\_getSlot

**Description:** get the dmi information about slots

**Parameter:**

- clientName: name of the client

## 8.59 ./inc/i18n.php

Author: Hauke Goos-Habermann (HHabermann@pc-kiel.de)

Description: functions for multi language handling in m23.

### 8.59.1 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.

### 8.59.2 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.

### 8.59.3 I18N\_number\_format

**Description:** Converts numbers to the language specific number formatting.

**Parameter:**

- in: Input number.

**Returns:**

- Language specific number formatted number.

### 8.59.4 I18N\_convertToHumanReadableName

**Description:** Converts a short language into a human readable name.

**Parameter:**

- lang: Two letter TLD (or longer code for countries that have more than a language (e.g. be-nl, be-fr))

**Returns:**

- Language in human readable notation.

### 8.59.5 I18N\_m23instLanguage

**Description:** Checks if a m23inst.php exists for the given language and returns "en" if not.

**Parameter:**

- shortLanguage: Two letter TLD (or longer code for countries that have more than a language (e.g. be-nl, be-fr))

**Returns:**

- Language code with available m23inst.php file.

### 8.59.6 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 keyboard setting.
- kdekeyboard: Language setting for the KDE keyboard.
- locale: List of locales (seperated by newlines) for the locale tool.
- kdekeyboards: List of additional KDE keyboards (seperated by commata).
- timezone: The timezone matching the language.
- packagelang: Language suffix that is added to the packages (e.g. language-pack-gnome-XX)

### 8.59.7 I18N\_listClientLanguages

**Description:** Lists all languages, the m23 clients can be installed with.

**Parameter:**

- default: the language that should be shown first
- directOutputtedSelection: Set to true, if the selection should be shown instead of returned.

### 8.59.8 I18N\_countCachedLanguages

**Description:** Counts the cached languages for a type.

**Parameter:**

- webinterface: Set to true, if it is a webinterface language. Set to false, for marking a client language.

### 8.59.9 I18N\_cacheWebinterfaceLanguages

**Description:** Caches the language information from the language.info files to the DB.

### 8.59.10 I18N\_getAllCachedLanguages

**Description:** Returns an associative array with the shortnames of the language as keys and the longnames as value.

**Parameter:**

- webinterface: Set to true, if it is a webinterface language. Set to false, for marking a client language.

### 8.59.11 I18N\_listWebinterfaceLanguages

**Description:** Lists all languages, the m23 webinterface is available in, as option lines

**Parameter:**

- default: the language that should be shown first
- directOutputtedSelection: Set to true, if the selection should be shown instead of returned.

### 8.59.12 I18N\_addClientLanguageToCache

**Description:** Adds a new client language to the i18n DB cache.

**Parameter:**

- shortLanguage: Two letter TLD (or longer code for countries that have more than a language (e.g. be-nl, be-fr))
- longLanguage: Long human readable country/language name.
- in: Associative array with the information for the language.

### 8.59.13 I18N\_cacheClientLanguages

**Description:** Caches the client languages in the DB.

### 8.59.14 I18N\_getLangVars

**Description:** Returns an associative array with language settings for the client.

**Parameter:**

- shortLanguage: Two letter TLD (or longer code for countries that have more than a language (e.g. be-nl, be-fr))

## 8.60 ./inc/ldap.php

Author: Hauke Goos-Habermann (HHabermann@pc-kiel.de)

Description: Functions for using a LDAP server

### 8.60.1 LDAP\_getTypes

**Description:** Returns an array with the known LDAP types.

**Returns:**

- Array with the known LDAP types.

### 8.60.2 LDAP\_connectServer

**Description:** Connects with read/write access to the LDAP server on the m23 server.

**Parameter:**

- name: name of the LDAP server stored in the configuration file

### 8.60.3 LDAP\_makeConnection

**Description:** Connects to a LDAP server.

**Parameter:**

- host: hostname or IP of the LDAP server
- baseDN: baseDN for the LDAP server
- pwd: password for the administrator or empty for anonymous access

### 8.60.4 LDAP\_getValueFromConfigLine

**Description:** Returns the value from a phpLDAPAdmin configuration line.

**Returns:**

- Value from a phpLDAPAdmin configuration line.

### 8.60.5 LDAP\_listServers

**Description:** Returns an associative array with the LDAP server names as keys and values.

**Returns:**

- Associative array with the LDAP server names as keys and values.

### 8.60.6 LDAP\_loadServer

**Description:** Loads the variables from a LDAP server.

**Parameter:**

- name: server name

### 8.60.7 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.

**8.60.8 LDAP\_fqdn2dn**

**Description:** Returns the DN converted from a FQDN

**Parameter:**

- domain: a full qualified domain name (e.g. test.m23.de)

**8.60.9 LDAP\_installServer**

**Description:** Generates a script that installs and configures an openLDAP server

**Parameter:**

- host: the IP or hostname of the LDAP server
- org: name of the organisation
- domain: the DN (e.g. foo.m23.de)
- pwd: the unencrypted password for the admin

**8.60.10 LDAP\_addServerTophpLdapAdmin**

**Description:** Adds an LDAP server to the phpLDAPadmin configuration file.

**Parameter:**

- name: how the LDAP server should be called
- host: the IP or hostname of the LDAP server
- base: the base DN (e.g. dc=m23, dc=de)
- pwd: the unencrypted password for the admin

**8.60.11 LDAP\_delServerFromphpLdapAdmin**

**Description:** Deletes a LDAP server from the phpLDAPadmin configuration file.

**Parameter:**

- name: the name of the LDAP server that should be deleted

**8.60.12 LDAP\_checkphpLdapAdminConfiguration**

**Description:** Checks if the phpLDAPadmin configuration file is existing and creates it if it's missing

**8.60.13 LDAP\_showServerManagementDialog**

**Description:** Shows a dialog for adding, removing and changing LDAP servers.



#### 8.60.14 LDAP\_I18NLdapType

**Description:** Returns the human readable description of the LDAP usage type

**Parameter:**

- status: status string

#### 8.60.15 LDAP\_getNextID

**Description:** Returns the next higher than the highest ID

**Parameter:**

- type: "user" for user IDs

#### 8.60.16 LDAP\_getNextUserID

**Description:** Returns the next higher than the highest user ID

#### 8.60.17 LDAP\_addNewID

**Description:** Adds a new ID to the table of used IDs or returns "false" if the ID exists

**Parameter:**

- type: "user" for user IDs
- id: number of the new ID

#### 8.60.18 LDAP\_deleteID

**Description:** Deletes an ID from the table of used IDs

**Parameter:**

- type: "user" for user IDs
- id: number of the new ID

#### 8.60.19 LDAP\_addNewUserID

**Description:** Adds a new user ID to the table

**Parameter:**

- id: number of the new ID

#### 8.60.20 LDAP\_addNewGroupID

**Description:** Adds a new user ID to the table

**Parameter:**

- id: number of the new ID

#### 8.60.21 LDAP\_getNextGroupID

**Description:** Returns the next higher than the highest group ID

### 8.60.22 LDAP\_getFreeIDs

**Description:** Returns an array with free IDs of the selected type.

**Parameter:**

- type: "user" for user IDs
- start: start ID to check if it's free
- amount: the amount of IDs to return

### 8.60.23 LDAP\_getFreeUserIDs

**Description:** Returns an array with free user IDs of the selected type.

**Parameter:**

- start: start ID to check if it's free
- amount: the amount of IDs to return

### 8.60.24 LDAP\_getFreeGroupIDs

**Description:** Returns an array with free group IDs of the selected type.

**Parameter:**

- start: start ID to check if it's free
- amount: the amount of IDs to return

### 8.60.25 LDAP\_matchLDAPserver

**Description:** Searches for the name of a LDAP server and returns the name of the found server or false

**Parameter:**

- host: the IP or hostname of the LDAP server
- base: the base DN (e.g. dc=m23, dc=de)

## 8.61 `./inc/m23shared/m23shared.php`

Author: Hauke Goos-Habermann (HHabermann@goos-habermann.de)

Description: database functions, open, close the database, get ip of the calling client

### 8.61.1 `m23SHARED_deleteBills`

**Description:** Deletes all bills.

### 8.61.2 `m23SHARED_deleteUserdataFromDB`

**Description:** Deletes user data not needed for archiving purpose.

### 8.61.3 `m23SHARED_isMarkedForDeletion`

**Description:** Checks if the current user is marked for deletion.

**Returns:**

- True, if the user should be deleted on the next call or false if not.

### 8.61.4 `m23SHARED_markForDeletion`

**Description:** Marks the current user's data for deletion and disable his login.

### 8.61.5 `m23SHARED_DBname`

**Description:** Generates a database name for m23 shared by input and salt.

**Parameter:**

- adminName: Name of the m23shared admin

**Returns:**

- Database name.

### 8.61.6 `m23SHARED_getDBnameByInterface`

**Description:** Calculates the database name for m23 shared by setting in the user interface

**Returns:**

- Database name or false if there is no user logged in.

### 8.61.7 `m23SHARED_getDBnameByClient`

**Description:** Gets the database name for m23 shared as part of the client ID.

**Returns:**

- Database name or false if there is no DB name in the ID.

### 8.61.8 `m23SHARED_init`

**Description:** Checks if a m23shared environment is present, sets the variables in the session and chooses the according DB.

**Returns:**

- True if an m23 m23shared environment is present, false on error or no m23shared.

### 8.61.9 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 successfully, otherwise false.

### 8.61.10 m23SHARED\_setLicense

**Description:** Sets the maximum allowed amount of clients and the evaluation time in days.

**Parameter:**

- payType: Payment type (M23SHARED\_PAYTYPE\_EVAL, M23SHARED\_PAYTYPE\_PROFESSIONAL)
- clientAmount: Amount of clients that can be manages with the license.
- evaluationDays: Days for evaluation time.

### 8.61.11 m23SHARED\_evaluationDaysLeft

**Description:** Calculates how many days are left from evaluation period.

**Returns:**

- : Amount of days left (can be negative too if the evaluation time is over)

### 8.61.12 m23SHARED\_evaluationEndDate

**Description:** Generates a string with the end date and time in current selected language of the evaluation period.

**Returns:**

- : String with the end date and time in current selected language of the evaluation period.

### 8.61.13 m23SHARED\_paidClients

**Description:** Calculates and returns the amount of clients the customer has paid for.

**Returns:**

- : Amount of clients the customer has paid for.

### 8.61.14 m23SHARED\_getCompleteClientName

**Description:** Returns the complete name of a m23 shared client (\$\_SESSION variables need to be set).

**Parameter:**

- : clientName: Name of the client (if the value from the session should not/cannot be taken)

**Returns:**

- : Client name of a m23 shared client (e.g. m23SrGH1ikdPZ8.test)

### 8.61.15 m23SHARED\_getServerIP

**Description:** Returns the FQDN of the m23shared server.

**Returns:**

- : FQDN of the m23shared server.

#### 8.61.16 m23SHARED\_getPayTypeArray

**Description:** Returns an array with the numeric 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 numeric pay types and human readable translations.

#### 8.61.17 m23SHARED\_getSalutationsArray

**Description:** Returns an array with salutation forms.

**Returns:**

- : Array with gender character as key and localised salutation.

#### 8.61.18 m23SHARED\_getSalutationHeadline

**Description:** Returns a salutation headline that can be used as beginning of an email.

**Returns:**

- : Salutation headline in current language.

#### 8.61.19 m23SHARED\_getPayTypeHumanReadable

**Description:** Returns the license and payment type of the m23@web account as human readable string.

**Returns:**

- : Human readable string with license and payment type of the m23@web account.

#### 8.61.20 m23SHARED\_getMonthlyFee

**Description:** Calculates the monthly fee with the given amount of paid clients.

**Returns:**

- : Monthly price.

#### 8.61.21 m23SHARED\_getCustomerEmail

**Description:** Returns the eMail address of customer.

**Returns:**

- : eMail address of customer.

#### 8.61.22 m23SHARED\_setCustomerEmail

**Description:** Sets the eMail address of customer.

#### 8.61.23 m23SHARED\_generateActivationKey

**Description:** Calculates a random activation key and stores it into the DB.

**Returns:**

- : Random activation key.

#### 8.61.24 m23SHARED\_getActivationKey

**Description:** Returns the activation key for this customer.

**Returns:**

- : Activation key.

#### 8.61.25 m23SHARED\_generateCustomerNumber

**Description:** Calculates the customer number by current time and random value and stores it into the DB.

**Returns:**

- : Customer number.

#### 8.61.26 m23SHARED\_getCustomerNr

**Description:** Returns the customer number for this customer.

**Returns:**

- : Customer number.

#### 8.61.27 m23SHARED\_activate

**Description:** Activates a customer account identified by user and activation code, makes some checks if the data is correct and sends a welcome email.

**Parameter:**

- user: The user name of the customer.
- code: The activation code for the account.

#### 8.61.28 m23SHARED\_setCustomerLanguage

**Description:** Sets the language for this customer.

**Parameter:**

- lang: Language for the customer.

#### 8.61.29 m23SHARED\_getCustomerLanguage

**Description:** Returns the language setting for this customer.

**Returns:**

- : Language for this customer.

#### 8.61.30 m23SHARED\_sendActivationMail

**Description:** Prepares and sends the activation mail.

**Parameter:**

- eMail: eMail address of the new customer.
- username: Username of the new customer.
- lang: Language that should be used in the eMail.

### 8.61.31 m23SHARED\_sendActivationMail

**Description:** Prepares and sends the welcome mail.

**Parameter:**

- eMail: eMail address of the new customer.
- username: Username of the new customer.
- lang: Language that should be used in the eMail.

### 8.61.32 m23SHARED\_setRealName

**Description:** Sets the real name of the customer.

**Parameter:**

- name: Real name of the customer.

### 8.61.33 m23SHARED\_getRealName

**Description:** Returns the real name of the customer.

**Returns:**

- : Real name of the customer.

### 8.61.34 m23SHARED\_changeClientAmount

**Description:** Changes the amount of paid clients for the current customer.

**Parameter:**

- newClientAmount: The amount of clients, the customer wants to pay for.

### 8.61.35 m23SHARED\_showBill

**Description:** Shows the bill for a selected month.

**Parameter:**

- month: The month the bill should be generated.
- year: The year of the bill.

### 8.61.36 m23SHARED\_billDateSelection

**Description:** Creates an HTML selection with valid billing months.

**Parameter:**

- htmlName: Name of the HTML selection.

**Returns:**

- The selected month and year.

### 8.61.37 m23SHARED\_priceFormater

**Description:** Formats a price with two digit decimal place.

**Parameter:**

- price: The price to format.

**Returns:**

- The formatted price.

### 8.61.38 m23SHARED\_calculateBill

**Description:** Calculates the bill for a choosen month in a selected year.

**Parameter:**

- month: Month to get the bill for.
- year: Year to get the bill for.

**Returns:**

- Associative array with the bill data.

### 8.61.39 m23SHARED\_getLicenseType

**Description:** Returns the m23shared license of the current m23shared user.

**Returns:**

- License type as constant.

### 8.61.40 m23SHARED\_sendAdminMail

**Description:** Sends an GPG encrypted eMail to the admin.

**Parameter:**

- subject: The subject of the eMail.
- text: The eMail message.

### 8.61.41 m23SHARED\_getCurrentUser

**Description:** Returns the current m23shared user.

**Returns:**

- Current m23shared user.

### 8.61.42 m23SHARED\_changePasswordDialog

**Description:** Tries to change the password for the current m23shared user and shows an error or sucess message. Both of the entered passwords must be identically.

**Parameter:**

- pwd1: The password.
- pwd2: The retyped password.

### 8.61.43 m23SHARED\_changePasswordDialog

**Description:** Tries to change the eMail for the current m23shared user and shows an error or sucess message.

**Parameter:**

- email: New eMail address.

### 8.61.44 m23SHARED\_getBillDates

**Description:** Returns an array filled with all month and years where bills are present.

**Returns:**

- Associative array with all month and years where bills are present as array name and value.



#### 8.61.45 m23SHARED\_downloadBillPDFLink

**Description:** Generates a HTML link that points to the script that generates the bill for a given monath and year.

**Parameter:**

- month: Month to get the bill for.
- year: Year to get the bill for.

**Returns:**

- : HTML link to the bill PDF.

#### 8.61.46 m23SHARED\_getCustomerAddress

**Description:** Returns HTML formatted address information for the current customer.

**Returns:**

- Address information for the current customer.

#### 8.61.47 m23SHARED\_getCustomerBankHTML

**Description:** Returns HTML formatted bank account information for the current customer.

**Returns:**

- Bank account information for the current customer.

#### 8.61.48 m23SHARED\_pdfBill

**Description:** Generates a bill in PDF format for a choosen month in a selected year.

**Parameter:**

- month: Month to get the bill for.
- year: Year to get the bill for.

#### 8.61.49 m23SHARED\_setBankAccount

**Description:** Sets bank account information for the current customer.

**Parameter:**

- bankAccountHolder: Name of the account holder.
- bank: Name of the bank.
- accountNumber: The number of the bank account.
- bankCode: The bank code number.

#### 8.61.50 m23SHARED\_setAddress

**Description:** Sets address information for the current customer.

**Parameter:**

- resident: The name of the resident or his company.
- postCode: Postcode.
- city: City.
- street: Street.
- houseNumber: House number.

### 8.61.51 m23SHARED\_unusedPaidClientsAvailable

**Description:** Checks if there are unused paid clients and shows an error message if not.

**Returns:**

- : true if there are unused clients, false otherwise.

### 8.61.52 m23SHARED\_showLicenseDialog

**Description:** Shows a dialog for viewing and changing the license and paid client amount.

### 8.61.53 m23SHARED\_getAllm23sharedUsers

**Description:** Gets all m23shared users.

**Parameter:**

- onlyWithBillTable: If set to true, only m23shared users with existing bill table will be returned.

**Returns:**

- : Array with all m23 shared users.

### 8.61.54 m23SHARED\_switchUser

**Description:** Changes the current m23shared user.

### 8.61.55 m23SHARED\_prepareBillMailSending

**Description:** Prepares the bill mail sending queue.

**Parameter:**

- month: Month to mail the bill for.
- year: Year to mail the bill for.

**Returns:**

- Array with all users that have bill mails to send.

### 8.61.56 m23SHARED\_markBillMailAsSent

**Description:** Marks a bill mail as sent.

**Parameter:**

- month: Month to mail the bill for.
- year: Year to mail the bill for.
- user: DB name of the customer.

### 8.61.57 m23SHARED\_sendAllBillMails

**Description:** Sends the bills of all users from the previous month as PDF attachment. This should be run at the beginning of a month.

**8.61.58 m23SHARED\_addExtraBill**

**Description:** Adds an extra entry to the bill.

**Parameter:**

- amount: Amount of good.
- description: Description of the good.
- unitprice: Price per unit.

**8.61.59 m23SHARED\_showDonationDialog**

**Description:** Shows a dialog where the user can donate to the m23 project.

**8.61.60 m23SHARED\_showBootMediaDownloadDialog**

**Description:** Shows a dialog with download icons for the different boot media.

**8.61.61 m23SHARED\_allUserDBQuery**

**Description:** Executes an SQL query on all m23shared databases and returns an associated array with all results.

**Parameter:**

- sqlIN: The input query string that MUST include m23SHAREDDB as placeholder for the current m23shared DB name.

**Returns:**

- Associated array with the query result of all m23shared DBs.

**8.61.62 m23SHARED\_getUserByResident**

**Description:** Gets the user name of m23shared customer by the resident name.

**Parameter:**

- resident: Resident name of the customer.

**Returns:**

- m23shared user name for the searched customer or empty string if none was found.

**8.61.63 m23SHARED\_getUserByCustomerNr**

**Description:** Gets the user name of m23shared customer by the customer number.

**Parameter:**

- customerNr: Customer number of the m23shared customer.

**Returns:**

- m23shared user name for the searched customer or empty string if none was found.

#### 8.61.64 m23SHARED\_getUserByInfo

**Description:** Gets the user name of m23shared customer by searching all m23shared DBs for var and value in the remotevar table.

**Parameter:**

- remotevarVar: Variable name to search in the remotevar table.
- remotevarValue: Value to search in the remotevar table.

**Returns:**

- m23shared user name for the searched customer or empty string if none was found.

#### 8.61.65 m23SHARED\_showAdminDialog

**Description:** Shows a dialog for the admin to search users for and change values.

#### 8.61.66 m23SHARED\_blockAccount

**Description:** Blocks or unblocks an account.

**Parameter:**

- block: Set to true blocks, false unblocks.

#### 8.61.67 m23SHARED\_isAccountBlocked

**Description:** Checks if an account is blocked.

**Returns:**

- : True if the account is blocked, false if not.

#### 8.61.68 m23SHARED\_showAddExtraBillDialog

**Description:** Shows a dialog for adding extra entries to the bill.

#### 8.61.69 m23SHARED\_showBillDialog

**Description:** Shows a dialog for viewing and choosing the bill.

#### 8.61.70 m23SHARED\_showPriceListTable

**Description:** Shows a table with the price list.

#### 8.61.71 m23SHARED\_getInformationForBootingYourClientLink

**Description:** Generates a link to the help page information for booting the client.

**Parameter:**

- : client: Name of the client (if the value from the session should not/cannot be taken)

**Returns:**

- Link to the help page information for booting the client.

## 8.62 *./inc/mail.php*

Author: Hauke Goos-Habermann (HHabermann@pc-kiel.de)

Description: Functions for sending mails.

### 8.62.1 *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.

### 8.62.2 *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.

### 8.62.3 *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.

### 8.62.4 *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.

### 8.62.5 *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 attach.
- 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.

### 8.62.6 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.

### 8.62.7 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.

### 8.62.8 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.

### 8.62.9 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.

### 8.62.10 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.

### 8.62.11 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.

### 8.62.12 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.

### 8.62.13 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.

### 8.62.14 MAIL\_importGPGKey

**Description:** Imports an GPG key into the bunch of GPG keys.

**Returns:**

- : True on sucess otherwise false.

### 8.62.15 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.

### 8.62.16 MAIL\_manageGPGKeysDialog

**Description:** Shows a dialog for importing and deleting GPG keys.

### 8.62.17 MAIL\_sendMail

**Description:** Sends a mail, that may be GPG encrypted and contain an attachment 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.

### 8.62.18 MAIL\_cryptMailServer

**Description:** Server part for sending AES mails.



## 8.63 *./inc/massTools.php*

Author: Hauke Goos-Habermann (HHabermann@pc-kiel.de)

Description: routines for mass installations

### 8.63.1 *MASS\_EGKradioBoxes*

**Description:** Generates HTML code for showing 3 elements, that can be each a "radio button", selection "disabled" or "always selected".

**Parameter:**

- RB\_name: name of the radio button
- arr: array with 3 values for [Enter,Generate,Keep]. Setting a value to "e" means that the user can select, "n" selection is disabled, "y" is always select.
- checkNr: the number of radio button that is enabled by default.

### 8.63.2 *MASS\_FHradioBoxes*

**Description:** Generates HTML code for showing 2 radio buttons for selecting file or handy source

**Parameter:**

- RB\_name: name of the radio button
- checkNr: the number of radio button that is enabled by default.

### 8.63.3 *MASS\_showFileHandDialog*

**Description:** shows a dialog for selecting "by file" or "by hand" for the "enter" properties.

**Parameter:**

- EGKparams: enter generate keep parameters, that hold information about handling of the properties

### 8.63.4 *MASS\_propertyKeys*

**Description:** returns the keys for all properties

### 8.63.5 *MASS\_showFileFormatDialog*

**Description:** shows a dialog that lets the user select a DB file and assign the columns to the fields of the file

**Parameter:**

- EGKparams: enter generate keep parameters, that hold information about handling of the properties

### 8.63.6 *MASS\_keyToI18N*

**Description:** converts the property names to I18N names

**Parameter:**

- key: property name

### 8.63.7 *MASS\_I18NTokey*

**Description:** converts the I18N names to property names

**Parameter:**

- key: property name

### 8.63.8 MASS\_showTableDefinition

**Description:** shows a dialog that lets the user define which field in the DB file should be assigned to which property

**Parameter:**

- EGKparams: enter generate keep parameters, that hold information about handling of the properties
- DBfileName: file name of the DB file

### 8.63.9 MASS\_checkAndSaveFields

**Description:** saved the assignments from field number to property and other information to EGKparams and performs a simple check, to verify that the values of the properties are valuable. An error message is returned or an empty string, if all is ok.

**Parameter:**

- EGKparams: enter generate keep parameters, that hold information about handling of the properties

### 8.63.10 MASS\_openDBFile

**Description:** opens a DB file

**Parameter:**

- fileName: name of the DB file

### 8.63.11 MASS\_readDBFile

**Description:** reads a line from the DB file and returns an associated array with the properties as key and the fields of the file as values.

**Parameter:**

- fileName: name of the DB file

**Returns:**

- Associative array with the values of the DB line or false, if the line was empty.

### 8.63.12 MASS\_readDBFileRaw

**Description:** reads a line from the DB file and returns the fields splitted to a normal array.

**Parameter:**

- file: file pointer
- glue: the separator used to separate the fields

### 8.63.13 MASS\_closeDBFile

**Description:** closes the DB file.

**Parameter:**

- file: file pointer

### 8.63.14 MASS\_getXProperties

**Description:** returns the amount and keys of a social kind (enter, generate, keep, hand, file)

**Parameter:**

- EGKparams: enter generate keep parameters, that hold information about handling of the properties
- x: the 1-letter code of enter, generate, keep, hand or file
- pre: set if there is a prefix before the key name

### 8.63.15 MASS\_showGeneratorOptions

**Description:** shows the dialog for configuring the generator options

**Parameter:**

- EGKparams: enter generate keep parameters, that hold information about handling of the properties

### 8.63.16 MASS\_passGenerator

**Description:** generates the selected amount of passwords with a random algorithm or the pwgen tool.

**Parameter:**

- length: length of the passwords to generate
- method: random or pwgen generated passwords that can be memorized by humans easily
- amount: the amount of passwords to generate

**Returns:**

- Array with the generated passwords as keys.

### 8.63.17 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, "ForeFamily-Name" if the logins should be created from fore- and familynames
- amount: the amount of logins to generate

### 8.63.18 MASS\_ipGenerator

**Description:** generates the selected amount of IPs in the selected ranges. Only IPs are generated that aren't in use by m23 or (if activated) pingable.

**Parameter:**

- amount: the amount of IPs to generate
- rangeStr: string with IP range information
- ping: set to true, if each IP should be pinged before it becomes valid

### 8.63.19 MASS\_minMaxIP

**Description:** calculates the possible minimum and maximum IP of a given netmask. The IPs are returned as an array: index 0 = minimum; index 1 = maximum.

**Parameter:**

- netmask: netmask to use
- ip: is used if the can only be set the current part of the ip (max and min ip part == 255)

### 8.63.20 MASS\_generateNetmask

**Description:** generate netmasks from ip addresses via network class definitions.

**Parameter:**

- ip: the ip that should be used to calculate the netmask

### 8.63.21 MASS\_generateClientNames

**Description:** generates client names through appending of numbers.

**Parameter:**

- base: the client base name
- start: the start number
- amount: the amount of client names to generate

### 8.63.22 MASS\_saveGeneratorOptions

**Description:** saves all generator options to EGKparams

**Parameter:**

- EGKparams: enter generate keep parameters, that hold information about handling of the properties

### 8.63.23 MASS\_showOverview

**Description:** shows a table with all generated client settings, that can be edited

**Parameter:**

- EGKparams: enter generate keep parameters, that hold information about handling of the properties

### 8.63.24 MASS\_getAllFromFile

**Description:** returns all values from one key of the DB file as an array.

**Parameter:**

- key: the key of the property
- generateAmount: the amount of values to be read from the DB file
- EGKparams: enter generate keep parameters, that hold information about handling of the properties
- fromDBFile: 2D array filled with the values for the keys

### 8.63.25 MASS\_getLongestLength

**Description:** returns the length of the longest entry in the array or max if bigger than max

**Parameter:**

- arr: the array
- amount: the amount of entries in the array
- max: maximal value to be returned

### 8.63.26 MASS\_startInstall

**Description:** starts the installation of all client with all parameters defined in the table

**Parameter:**

- EGKparams: enter generate keep parameters, that hold information about handling of the properties

## 8.64 *./inc/menu.php*

Author: Hauke Goos-Habermann (HHabermann@pc-kiel.de)

Description: functions for menus

### 8.64.1 **MENU\_showEntry**

**Description:** generates a menu entry, highlights it (if selected) and removes menu entry formatting tags from the menu entry label

**Parameter:**

- text: label of the menu entry
- link: link to the page
- icon: directory and name of the icon to show in front of the entry

### 8.64.2 **MENU\_startGroup**

**Description:** shows the start of a menu group

**Parameter:**

- name: name of the menu group

### 8.64.3 **MENU\_endGroup**

**Description:** shows the end of a menu group

## 8.65 `./inc/message.php`

Author: Hauke Goos-Habermann (HHabermann@pc-kiel.de)

Description: function to show an info box

### 8.65.1 `MSG_showMessageBoxPlaceholder`

**Description:** Shows a placeholder for message boxes at the position of execution.

### 8.65.2 `MSG_placeOrReturnMessageBox`

**Description:** Replaces a (maybe) existing message box placeholder with the given text.

**Parameter:**

- text: Message to embed in the placeholder.

**Returns:**

- Empty string, if the text could be inserted into the existing placeholder or the message, if no placeholder was found.

### 8.65.3 `MSG_getm23UpdateFeed`

**Description:** Shows the m23 server update feed.

**Parameter:**

- width: Width of the box
- refreshTime: The time in minutes the file is downloaded again.

### 8.65.4 `MSG_getm23DevelopmentBlog`

**Description:** Shows the m23 development blog.

**Parameter:**

- width: Width of the box
- refreshTime: The time in minutes the file is downloaded again.

### 8.65.5 `MSG_getRSSFeed`

**Description:** Shows a RSS feed.

**Parameter:**

- url: The URL pointing to the RSS XML file.
- tableType: Name of the CSS table type
- width: Width of the box
- storeFile: The file name to store the download in.
- refreshTime: The time in minutes the file is downloaded again.

### 8.65.6 MSG\_showInfo

**Description:** Shows the help block for the online help.

**Parameter:**

- message: the text for the info message
- language: two character language description (e.g. de, en, fr,...)
- width: width of the box

### 8.65.7 MSG\_showError

**Description:** Shows the error block for the error messages.

**Parameter:**

- message: the text for the info message
- language: two character language description (e.g. de, en, fr,...)
- width: width of the box

### 8.65.8 MSG\_showWarning

**Description:** Shows the warning block for the warning messages.

**Parameter:**

- message: the text for the info message
- language: two character language description (e.g. de, en, fr,...)
- width: width of the box

### 8.65.9 MSG\_show

**Description:** Shows the message block for the messages.

**Parameter:**

- message: the text for the info message
- language: two character language description (e.g. de, en, fr,...)
- width: width of the box
- urgency: type of message (e.g. errortable, warningtable, infotable)

### 8.65.10 MSG\_showMessageBoxHeader

**Description:** shows the header of the message block for the online help

**Parameter:**

- tableType: name of the CSS table type
- type: the heading of the box
- width: width of the box
- dontShowButReturn: Set to true if the HTML output should be returned rather than show.

### 8.65.11 MSG\_showMessageBoxFooter

**Description:** shows the footer of the message block for the online help

**Parameter:**

- dontShowButReturn: Set to true if the HTML output should be returned rather than show.

### 8.65.12 MSG\_showMessageBox

**Description:** shows the message block for the online help

**Parameter:**

- message: the text for the info message
- tableType: name of the CSS table type
- type: the heading of the box
- width: width of the box
- dontShowButReturn: Set to true if the HTML output should be returned rather than show.

### 8.65.13 MSG\_showUpdateInfo

**Description:** shows a info message about the stored update jobs

**Parameter:**

- unr: the amount of update jobs and clients
- language: two character language description (e.g. de, en, fr,...)

### 8.65.14 MSG\_showUpdateInfo

**Description:** shows a info message about stored jobs on N clients

**Parameter:**

- jobNr: the amount of jobs
- clNr: the amount of clients
- language: two character language description (e.g. de, en, fr,...)

### 8.65.15 MSG\_showNewFeature

**Description:** shows the new feature block

**Parameter:**

- url: the url to the forum for the new feature
- language: two character language description (e.g. de, en, fr,...)
- width: width of the box

### 8.65.16 MSG\_DeActivateBlogDialog

**Description:** Creates a dialog to en/disable a blog. The displaying state is written to the DB.

**Parameter:**

- blogVarName: Variable name of the blog to store in the DB.
- css: Name of the CSS class to color the dialog.
- blogName: Name of the blog als human readle heading.
- width: Width of the box containing the switch dialog.
- dialogCode: The HTML code of the dialog element is written to this variable.

**Returns:**

- : True if the blog should be shown otherwise false.



## 8.66 *./inc/messageReceive.php*

Author: Hauke Goos-Habermann (HHabermann@pc-kiel.de)

Description: Functions for handling messages sent by m23 clients.

### 8.66.1 *MSR\_decodeMessage*

**Description:** checks the type of the message and calls the right procedure

### 8.66.2 *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.

### 8.66.3 *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.

### 8.66.4 *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.

### 8.66.5 *MSR\_statusBarInc*

**Description:** Increments the status bar percent by a given amount for the current client and for the "install-Status" status bar.

**Parameter:**

- percent: Percent value of the current job.

### 8.66.6 *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.

### 8.66.7 *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).

### 8.66.8 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.

### 8.66.9 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).

### 8.66.10 MSR\_curDynIP

**Description:** Sets the current IP of a client with dynamic IP.

**Parameter:**

- curIP: The current IP.

### 8.66.11 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.

### 8.66.12 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.

### 8.66.13 MSR\_CommandCopyClientPackageStatus

**Description:** Generates commands to transfer the package status file from the client to the server.

### 8.66.14 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)

### 8.66.15 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.

### 8.66.16 MSR\_decodeClientSideBase64

**Description:** Decodes the slightly modiflicated 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.

### 8.66.17 MSR\_clientSideBase64Encode

**Description:** Encodes a given file to (a slightly different, + is converted to - for sending it as post variable via wget) base64 format and appends the output to statusdata.post. There are two methods for generating the base64 output. First the native uuencode tool that is very fast and second 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.

### 8.66.18 MSR\_genSendBinayFileCommand

**Description:** Sends a binary file to postMessage.php. In contrast to MSR\_genSendCommand MSR\_genSendBinayFileComm is binary safe.

**Parameter:**

- fileName: Name of the file
- type: Type of the message understood by MSR\_decodeMessage

### 8.66.19 MSR\_m23ImagerMBR

**Description:** Saves the received master boot record to a file after cecking the MD5.

### 8.66.20 MSR\_m23ImagerSize

**Description:** Sets the size of the decompressed image in its file name.

### 8.66.21 MSR\_getm23clientIDCMD

**Description:** returns the BASH code for storing the m23clientID in the BASH variable "varid"

**Parameter:**

- addChar: is the

### 8.66.22 MSR\_importPackageStatus

**Description:** imports the data from a package status message to the database. pNr=package, vNr=version, sNr=status

### 8.66.23 MSR\_importLog

**Description:** reads log data from the post data and store it in the DB

### 8.66.24 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.

### 8.66.25 MSR\_statusFileCommand

**Description:** generates the commands to send the package infos and package status file

### 8.66.26 MSR\_importStatusFile

**Description:** reads status file data from the post data and store it under /m23/var/cache/clients/clientName/packageStatus

### 8.66.27 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.

### 8.66.28 MSR\_clientChangeCommand

**Description:** sends the ID of a "m23changeClient" job to the server.

**Parameter:**

- id: the ID of the client job

### 8.66.29 MSR\_clientChange

**Description:** executes changes of a "m23changeClient" job on the server.

### 8.66.30 MSR\_partHwDataCommand

**Description:** generates the commands to send partition and hardware info

**8.66.31 MSR\_importPartHwData**

**Description:** imports partition and hardware information

**8.66.32 MSR\_getClientSettingsCommand**

**Description:** Generates a script to gather network, release, distribution, login, LDAP, NFS and kernel informations for client import on the client

**8.66.33 MSR\_clientSettings**

**Description:** Imports the clients settings

**8.66.34 MSR\_setOnline**

**Description:** Sets the online-state of a client.

**Parameter:**

- online: The new online-state.

**8.66.35 MSR\_addMonitor**

**Parameter:**

- vendor: The vendor of the monitor
- model: The model of the monitor
- edid: The monitor information

**8.66.36 MSR\_listMonitors**

**Returns:**

- list of monitors

## 8.67 `./inc/packageBuilder.php`

Author: Hauke Goos-Habermann (HHabermann@pc-kiel.de)

Description: Functions for building pools.

### 8.67.1 `PKGBUILDER_showDialog`

**Description:** Shows a dialog for creating Debian packages from archives.

### 8.67.2 `PKGBUILDER_showKeySelectionDialog`

**Description:** Shows a dialog for choosing the GPG signing key for the extra packages.

### 8.67.3 `PKGBUILDER_showUploadDialog`

**Description:** Shows a dialog for uploading the tar files with checking of the file extension.

### 8.67.4 `PKGBUILDER_listFiles`

**Description:** Shows a dialog of all files in `EXTRA_DEBS_DIRECTORY` with possibility to create Debian packages from tar files and to delete files.

### 8.67.5 `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. `"^BUT_action/"`). **Returns:**
  - : The first found key that matches the expression or false if no matching key was found.

### 8.67.6 `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.

## 8.68 *./inc/packages.php*

Author: Hauke Goos-Habermann (HHabermann@pc-kiel.de)

Description: basic package operations (search, add,...)

### 8.68.1 **PKG\_addHSUser**

**Description:** Adds a job for creating an user on a halfSister client.

**Parameter:**

- client: Name of the client.
- login: Login name of the new user.
- firstpw: Password for the new user.
- uid: Optional user ID of the new user.
- gid: Optional group ID of the new user.

### 8.68.2 **PKG\_addUbuntuUser**

**Description:** Adds a job for creating an user on a Ubuntu client.

**Parameter:**

- client: Name of the client.
- login: Login name of the new user.
- firstpw: Password for the new user.
- uid: Optional user ID of the new user.
- gid: Optional group ID of the new user.

### 8.68.3 **PKG\_addDebianUser**

**Description:** Adds a job for creating an user on a Debian client.

**Parameter:**

- client: Name of the client.
- login: Login name of the new user.
- firstpw: Password for the new user.
- uid: Optional user ID of the new user.
- gid: Optional group ID of the new user.

### 8.68.4 **PKG\_addUser**

**Description:** Adds a job for creating an user on the client.

**Parameter:**

- client: Name of the client.
- login: Login name of the new user.
- firstpw: Password for the new user.
- groups: Array of groups the user should be added.
- uid: Optional user ID of the new user.
- gid: Optional group ID of the new user.

### 8.68.5 PKG\_cleanPackageLine

**Description:** Removes unwanted characters from a line containing package names and makes sure that there is only one line without line breaks.

**Parameter:**

- packageLine: Space separated line containing the package names. The changed line will be written to the parameter too.

### 8.68.6 PKG\_combinem23normal

**Description:** Combines the package names of multiple entries for m23normal and m23normalRemove jobs in a package selection.

**Parameter:**

- packageSelectionName: Name of the package selection to optimise.

### 8.68.7 PKG\_importSelectedPackagesFromFile

**Description:** Imports space-separated 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-separated packages.

### 8.68.8 PKG\_exportSelectedPackages

**Description:** Exports the wait4acc/selected packages of a client.

**Parameter:**

- client: Name of the client or empty.

### 8.68.9 PKG\_getPackageStatusCSV

**Description:** Exports the packages of a client in tabulator separated CSV.

**Parameter:**

- client: Name of the client.

### 8.68.10 PKG\_getDebootstrapCacheFilename

**Description:** Returns the file name of the debootstrap cache file (without path).

**Parameter:**

- release: Select the Debian/Ubuntu suite (squeeze, sarge, sid, precise).
- arch: the computer architecture of the client

**Returns:**

- The file name of the debootstrap cache file (without path).



### 8.68.11 PKG\_getDebootstrapCacheSfURL

**Description:** Returns the URL to the debootstrap cache file on the SourceForge server.

**Parameter:**

- release: Select the Debian/Ubuntu suite (squeeze, sarge, sid, precise).
- arch: the computer architecture of the client

**Returns:**

- The URL to the debootstrap cache file on the SourceForge server.

### 8.68.12 PKG\_baseSysDownloadedCompletelyTom23Server

**Description:** Checks, if the debootstrap cache file was downloaded completely to the m23 server.

**Parameter:**

- release: Select the Debian/Ubuntu suite (squeeze, sarge, sid, precise).
- arch: the computer architecture of the client

**Returns:**

- true, on complete download otherwise false.

### 8.68.13 PKG\_downloadBaseSysTom23Server

**Description:** Downloads the debootstrap cache file to the m23 server and checks its validity (by signature).

**Parameter:**

- release: Select the Debian/Ubuntu suite (squeeze, sarge, sid, precise).
- arch: the computer architecture of the client

**Returns:**

- true, if the download is completed, otherwise false.

### 8.68.14 PKG\_getDebootstrapCacheServerURL

**Description:** Returns the URL to the debootstrap cache file on the m23 server.

**Parameter:**

- release: Select the Debian/Ubuntu suite (squeeze, sarge, sid, precise).
- arch: the computer architecture of the client

**Returns:**

- The URL to the debootstrap cache file on the m23 server.

### 8.68.15 PKG\_getDebootstrapCacheServerFile

**Description:** Returns the full path to the debootstrap cache file on the m23 server.

**Parameter:**

- release: Select the Debian/Ubuntu suite (squeeze, sarge, sid, precise).
- arch: the computer architecture of the client

**Returns:**

- The full path to the debootstrap cache file on the m23 server.

### 8.68.16 **PKG\_isReconfiguredWithExtraDistr**

**Description:** Checks, if the distribution is used for configuring a system that was installed via image.

**Parameter:**

- pkgID: The ID of the base installation package.

**Returns:**

- True, if the distribution is used for configuring, otherwise false.

### 8.68.17 **PKG\_translateClientjobsStatus**

**Description:** Translates the clientjobs status from the DB into a human readable form.

**Parameter:**

- status: Status code from the DB.

**Returns:**

- Human readable translation of the clientjobs status.

### 8.68.18 **PKG\_isSpecialPackageAvailableForClient**

**Description:** Checks if a special package is available for the client's distribution.

**Parameter:**

- client: Name of the client.
- package: Name of the special package.

**Returns:**

- True if the special package is available otherwise false.

### 8.68.19 **PKG\_OptionPageHeader2**

**Description:** Starts the option page for debconf settings with all necessary options.

**Parameter:**

- title: the window title of the OptionPage

### 8.68.20 **PKG\_OptionPageTail2**

**Description:** Generates the bottom of the OptionPage for debconf settings.

**Parameter:**

- elem: The elements (variable name, type (string, boolean, (multi)select), default value, description) of the debconf templates.

### 8.68.21 **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-formatted description.

### 8.68.22 PKG\_OptionPageRender2

**Description:** Renderes the layout of an OptionPage for debconf and stored the debconf settings into the DB.

**Parameter:**

- layout: The elements (variable name, type (string, boolean, (multi)select), default value, description) of the debconf templates.
- client: The name of the client, the debconf settings should be stored for.
- package: Name of the package, the debconf settings should be stored for.

### 8.68.23 PKG\_countSpecialPackages

**Description:** counts the special packages of a clients matching the package name and status

**Parameter:**

- clientName: name of the client
- packageName: name of the special package
- status: status of the package

### 8.68.24 PKG\_countWaitingJobs

**Description:** returns the amount of a special waiting package

**Parameter:**

- client: name of the client
- package: name of the package you want to know the amount

### 8.68.25 PKG\_closeSearch

**Description:** closes a search request

**Parameter:**

- file: file handle

### 8.68.26 PKG\_getNextPackage

**Description:** fetches the next package

**Parameter:**

- file: file handle

### 8.68.27 PKG\_listRecommendPackages

**Description:** lists recommended packages matching the key

**Parameter:**

- key: search key
- install: set to true, if the selection for installing all packages should be first

### 8.68.28 PKG\_listRecommendSubPackages

**Description:** returnes subpackages of a package

**Parameter:**

- cut: cuts the packages by \$cut
- params: variable to write package names to

### 8.68.29 PKG\_addRecommendPackages

**Description:** adds recommended packages to db

**Parameter:**

- amount: amount of selected packages
- client: name of client to install packages on
- normalPackageType2: m23normal, m23normalRemove or orig: select if the packages should be (de)installed or use the saved action
- distr: Name of the distribution.

### 8.68.30 PKG\_addPackageSelection

**Description:** Adds a package selections to the list of packages to install.

**Parameter:**

- client: name of client to install packages on
- packageSelectionName: Name of the package selection to install.
- normalPackageType2: m23normal, m23normalRemove or orig: select if the packages should be (de)installed or use the saved action
- distr: Name of the distribution.

### 8.68.31 PKG\_addNormalPackagesToWait4Aac

**Description:** adds a package to waiting 4 accept status

**Parameter:**

- client: name of client to install packages on
- priority: priority of the package
- params: parameter for installing the package

### 8.68.32 PKG\_addSpecialPackagesToWait4Aac

**Description:** adds a special package to waiting 4 accepts status

**Parameter:**

- client: name of client to install packages on
- priority: priority of the package
- params: parameter for installing the package
- distr: Name of the distribution.

### 8.68.33 PKG\_countJobsWithStatus

**Description:** Counts named jobs on a client that have a special status.

**Parameter:**

- client: name of the client
- package: name of the package
- status: The status to search for

**Returns:**

- The amount of packages on the client with the given status.

### 8.68.34 PKG\_getClientjobsStatus

**Description:** return the status of a job

**Parameter:**

- client: name of the client
- package: name of the package
- distr: the name of the distribution
- params: parameter for installing the package
- normalPackage: the name of a normal package

### 8.68.35 PKG\_addNormalPackages

**Description:** adds normal packages to db

**Parameter:**

- amount: amount of selected packages
- client: name of client to install packages on

### 8.68.36 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.

### 8.68.37 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.

### 8.68.38 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.

### 8.68.39 **PKG\_listSelectedpackages**

**Description:** lists the packages with wait4acc status

**Parameter:**

- client: name of client to install packages on
- distr: the name of the distribution
- release: release of the distribution

### 8.68.40 **PKG\_countSelectedpackages**

**Description:** counts the preselected packages

**Parameter:**

- client: name of client to install packages on

### 8.68.41 **PKG\_countJobs**

**Description:** counts all packages of a client with a given status

**Parameter:**

- client: name of client
- status: status of the packages to be count or empty to count all jobs

### 8.68.42 **PKG\_hasOptions**

**Description:** generates a link to the package option page (if it exists)

**Parameter:**

- package: name of package
- packageID: id for the selected package
- distr: the name of the distribution
- client: Name of the current client.
- release: The release of the client's distribution.

### 8.68.43 **PKG\_savePackageselection**

**Description:** saves all selected packages a package selection

**Parameter:**

- client: name of client to install packages on
- selectionName: name for the package selection
- showMsg: set to true, if a message should be shown
- status: Status of the clientjobs that should be added.

#### 8.68.44 PKG\_addPackageToPackageselection

**Description:** Add packages to selection

**Parameter:**

- client: name of client to install packages on
- selectionName: name for the package selection
- packageName: name of the normal package
- params: parameter for the package
- normalPackage: the name of a normal package
- installedSize: the size of the package if it is installed
- priority: The priority of the package.

#### 8.68.45 PKG\_listSpecialpackages

**Description:** lists special packages matching a key

**Parameter:**

- key: search key

#### 8.68.46 PKG\_addSpecialPackages

**Description:** adds normal packages to db

**Parameter:**

- amount: amount of selected packages
- client: name of client to install packages on

#### 8.68.47 PKG\_getSpecialPackagePriority

**Description:** gets the priority of a special package

**Parameter:**

- package: name of package
- distr: name of the distribution

#### 8.68.48 PKG\_getSpecialPackageDescription

**Description:** gets the description of a special package

**Parameter:**

- package: name of package

#### 8.68.49 PKG\_getSpecialPackageInfo

**Description:** gets informations from special packages

**Parameter:**

- package: name of package
- infoType: the type of information you want to get
- dist: shortname of the distribution

**Returns:**

- The information or false, if no information could be got.

### 8.68.50 PKG\_getPackageID

**Description:** get the id for a wait4acc job

**Parameter:**

- client: name of the client
- package: name of the package

### 8.68.51 PKG\_rmNormalJob

**Description:** adds a normal package removal job to the clientjobs table

**Parameter:**

- client: name of the client
- packageName: name of the package
- priority: The priority of the job.

### 8.68.52 PKG\_addJob

**Description:** adds a job to the clientjobs table

**Parameter:**

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

### 8.68.53 PKG\_discardNormalJob

**Description:** discards all normal packages from the clientjobs table, that match the param

**Parameter:**

- client: name of the client
- packageName: name of the package

### 8.68.54 PKG\_addWait4AccJob

**Description:** adds a wait 4 accept job to the clientjobs table

**Parameter:**

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



### 8.68.55 PKG\_addStatusJob

**Description:** adds a job to the clientjobs table

**Parameter:**

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

### 8.68.56 PKG\_acceptJobs

**Description:** accepts all preselected jobs

**Parameter:**

- client: name of the client
- showMsg: set to true, if a message about assigned jobs should be shown

### 8.68.57 PKG\_discardJobs

**Description:** discards all preselected jobs

**Parameter:**

- client: name of the client

### 8.68.58 PKG\_discardJob

**Description:** discards a selected job

**Parameter:**

- client: name of the client
- package: name of package you want to discard

### 8.68.59 PKG\_changeClientPackageAction

**Description:** changes the status of a already installed package

**Parameter:**

- client: name of the client
- package: name of package you want to discard
- action: the action you want the package set to

### 8.68.60 PKG\_setClientPackageWait4Rm

**Description:** changes the status of a already installed package to wait 4 delete

**Parameter:**

- client: name of the client
- package: name of package

### 8.68.61 PKG\_setClientPackageInstalledOK

**Description:** changes the status of a package to "installed ok"

**Parameter:**

- client: name of the client
- package: name of package

### 8.68.62 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

### 8.68.63 PKG\_addShutdownOrRebootPackage

**Description:** Adds a shutdown or a reboot package. No new job is added if there is already a waiting shutdown or reboot job. A shutdown package is added if the client can't be pinged and a reboot package if it is reachable via the network.

**Parameter:**

- client: name of the client

### 8.68.64 PKG\_getAllParams

**Description:** gets all parameters of the parameters column of a clientjob

**Parameter:**

- packageID: the ID of the package

### 8.68.65 PKG\_setAllParams

**Description:** sets all parameters in the parameters column of a clientjob

**Parameter:**

- packageID: the ID of the package
- params: the parameters as associative array

### 8.68.66 PKG\_OptionPageHeader

**Description:** starts the option page with all necessary options

**Parameter:**

- title: the window title of the OptionPage

### 8.68.67 PKG\_OptionPageRender

**Description:** renders the layout of a OptionPage

**Parameter:**

- layout: for the definition of the layout array see the development guide

### 8.68.68 PKG\_OptionPageTail

**Description:** generates the bottom of the OptionPage

**Parameter:**

- layout: for the definition of the layout array see the development guide

### 8.68.69 PKG\_OptionPageSaveAlsParameters

**Description:** saves the entered values in the packagejobs params

**Parameter:**

- layout: for the definition of the layout array see the development guide

### 8.68.70 PKG\_OptionPageGetValue

**Description:** gets the 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 associative array

### 8.68.71 PKG\_listParams

**Description:** lists the parameters from a package in a nice format

**Parameter:**

- paramStr: the parameters as string (simply read from the packagejobs table)

### 8.68.72 PKG\_getRecommendPackageAllInstalledSize

**Description:** calculates the whole sum of the installedSizes of all recommend packages of one package selection

**Parameter:**

- packageSelection: the name of the package selection

### 8.68.73 PKG\_previewInstallationDeinstallation

**Description:** shows what happens if a client deinstalls/ installs waiting packages and generates a table with title

**Parameter:**

- clientName: name of the client
- install: set to true, if packages should be installed. if false the packages should be deinstalled

### 8.68.74 PKG\_showPreviewInstallationDeinstallation

**Description:** shows what happens if a client installs / deinstalls waiting packages and generates a table with title

**Parameter:**

- clientName: name of the client
- install: set to true, if packages should be installed. if false the packages should be deinstalled

### 8.68.75 PKG\_updateSourcesListAtAllClients

**Description:** updates the sources.list at all clients using it

**Parameter:**

- sourcename: name of the sources.list that should be updated

### 8.68.76 PKG\_executeOnClientJobs

**Description:** Executes a sql statement on all package IDs.

**Parameter:**

- sql: initial SQL statement e.g. "DELETE FROM 'clientjobs' WHERE "
- packageIDList: the list of IDs of jobs to be deleted

### 8.68.77 PKG\_removeFromJobList

**Description:** removes all jobs identified by the IDs in packageIDList

**Parameter:**

- packageIDList: the list of IDs of jobs to be deleted

### 8.68.78 PKG\_changeClientJobsStatus

**Description:** Sets a new status on all jobs identified by the IDs in packageIDList

**Parameter:**

- packageIDList: the list of IDs of jobs to be deleted
- status: New status to set

### 8.68.79 PKG\_removeSpecialFromJobList

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

**Parameter:**

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

### 8.68.80 PKG\_getHigestIDOfSpecialPackage

**Description:** Gets the highest ID of a special package.

**Parameter:**

- clientName: Name of the client.
- package: Name of the special package.

### 8.68.81 PKG\_previewUpdateSystem

**Description:** returns the information of an system update request

**Parameter:**

- clientName: name of the client
- completeUpdate: set it to "true", if it should be a full update (installation and removal of packages) or to "false" for an update of existing packages

### 8.68.82 PKG\_showPreviewUpdateSystem

**Description:** generates HTML code with information about the update preview

**Parameter:**

- clientName: name of the client
- completeUpdate: set it to "true", if it should be a full update (installation and removal of packages) or to "false" for an update of existing packages

### 8.68.83 PKG\_rmAllSpecialPackagesByName

**Description:** deletes all special packages from a client matching the package name

**Parameter:**

- clientName: name of the client
- packageName: name of the special package

### 8.68.84 PKG\_getClientsWithPackage

**Description:** Gets all clients that have the specific package installed (or with another status).

**Parameter:**

- packageName: Name of the package.
- status: The status the package should have.

**Returns:**

- Array with all clients that have the specific package installed (or with another status).

### 8.68.85 PKG\_getClientsWithWaitingJobs

**Description:** Gets all clients that have waiting jobs.

**Returns:**

- Array with all clients that have waiting jobs.

### 8.68.86 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 packages with the given status will be added to the output array.

**Returns:**

- Array with all clients that have the specific package (not) installed (or with another given status).

### 8.68.87 PKG\_countPackages

**Description:** counts all packages on a client

**Parameter:**

- clientName: name of the client

### 8.68.88 PKG\_copyWait4accPackagesToClient

**Description:** copies the waiting jobs from one client to another

**Parameter:**

- from: the source client
- to: the destination client

### 8.68.89 PKG\_copyPackagesToClient

**Description:** copies all with a selected status jobs from one client to another

**Parameter:**

- from: the source client
- to: the destination client
- status: can be set to a package status or be empty to copy all jobs

### 8.68.90 PKG\_remNormalPackages

**Description:** adds normal deinstallation jobs to db

**Parameter:**

- amount: amount of selected packages
- client: name of client to deinstall packages on

### 8.68.91 PKG\_addRemovePackagesToWait4Aac

**Description:** adds a remove job to waiting 4 accept status

**Parameter:**

- client: name of client to frinstall packages from
- priority: priority of the package
- params: parameter for deinstalling the package

### 8.68.92 PKG\_discardRemoveJob

**Description:** discards all remove jobs from the clientjobs table, that match the param

**Parameter:**

- client: name of the client
- packageName: name of the package

### 8.68.93 PKG\_deletePackageselection

**Description:** delete all packages from package selection

**Parameter:**

- selectionName: name for the package selection

### 8.68.94 PKG\_getAllPackageSelections

**Description:** returns all package selection names

**Parameter:**

- addEmpty: set to true to add an empty entry at the beginning.

### 8.68.95 PKG\_multiPackageSelectionsSelection

**Description:** Generates a multi selection with all package selections.

**Parameter:**

- selName: name of the selection
- first: entry that should be shown first (this is the internal value and NOT the name shown to the user). the first value from the list will be written to \$first. set first to "false" to disable writing the first entry.
- addEmpty: set to true to add an empty entry at the beginning.

### 8.68.96 PKG\_showAllPackageSelections

**Description:** returns all package selection as HTML selection

**Parameter:**

- selName: name of the selection
- first: entry that should be shown first (this is the internal value and NOT the name shown to the user). the first value from the list will be written to \$first. set first to "false" to disable writing the first entry.
- addEmpty: set to true to add an empty entry at the beginning.

### 8.68.97 PKG\_getPackageParams

**Description:** gets the parameters for a selected package

**Parameter:**

- id: package ID

### 8.68.98 PKG\_getClientbyPackageID

**Description:** gets the clientname that owns a selected package ID

**Parameter:**

- id: package ID

### 8.68.99 PKG\_getInfoFromPackageID

**Description:** gets a row from "clientjobs" for a given package ID

**Parameter:**

- id: package ID
- variable: the name of the row (e.g. client)

### 8.68.100 PKG\_getClientIDbyPackageID

**Description:** returns the ID of a client that owns a selected package ID

**Parameter:**

- id: package ID

### 8.68.101 PKG\_getPackageParamsVar

**Description:** fetch the device for installation

**Parameter:**

- id: package ID
- var: name of variable you want to get the value of

### 8.68.102 PKG\_getPackageIDsByName

**Description:** returns all IDs as an array for jobs matching the client and job name and are a normal or special package.

**Parameter:**

- client: the name of the client, the jobs are for
- packageName: name of the package, can be the name of a normal or special package
- specialPackage: set to true, if you want to search for a special package

### 8.68.103 PKG\_getClientPackages

**Description:** returns an array or a space separated list of all packages installed on a client

**Parameter:**

- client: the name of the client
- key: if it is not empty only packages that contain the key are returned
- arr: set to true if the result should be an array otherwise it's a string
- status: If set only returns packages of the given status

### 8.68.104 PKG\_getPackagesListMarker

**Description:** returns the string to mark client names to store packages

### 8.68.105 PKG\_savePackagesList

**Description:** stores the package names in the DB

**Parameter:**

- listName: name of the list to store the packages
- packages: array or blank separated list of packages

### 8.68.106 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.

### 8.68.107 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.

### 8.68.108 PKG\_loadPackagesList

**Description:** returns an array or a blank separated 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 array otherwise it's a string

### 8.68.109 PKG\_deletePackagesList

**Description:** deletes a packages list

**Parameter:**

- listName: name of the list to delete



**8.68.110 PKG\_addNormalJob**

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

**Parameter:**

- client: the name of the client, the jobs are for
- packageName: name of the normal package
- priority: The priority of the job.

**8.68.111 PKG\_recountAllClientPackages**

**Description:** Recounts all packages of a client and updates corresponding fields in the database.

## 8.69 `./inc/pdf.php`

Author: Hauke Goos-Habermann (HHabermann@pc-kiel.de)

Description: Functions for generating PDF files

### 8.69.1 `PDF_showTableRow`

**Description:** Shows a table row with a variable amount of entries. The parameters are shown side by side as rows in a table. If more than one `PDF_showTableRow` commands are executed in one table it is needed to always use the same amount of paramaters in each call.

**Parameter:**

- Arbitrary amount of cells to show in a table.

### 8.69.2 `PDF_showTableHeader`

**Description:** Inits some values for starting a new PDF table.

### 8.69.3 `PDF_showTableEnd`

**Description:** Prints the PDF table.

### 8.69.4 `PDF_init`

**Description:** Inits some basic variables for PDF creation.

**Parameter:**

- orientation: Orientation of the PDF document (P or Portrait, L or Landscape)
- unit: pt (point), mm (millimeter), cm (centimeter) or in (inch)
- format: PDF page format A3, A4, A5, Letter or Legal

### 8.69.5 `PDF_output`

**Description:** Shows the created PDF.

## 8.70 *./inc/plugin.php*

Author: Hauke Goos-Habermann (HHabermann@pc-kiel.de)

Description: functions dealing with (de)installation of plugins and status information.

### 8.70.1 *PLG\_listMenuPlugins*

**Description:** generate the menu entries for the plugins

**Parameter:**

- path: the path you want to scan for plugins

### 8.70.2 *PLG\_isPluginSelected*

**Description:** checkes if the plugin was clicked

**Parameter:**

- path: path where to find the plugin files
- value: value the menu item should have to load the plugin page

### 8.70.3 *PLG\_isPluginSelected*

**Description:** gets values from the plugins like plugin name, version,...

**Parameter:**

- fileName: file name with whole path to the plugin file
- var: variable you want to get the value from

### 8.70.4 *PLG\_getPLGName*

**Description:** gets the name of the plugin

**Parameter:**

- fileName: file name with whole path to the plugin file

### 8.70.5 *PLG\_getPLGPage*

**Description:** gets the page of the plugin

**Parameter:**

- fileName: file name with whole path to the plugin file

### 8.70.6 *PLG\_getPLGAuthor*

**Description:** gets the author of the plugin

**Parameter:**

- fileName: file name with whole path to the plugin file

### 8.70.7 *PLG\_getPLGUpdateURL*

**Description:** gets the update address of the plugin

**Parameter:**

- fileName: file name with whole path to the plugin file

### 8.70.8 PLG\_getPLGClientRequires

**Description:** gets the "client requires" packages of the plugin

**Parameter:**

- fileName: file name with whole path to the plugin file

### 8.70.9 PLG\_getPLGVersion

**Description:** gets the version of the plugin

**Parameter:**

- fileName: file name with whole path to the plugin file

### 8.70.10 PLG\_showDownloadStatus

**Description:** shows the status of the plugin download

**Parameter:**

- fileName: file name with whole path to the plugin file

### 8.70.11 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 a valid url.
- tempDir: where to store the plugin file temporary
- fileName: file name with whole path to the plugin file

### 8.70.12 PLG\_checkOverwriting

**Description:** checks if the plugin files would overwrite existing files. the plugin file is extracted to a temporary directory. all file names are logged to a file that contains only the file names. these file names are checked against currently installed files. this routine checks if current files would be overwritten by the files of the plugin package. a list of files that would be overwritten is generated and aligned by a table.

**Parameter:**

- logfile: filename with whole path of the logfile containing the file names of the plugin file
- tempDir: where to store the plugin file temporary

### 8.70.13 PLG\_DBInstall

**Description:** stores information about the plugin in the data base.

**Parameter:**

- tempDir: where to find the extracted files of the plugin
- files: all file names of the plugin file name

### 8.70.14 PLG\_realInstall

**Description:** does the real installation

**Parameter:**

- tempDir: where to find the extracted files of the plugin

### 8.70.15 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 a valid url.

### 8.70.16 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 a valid url.

### 8.70.17 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 a valid url.

### 8.70.18 PLG\_showPluginOverview

**Description:** shows a overview of all plugins

### 8.70.19 PLG\_uninstall

**Description:** deletes a plugin

**Parameter:**

- name: name of the plugin

### 8.70.20 PLG\_getUpdateFile

**Description:** gets the update info file

**Parameter:**

- name: name of the plugin

### 8.70.21 PLG\_update

**Description:** initializes the update, shows information about the plugin update

**Parameter:**

- name: name of the plugin

### 8.70.22 PLG\_listInfofile

**Description:** lists information of a plugin update file

**Parameter:**

- name: name of the plugin

### 8.70.23 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 a valid url.

## 8.71 *./inc/pool.php*

Author: Hauke Goos-Habermann (HHabermann@pc-kiel.de)

Description: functions for administrating package pools

### 8.71.1 *POOL\_selectPoolType*

**Description:** shows buttons for selecting the type of pool and returns the pressed button

### 8.71.2 *POOL\_getPools*

**Description:** returns an array with all pool names

### 8.71.3 *POOL\_showLoadDeleteCreate*

**Description:** shows a dialog for loading, deleting and creating a pool

**Parameter:**

- poolName: name of the pool

### 8.71.4 *POOL\_create*

**Description:** creates a new pool directory and type property file

**Parameter:**

- poolName: name of the pool
- poolType: type of the pool (cd or download)
- poolArch: CPU architecture for the packages

### 8.71.5 *POOL\_setProperty*

**Description:** sets the contents of a property file

**Parameter:**

- poolName: name of the pool
- property: name of the pool property
- value: value to write in the pool property file

### 8.71.6 *POOL\_getProperty*

**Description:** returns the contents of a property file

**Parameter:**

- poolName: name of the pool
- property: name of the pool property

### 8.71.7 *POOL\_delete*

**Description:** deletes a pool

**Parameter:**

- poolName: name of the pool

### 8.71.8 POOL\_showReadCD

**Description:** shows a dialog for copying the CD contents to the pool

**Parameter:**

- poolName: name of the pool

### 8.71.9 POOL\_readCD

**Description:** copys the CD contents to the pool

**Parameter:**

- poolName: name of the pool
- mountPoint: the mount point of the CD drive

### 8.71.10 POOL\_createExtendedPackageIndex

**Description:** creates the Packages\* index files for the pool

**Parameter:**

- poolName: name of the pool

### 8.71.11 POOL\_showCreatePackageIndex

**Description:** shows information (status of the Packages\* generation, sources.list) about the currently generated pool

### 8.71.12 POOL\_getSize

**Description:** returns the size of a pool in MB

**Parameter:**

- poolName: name of the pool

### 8.71.13 POOL\_getDir

**Description:** returns the directory of the pool

**Parameter:**

- poolName: name of the pool

### 8.71.14 POOL\_download

**Description:** shows error messages if the checks for distribution, sourceslist or packages are failing. Otherwise starts the distribution specific download routine.

**Parameter:**

- poolName: name of the pool
- distr: name of the distribution
- sourceslist: list of the package sources
- release: release branch of the choosen distribution to download the packages from
- downloadBasePackages: set to true if a bunch of basic packages should be included into the pool
- arch: download the packages for a specific CPU architecture



### 8.71.15 POOL\_showDownloadStatus

**Description:** shows the package download status of a pool

**Parameter:**

- poolName: name of the pool

### 8.71.16 POOL\_prepare

**Description:** Generates the needed configuration file for reprepro.

**Parameter:**

- poolName: name of the pool
- release: release of the distribution (e.g. sarge)
- distr: name of the distribution (e.g. Debian)
- arch: CPU architecture for the packages

### 8.71.17 POOL\_makeRepository

**Description:** Generates a package source from packages stored in one directory.

**Parameter:**

- poolName: name of the pool
- archivPath: start search for packages in this subdirectory
- addCmds: additional commands that should be executed before starting the screen
- 

### 8.71.18 POOL\_getCDDistributionRelease

**Description:** Reads the distribution and the release name from a mounted CD and writes these information to the variables.

**Parameter:**

- mountPoint: the directory where the CD is mounted
- distr: the variable the name of the distribution (e.g. Debian) should be written to
- release: the variable release of the distribution (e.g. sarge) should be written to

### 8.71.19 POOL\_showSourcesList

**Description:** Shows the sources list of a selected package source.

**Parameter:**

- poolName: name of the pool

## 8.72 ./inc/preferences.php

Author: Hauke Goos-Habermann (HHabermann@pc-kiel.de)

Description: functions to save and load preferences for client setup.

### 8.72.1 PREF\_preferenceLoadManagerHandler

**Description:** Executes loading and deletion of preferences after pressing the according buttons and defines the buttons for PREF\_showPreferenceManager();

### 8.72.2 PREF\_preferenceSaveManagerHandler

**Description:** Executes the saving of preferences.

### 8.72.3 PREF\_showPreferenceManager

**Description:** Shows a dialog to load and delete existing preferences and to create new preferences.

### 8.72.4 PREF\_saveAllPreferenceValues

**Description:** Saves all values of a session into the preference.

### 8.72.5 PREF\_loadAllPreferenceValues

**Description:** Loads all values of a preference into the session.

### 8.72.6 PREF\_getClientPreferences

**Description:** list all preferences

**Parameter:**

- default: The name of the preference to list first
- directOutput: If enabled the preference names will be given out as a HTML option list. If disabled an array with the preference names as key and value will be returned.

**Returns:**

- Array with the preference names or nothing on enabled directOutput.

### 8.72.7 PREF\_getValue

**Description:** gets a value from a selected preference. with preferences you can store variables and values for reuse.

**Parameter:**

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

### 8.72.8 PREF\_putValue

**Description:** stores a value to a selected preference. with preferences you can store variables and values for reuse.

**Parameter:**

- name: the name of the preference
- var: variable of the preference
- value: value you want to set

### 8.72.9 PREF\_delete

**Description:** deletes a preference

**Parameter:**

- name: the name of the preference

### 8.72.10 PREF\_exists

**Description:** checks if a preference with the selected name exists

**Parameter:**

- name: the name of the preference

### 8.72.11 PREF\_putAllOptions

**Description:** stores all settings in the options array to the preferences

**Parameter:**

- prefName: name of the preference the options should be stored under
- options: the array with the options

### 8.72.12 PREF\_getAllValues

**Description:** gets all preferences and adds them to the options array

**Parameter:**

- prefName: name of the preference the options should be stored under
- options: the array with the options

## 8.73 `./inc/remotovar.php`

Author: Hauke Goos-Habermann (HHabermann@pc-kiel.de)

Description: functions for saving and loading serverside variables.

### 8.73.1 `RMV_exists4IP`

**Description:** checks if a variable exists for a certain IP

**Parameter:**

- var: name of the variable to store
- ip: ip address to set the variable for

### 8.73.2 `RMV_set4IP`

**Description:** creates or updates a variable for a special ip

**Parameter:**

- var: name of the variable to store
- value: value to set
- ip: ip address to set the variable for

### 8.73.3 `RMV_get4IP`

**Description:** gets the value of a variable for the given ip

**Parameter:**

- var: name of the variable to get the value from
- ip: ip you want to get the value for

### 8.73.4 `RMV_set`

**Description:** creates or updates a variable for the ip of the calling client

**Parameter:**

- var: name of the variable to store
- value: value to set

### 8.73.5 `RMV_get`

**Description:** gets the value of a variable for the ip of the calling client

**Parameter:**

- var: name of the variable to get the value from

### 8.73.6 `RMV_rm4IP`

**Description:** removes a variable for a selected ip

**Parameter:**

- var: name of the variable to get the value from
- ip: ip you want to delete the value from

### 8.73.7 RMV\_rm

**Description:** removes a variable for the ip of the calling client

**Parameter:**

- var: name of the variable to get the value from

### 8.73.8 RMV\_rm\_old

**Description:** removes all vars older than \$time seconds

**Parameter:**

- time: time in seconds

## 8.74 `./inc/scredit.php`

Author: Hauke Goos-Habermann (HHabermann@pc-kiel.de)

Description: Functions for the embedded script editor.

### 8.74.1 `SCREDIT_correctScriptFilename`

**Description:** Corrects the given filename to have it a valid prefix and suffix.

**Parameter:**

- filename: The filename to check and correct.

### 8.74.2 `SCREDIT_showEditor`

**Description:** Shows a script editor with syntax highlighting if JavaScript is enabled or a normal textarea input dialog.

### 8.74.3 `SCREDIT_newScriptTemplate`

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

**Returns:**

- Text of the script template.

## 8.75 `./inc/serverBackup.php`

Author: Hauke Goos-Habermann (HHabermann@pc-kiel.de)

Description: Functions for handling m23 server backups.

### 8.75.1 `SERVERBACKUP_getBackupConfiguration`

**Description:** Gets the server backup configuration.

**Returns:**

- Associative array with the variable name as key and its value as value.

### 8.75.2 `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.

### 8.75.3 `SERVERBACKUP_showConfigurationDialog`

**Description:** Shows a dialog for configuring the server backup.

### 8.75.4 `SERVERBACKUP_runBackupNowDialog`

**Description:** Shows a dialog for starting the server backup manually at once.

### 8.75.5 `SERVERBACKUP_getBackupList`

**Description:** Generates a list of existing server backups.

**Returns:**

- Associative array with information about sizes and dates of the backups.

### 8.75.6 `SERVERBACKUP_rmBackup`

**Description:** Removes server backup.

**Parameter:**

- `name`: Name of the backup (that is a in form of YYYY-MM-DD-HH-MM)

### 8.75.7 `SERVERBACKUP_backupOverviewDialog`

**Description:** Shows a dialog with overview of all existing server backups with possibility for deletion.

## 8.76 `./inc/server.php`

Author: Hauke Goos-Habermann (HHabermann@pc-kiel.de)

Description: functions for getting information from the server

### 8.76.1 `SERVER_importGPGPackageSignKey`

**Description:** Imports the m23 GPG package sign key.

### 8.76.2 `SERVER_logLocalScreenSessionToFile`

**Description:** Logs the output of a local screen session to a file.

**Parameter:**

- session: name screen session to log.
- user: user the script runs under

**Returns:**

- Full path to the log file.

### 8.76.3 `SERVER_setSSLCertCheckDisabled`

**Description:** Sets, if the SSL certificate check is disabled globally for all clients.

**Parameter:**

- : disableSSLCertCheck: true, if the check is disabled otherwise false.

### 8.76.4 `SERVER_isSSLCertCheckDisabled`

**Description:** Determines, if the SSL certificate check is disabled globally for all clients.

**Returns:**

- : true, if the check is disabled otherwise false.

### 8.76.5 `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.

### 8.76.6 `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.

### 8.76.7 `SERVER_isInstallReasonEnabled`

**Description:** Determines, if the install reason feature is enabled

**Returns:**

- : true, if the install reason feature is enabled, false otherwise



### 8.76.8 SERVER\_setInstallReasonEnabled

**Description:** Enables the install reason feature

**Parameter:**

- \$enabled true, if the install reason feature shall be enabled, false otherwise

### 8.76.9 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.

### 8.76.10 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.

### 8.76.11 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.

### 8.76.12 SERVER\_startUpdateClientOnlineInDBBackgroundJob

**Description:** Starts a background job to update the online status of the clients cyclically.

### 8.76.13 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.

### 8.76.14 SERVER\_setServerBoolSetting

**Description:** Sets a bool value of a server setting.

**Parameter:**

- : var: Name of the setting.
- : val: true or false.

### 8.76.15 SERVER\_getServerBoolSetting

**Description:** Gets the boolean value of a server setting.

**Parameter:**

- : var: Name of the setting.

**Returns:**

- : true or false.

### 8.76.16 SERVER\_setServerSetting

**Description:** Sets the value of a server setting.

**Parameter:**

- : var: Name of the setting.
- : val: Value of the setting.

### 8.76.17 SERVER\_getServerSetting

**Description:** Gets the value of a server setting.

**Parameter:**

- : var: Name of the setting.

**Returns:**

- : Value the setting.

### 8.76.18 SERVER\_existsServerSetting

**Description:** Checks, if a named server setting exists in the DB.

**Parameter:**

- : var: Name of the setting.

**Returns:**

- : True if the setting exists.

### 8.76.19 SERVER\_killPID

**Description:** Kills a process running under a given PID or sends a signal.

**Parameter:**

- : pid: The PID of the process to kill.
- : signal: The signal to send to the process.

**Returns:**

- : True if the process was killed or got the signal.

### 8.76.20 SERVER\_killBackgroundJob

**Description:** Kills a job (that runs in screen) with a given name.

**Parameter:**

- job: Name of the job that should be killed.
- user: User the job runs under.

### 8.76.21 SERVER\_insertLineNumber

**Description:** Inserts a text AT or AFTER a line number or creates a new file with the given name, if it doesn't exists.

**Parameter:**

- file: the name of the file
- lineNumber: reference line number for inserting

- insertText: text to insert
- mode: The access mode the newly created file should have.
- insertMode: "0" insert AT, "1" insert AFTER line number
- addIfNotExists: set to true, if the the line should be added only if the line doesn't exist. false, if the line should be added on every execution.

### 8.76.22 SERVER\_addAdmin

**Description:** Adds an administrator with all access rights.

**Parameter:**

- newadmin: Name of the new admin to create.
- password: Password for the admin account.

**Returns:**

- : true, if the deletion was sucessfully otherwise false.

### 8.76.23 SERVER\_delAdmin

**Description:** Deletes an administrator with all access rights.

**Parameter:**

- name: Name of the admin to delete.

**Returns:**

- : true, if the deletion was sucessfully otherwise false.

### 8.76.24 SERVER\_fileExists

**Description:** Checks if a file exists, that the Apache user has never access to.

**Parameter:**

- : file: Name (with full path) of the file to check.

**Returns:**

- : True, if the file exists other wise false.

### 8.76.25 SERVER\_getPublicSSHKeyOfm23Server

**Description:** Returns the public SSH key of the m23 server.

**Returns:**

- : Public SSH key of the m23 server.

### 8.76.26 SERVER\_changeHtpasswd

**Description:** Changes the password of a user in a htpasswd file.

**Parameter:**

- : htpasswdFile: htpasswd file that contains user names and crypted passwords.
- : username: Name of the user to change
- : password: The according new password

**Returns:**

- : True if the password was changed sucessfully.

### 8.76.27 SERVER\_delFromHtpasswd

**Description:** Removes a user with password to a htpasswd file.

**Parameter:**

- : htpasswdFile: htpasswd file that contains user names and crypted passwords.
- : username: Name of the user to remove

**Returns:**

- : True if the new user was added sucessfully.

### 8.76.28 SERVER\_addToHtpasswd

**Description:** Adds a new user with password to a htpasswd file.

**Parameter:**

- : htpasswdFile: htpasswd file that contains user names and crypted passwords.
- : username: Name of the new user to add
- : password: The according password for the new user

**Returns:**

- : True if the new user was added sucessfully.

### 8.76.29 SERVER\_dhcpServerInNetWarn

**Description:** Shows an error message if there is found another DHCP server on the net.

**Returns:**

- : false, if the IP address is static.

### 8.76.30 SERVER\_sendScriptToSF

**Description:** Uploads a script to m23.sf.net for public use.

**Parameter:**

- name: Name of the script.
- author: Name of the script author.
- description: Short descriptive text for the purpose of the script.
- script: Source code of the script.

### 8.76.31 SERVER\_dynamicIPWarn

**Description:** Shows an error message if the m23 server has a dynmic IP address.

**Returns:**

- : false, if the IP address is static.

### 8.76.32 SERVER\_tmpNotWritable

**Description:** Shows an error message if /tmp is not writable.

**Returns:**

- : false, if /tmp is writable.

### 8.76.33 SERVER\_rootFreeSpace

**Description:** Shows an error message if the free space of the root partition is low.

**Returns:**

- : false, if there is enough space.

### 8.76.34 SERVER\_isProgramRunning

**Description:** checks if a certain program is running and returns true, if yes "no" otherwise

**Parameter:**

- progname: the name of the programm (e.g. "apache" for the Apache web server)

### 8.76.35 SERVER\_checkPackageInstalled

**Description:** checks if a certain package is installed

**Parameter:**

- pkgName: the name of the package

### 8.76.36 SERVER\_daemonStartStop

**Description:** starts, stops and restarts daemons

**Parameter:**

- daemonScript: the file name of the script, that handles the real starting, stopping and restarting and understands the \$action
- action: start, stop or restart

**Returns:**

- : true on successfully execution otherwise false.

### 8.76.37 SERVER\_getAptGetInstallCommand

**Description:** Returns the apt-get commands to install a tool on the server.

**Parameter:**

- pkgName: name of the software package

**Returns:**

- : apt-get commands to install a tool on the server.

### 8.76.38 SERVER\_installTool

**Description:** installs a tool on the server

**Parameter:**

- pkgName: name of the software package

**Returns:**

- : true on successfully execution otherwise false.

### 8.76.39 SERVER\_installToolInBackground

**Description:** Installs a tool on the server in background.

**Parameter:**

- pkgName: name of the software package

### 8.76.40 SERVER\_programmStatus

**Description:** shows a row with information about the status of a certain program, with the possibility to start, stop or restart the program.

**Parameter:**

- progname: the name of the programm (e.g. "apache" for the Apache web server)
- daemonScript: set it to the script that should be used for starting, stopping and restarting. If the script name isn't set, this is a normal tool and NOT a daemon.
- canBeInstalled: set to "true" if the programm can be installed by the package name

### 8.76.41 SERVER\_apacheInfo

**Description:** returns an information string for the Apache server

### 8.76.42 SERVER\_mysqlInfo

**Description:** returns an information string for the MySQL server

### 8.76.43 SERVER\_dhcpInfo

**Description:** returns an information string for the DHCP server

### 8.76.44 SERVER\_LDAPInfo

**Description:** Returns an information string for the LDAP server.

### 8.76.45 SERVER\_programmStatusTableHeader

**Description:** shows the header of the table needed for the programm status lines

### 8.76.46 SERVER\_runInBackground

**Description:** Runs a script with "screen" in the background under a given user

**Parameter:**

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

### 8.76.47 SERVER\_runningInBackground

**Description:** Returns "true" if a lock file for a given job name is existing.

**Parameter:**

- jobName: name of the job

### 8.76.48 SERVER\_runningInScreen

**Description:** Returns "true" if a screen session with a given name exists for a given user.

**Parameter:**

- jobName: name of the job.
- user: User the screen session is run under.

### 8.76.49 SERVER\_addLineToFile

**Description:** Adds (if the search pattern can't be found) a line to a file on the server

**Parameter:**

- file: name of the file to edit
- search: the search pattern
- add: the line to add

**Returns:**

- : true on successfully execution otherwise false.

### 8.76.50 SERVER\_deleteFile

**Description:** Deletes a file from the server.

**Parameter:**

- fileName: Name of the file to delete.

**Returns:**

- : true on successfully execution otherwise false.

### 8.76.51 SERVER\_getFileContents

**Description:** Get the contents of any file (even if only readable by root).

**Parameter:**

- fileName: Name of the file to read.

**Returns:**

- : Contents of the file

### 8.76.52 SERVER\_putFileContents

**Description:** Stores a text to a file and changes it's mode, user and group.

**Parameter:**

- fileName: Name of the file to put the text to.
- text: The contents the file should have.
- mode: The access mode the file should have.
- user: The owner of the file.
- group: The owning group of the file.

**Returns:**

- true on success and false otherwise.

### 8.76.53 SERVER\_delLineFromFile

**Description:** Deletes lines from the file that match the search pattern

**Parameter:**

- file: name of the file to edit
- search: the search pattern

**Returns:**

- : true on successfully execution otherwise false.

### 8.76.54 SERVER\_addEtcHosts

**Description:** Adds a host to /etc/hosts and /etc/backuppc/hosts (if it doesn't exists already)

**Parameter:**

- hostname: name of the host to add
- ip: its IP

### 8.76.55 SERVER\_delEtcHosts

**Description:** Deletes a host entry from /etc/hosts and /etc/backuppc/hosts

**Parameter:**

- hostname: name of the host to delete

### 8.76.56 SERVER\_getInstallationMedium

**Description:** Tries to figure out how the m23 server was installed

**Returns:**

- : CD, Internet or Unknown source.

### 8.76.57 SERVER\_getOS

**Description:** Returns the version string of the distribution.

**Returns:**

- : version string.

### 8.76.58 SERVER\_checkDownload

**Description:** Downloads a special file from m23.sf.net and checks if the size and md5 sum are matching.

**Parameter:**

- : useProxy: Set to true if the local proxy should be used.

**Returns:**

- : Status information if file size and md5 sum are matching.

### 8.76.59 SERVER\_checkDiskFree

**Description:** Reports the free space of all mounted media.

**Returns:**

- : Output of "df".



### 8.76.60 SERVER\_checkRunInVM

**Description:** Checks if the m23 server is executed in a virtual machine or on native hardware.

**Returns:**

- : VMWare, VirtualBox or native.

### 8.76.61 SERVER\_checkKernel

**Description:** Returns the kernel information string of "uname -a".

**Returns:**

- : Kernel information string.

### 8.76.62 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.

### 8.76.63 SERVER\_commandAvailable

**Description:** Checks, if a given command is available for the given user.

**Parameter:**

- user: user the command should be run under.

**Returns:**

- : true when the command is available otherwise false.

## 8.77 `./inc/sourceslist.php`

Author: Hauke Goos-Habermann (HHabermann@pc-kiel.de)

Description: function to generate the sources.list for the client

### 8.77.1 `SRCLST_getAddToFile`

**Description:** Returns addToFile paramters from the given sources list as an associative array, where file name and file contents are seperated.

**Parameter:**

- `sourceName`: The name of the package source list

**Returns:**

- : Associative array with file name and file contents (e.g. `[0] => Array ([file] => file1.txt, [text] => text1)`, `[1] => Array ([file] => file2.txt, [text] => text2)`, ...)

### 8.77.2 `SRCLST_getRelease`

**Description:** Gets a release from the sourceslist table.

**Parameter:**

- `name`: the name of the package source list

**Returns:**

- Release name of choosen sources list.

### 8.77.3 `SRCLST_genList`

**Description:** generates the sources.list file for the client

**Parameter:**

- `clientName`: the name of the client

### 8.77.4 `SRCLST_saveArchitectures`

**Description:** Saves the architectures for package source list.

**Parameter:**

- `sourceName`: the name of the package source list
- `archs`: Associative array with the supported CPU architectures.

### 8.77.5 `SRCLST_saveList`

**Description:** saves the package source list

**Parameter:**

- `name`: the name of the package source list
- `list`: the list of sources as simple text
- `description`: a descriptive text for the list
- `distr`: the name of the distribution the list is for
- `release`: the name of the release the list is for

### 8.77.6 SRCLST\_querySourceslists

**Description:** returns the result of the DB query after sourceslists for a special distribution

**Parameter:**

- distr: the distribution the sources list is for or "\*" for all distributions

### 8.77.7 SRCLST\_genSelection

**Description:** generates a HTML selection with the names of alls package sources

**Parameter:**

- selName: the name of the selection
- first: the package source that should be shown first
- distr: the distribution the sources list is for or "\*" for all distributions

### 8.77.8 SRCLST\_getValue

**Description:** gets a value from the sourceslist table

**Parameter:**

- name: the name of the package source list
- var: the name of the table row

### 8.77.9 SRCLST\_loadSourceListFromDB

**Description:** loads and returns the the package source list from the DB.

**Parameter:**

- name: the name of the package source list

### 8.77.10 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.

### 8.77.11 SRCLST\_loadSourceList

**Description:** Loads and returns the package source list and tries to find a valid mirror for m23debs.

**Parameter:**

- name: the name of the package source list

**Returns:**

- package source list

### 8.77.12 SRCLST\_getDescription

**Description:** returns the the package source description

**Parameter:**

- name: the name of the package source list

### 8.77.13 SRCLST\_delete

**Description:** deletes package source

**Parameter:**

- name: the name of the package source list

### 8.77.14 SRCLST\_checkList

**Description:** checks a package info and returns the output of the OS package update function

**Parameter:**

- sourceName: the name of the package source list

### 8.77.15 SRCLST\_packageInformationChangeInformationHumanReadable

**Description:** Returns the time point when the package information was changed last.

**Parameter:**

- distr: the short name of the distribution
- sourceName: the name of the package source list

**Returns:**

- : Time when the package information was changed last.

### 8.77.16 SRCLST\_packageInformationChangeTime

**Description:** Returns the time point when the package information was changed last.

**Parameter:**

- distr: the short name of the distribution
- sourceName: the name of the package source list
- changedBefore: Amount of seconds before the package information was changed.

**Returns:**

- : Time when the package information was changed last.

### 8.77.17 SRCLST\_packageInformationOlderThan

**Description:** Checks if a package info is older than a selected amount of minutes or if the package info directory is too smal.

**Parameter:**

- minutes: the amount of minutes the package information can be older to return true
- distr: the short name of the distribution
- sourceName: the name of the package source list

**Returns:**

- : true when package info is older than a selected amount of minutes or if the package info directory is too smal, otherwise false.

### 8.77.18 SRCLST\_getStorageFS

**Description:** Returns a file systems that can be used to install the OS and to store data. A wished file system is given and an alternative FS is returned, if this FS is not supported.

**Parameter:**

- fs: File system to probe.
- sourceName: The name of the package source list

**Returns:**

- : File systems that can be used to install the OS and to store data

### 8.77.19 SRCLST\_supportedFS

**Description:** Returns an array with file systems that supported by the OS.

**Parameter:**

- sourceName: The name of the package source list

**Returns:**

- : Array with file systems supported by the OS.

### 8.77.20 SRCLST\_alternativeFS

**Description:** Returns the alternative file system that is supported by the OS.

**Parameter:**

- sourceName: The name of the package source list

**Returns:**

- : File system.

### 8.77.21 SRCLST\_getParameter

**Description:** Returns special parameter(s) from the given sources list.

**Parameter:**

- sourceName: The name of the package source list
- parameter: The name of the parameter.

**Returns:**

- : Values for the given parameter in an array.

### 8.77.22 SRCLST\_getMirror

**Description:** returns the mirror from the sources list

**Parameter:**

- sourceName: the name of the package source list

**Returns:**

- URL to the mirror

### 8.77.23 SRCLST\_getDesktopList

**Description:** returns an array with all supported desktops

**Parameter:**

- sourceName: the name of the package source list

### 8.77.24 SRCLST\_showDesktopsSel

**Description:** returns a selections with all desktops supported by the sources list

**Parameter:**

- sourceName: the name of the package source list
- selName: the name of the selection
- first: the desktop that should be shown first

### 8.77.25 SRCLST\_doesDistrSupportEFI

**Description:** Checks, if a sources list contains a distribution that supports EFI.

**Parameter:**

- sourceName: the name of the package source list

**Returns:**

- true, if the distribution supports EFI, otherwise false.

### 8.77.26 SRCLST\_getListnamesWithEfiSupport

**Description:** Gets a list with all sources lists that support EFI.

**Returns:**

- Array with all sources lists that support EFI.

### 8.77.27 SRCLST\_clientUsesEfiButSourcesListDoesntSupportEfi

**Description:** Checks, if the client uses EFI and the choosen sources list doesn't.

**Parameter:**

- client: Name of the client.
- sourceName: The name of the package source list.

**Returns:**

- : true, if the client uses EFI and the choosen sources list doesn't, otherwise false.

### 8.77.28 SRCLST\_showErrorIfClientUsesEfiButSourcesListDoesntSupportEfi

**Description:** Shows an error message, if the client uses EFI and the choosen sources list doesn't.

**Parameter:**

- client: Name of the client.
- sourceName: The name of the package source list.

**Returns:**

- : false, if the client uses EFI and the choosen sources list doesn't, otherwise true.

### 8.77.29 SRCLST\_showAlternativeArchitectureSelection

**Description:** Shows a list with available CPU architectures of the sources list, in case that the wanted architecture is not available in the sources list. The alternative architecture will be written to the arch option of the client.

**Parameter:**

- sourceName: The name of the package source list.
- wantedArch: The CPU architecture of the m23 client.
- client: Name of the client.

**Returns:**

- : A CPU architecture supported by the package source list.

### 8.77.30 SRCLST\_isArchAvailable

**Description:** Checks if a given architecture is supported by the sources list.

**Parameter:**

- sourceName: the name of the package source list
- arch: Architecture to check for.

**Returns:**

- true, if the architecture is supported, false otherwise.

### 8.77.31 SRCLST\_getArchitectures

**Description:** Returns a list of all CPU architectures supported by the sources list.

**Parameter:**

- sourceName: the name of the package source list

**Returns:**

- Associative array with the supported CPU architectures as variable AND key.

### 8.77.32 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.

### 8.77.33 SRCLST\_getListnames

**Description:** Returns an array that contains all sourceslist names

**Parameter:**

- distr: the distribution the sources list is for or "\*" for all distributions

### 8.77.34 SRCLST\_cleanList

**Description:** Returns an array with all lines of the sources list that contain Debian sources

**Parameter:**

- list: the contents of the sources list

### 8.77.35 SRCLST\_matchList

**Description:** Returns the name of the sources list that matches the searched sources list contents for the distribution or false

**Parameter:**

- distr: the distribution to search the name of the sources list under
- search: the contents of the sources list to search

### 8.77.36 SRCLST\_possiblem23debsMirrors

**Description:** Returns an array with mirrors for m23 debs.

**Returns:**

- Array with mirrors for m23 debs.

### 8.77.37 SRCLST\_checkm23debsMirror

**Description:** Checks, if the url contains a valid mirror for m23debs.

**Parameter:**

- url: URL of the (possible) m23debs mirror.

**Returns:**

- true, if the url contains a valid mirror for m23debs, otherwise false.

### 8.77.38 SRCLST\_getWorkingm23debsMirror

**Description:** Get the url of a working m23debs mirror.

**Returns:**

- Url to a working m23debs mirror or false, if none could be found.



## 8.78 *./inc/ucs.php*

Author: Hauke Goos-Habermann (HHabermann@pc-kiel.de)

Description: Univention corporate server specific functions.

### 8.78.1 UCS\_getPrefnameByClient

**Description:** Generates the preference name for an UCS client.

**Parameter:**

- client: Client name.

**Returns:**

- Preference name for an UCS client

### 8.78.2 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.

### 8.78.3 UCS\_delUCSClientFromm23ClientPreferences

**Description:** Removes an UCS client from the client preferences in m23.

**Parameter:**

- client: Client name.

### 8.78.4 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)

### 8.78.5 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.

### 8.78.6 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.

### 8.78.7 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.

### 8.78.8 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.

### 8.78.9 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.

**Returns:**

- : Associative array with the information.

### 8.78.10 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.

### 8.78.11 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.

### 8.78.12 UCS\_getFirstElementFromDN

**Description:** Gets the first value of a distinguished name.

**Parameter:**

- dn: distinguished name.

**Returns:**

- : First value of a distinguished name.

### 8.78.13 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.

### 8.78.14 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.

### 8.78.15 UCS\_delClient

**Description:** Removes a client from the UCS LDAP.

**Parameter:**

- client: Client name.

### 8.78.16 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).

### 8.78.17 UCS\_disableClientPXEBoot

**Description:** Deactivates PXE booting of a client that is in the UCS LDAP via the univention-dhcp.

**Parameter:**

- client: Client name.

### 8.78.18 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.

### 8.78.19 UCS\_openFirewallPort

**Description:** Opens a port on the UCS firewall.

**Parameter:**

- port: Port number to open.
- type: tcp or udp.

### 8.78.20 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.

### 8.78.21 UCS\_getEtc\_ucr\_master

**Description:** Gets the contents for /etc/univention/ucr\_master on the client.

### 8.78.22 UCS\_enableClientLDAP

**Description:** Enables LDAP authentication on the client on the UCS.

### 8.78.23 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.

### 8.78.24 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.

## 8.79 *./inc/update.php*

Author: Hauke Goos-Habermann (HHabermann@pc-kiel.de)

Description: functions for updating the server

### 8.79.1 *UPDATE\_doUpdate*

**Description:** Makes sure the package repository is included and upgrades the m23 server.

### 8.79.2 *UPDATE\_running*

**Description:** checks, if an update is running (returns true otherwise false)

### 8.79.3 *UPDATE\_getUrl*

**Description:** returns a correct URL to the update source

**Parameter:**

- base: URL to the update script
- command: "info" or "cmd"
- version: m23 version
- patchLevel: patch version number

### 8.79.4 *UPDATE\_getInfo*

**Description:** returns the information text from the URL

**Parameter:**

- URL: URL to the information text
- refreshTime: The time in minutes the file is downloaded again.

## 8.80 `./inc/vm.php`

Author: Hauke Goos-Habermann (HHabermann@pc-kiel.de)

Description: Functions for managing virtual clients.

### 8.80.1 `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.

### 8.80.2 `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.

### 8.80.3 `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 occurred)

### 8.80.4 `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.

### 8.80.5 VM\_CloudStackCheckConstants

**Description:** Checks, if the given constant values are valid.

**Parameter:**

- CLOUDSTACK\_API\_ENDPOINT: The API endpoint.
- CLOUDSTACK\_API\_KEY: The API key.
- CLOUDSTACK\_SECRET\_KEY: The secret API key.

**Returns:**

- true, when the constant values are valid otherwise false.

### 8.80.6 VM\_CloudStackConfigGUI

**Description:** Shows a dialog for editing the CloudStack config file and uploading the m23 client ISO.

### 8.80.7 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.

### 8.80.8 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

### 8.80.9 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

### 8.80.10 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

### 8.80.11 VM\_CloudStackSendSetVisualURL

**Description:** Sends the visual URL (current client ip:22) to the m23 server, if run under CloudStack.

### 8.80.12 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

### 8.80.13 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



#### 8.80.14 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)

#### 8.80.15 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

#### 8.80.16 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

#### 8.80.17 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

#### 8.80.18 VM\_CloudStack\_getServerIP

**Description:** Gets the external m23 server IP if the m23 server is run as CloudStack VM.

**Returns:**

- External m23 server IP.

#### 8.80.19 VM\_GUIstepCreateCloudStackVM

**Description:** Shows a dialog to create a new VM in CloudStack.

### 8.80.20 VM\_CloudStack\_available

**Description:** Checks, if the CloudStack configuration file is included and contains the needed constants.

**Returns:**

- true, if the CloudStack are present.

### 8.80.21 VM\_CloudStack\_getObject

**Description:** Gets a new CloudStackClient object.

**Parameter:**

- CLOUDSTACK\_API\_ENDPOINT: The API endpoint.
- CLOUDSTACK\_API\_KEY: The API key.
- CLOUDSTACK\_SECRET\_KEY: The secret API key.

**Returns:**

- New CloudStackClient object.

### 8.80.22 VM\_CloudStack\_getVersion

**Description:** Gets the version of CloudStack.

**Returns:**

- CloudStack version.

### 8.80.23 VM\_shutdownAndDisableNetbootAfterInstall

**Description:** Reboots an VM and disables network booting.

**Parameter:**

- vmname: Name of the VM.

### 8.80.24 VM\_rebootAndActivateNetboot

**Description:** Reboots an VM and activates network booting.

**Parameter:**

- vmname: Name of the VM.

### 8.80.25 VM\_shutdownAndDisableNetbootAfterInstall

**Description:** Shuts down an VM and disables network booting.

**Parameter:**

- vmName: Name of the VM.

### 8.80.26 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.

### 8.80.27 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.

### 8.80.28 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.

### 8.80.29 VM\_setVBoxAddonAsDefault

**Description:** Sets a choosen VirtualBox addition package version as default.

**Parameter:**

- version: Version number of the VirtualBox addition to set as default.

### 8.80.30 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.

### 8.80.31 VM\_getVBoxAddonDefaultVersion

**Description:** Gets the version number of the VirtualBox addition package.

**Returns:**

- The default version of the VirtualBox addition package.

### 8.80.32 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.

### 8.80.33 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.

### 8.80.34 VM\_VBOXaddonDownloadDialog

**Description:** Shows a dialog for downloading the VirtualBox additions to the m23 server.

### 8.80.35 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.

### 8.80.36 VM\_listDownloadableVBoxAddons

**Description:** Returns an array with the version numbers of all VirtualBox addition ISOs that are 2.0.0 and above.

**Returns:**

- Array with the version numbers of all VirtualBox addition ISOs that are 2.0.0 and above as key and value.

### 8.80.37 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.

### 8.80.38 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.

### 8.80.39 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.

#### 8.80.40 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.

#### 8.80.41 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.

#### 8.80.42 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.

#### 8.80.43 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.

#### 8.80.44 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.

### 8.80.45 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.

### 8.80.46 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.

### 8.80.47 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 if the client is no VM guest.

### 8.80.48 VM\_GUIstepCreateGuest

**Description:** Shows a dialog to create a new VM on the chosen host.

### 8.80.49 VM\_GUIstepCheckHost

**Description:** Shows a dialog part with information about the chosen VM host.

### 8.80.50 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 chosen virtualisation solution.

### 8.80.51 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.

### 8.80.52 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.

### 8.80.53 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.

### 8.80.54 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.

### 8.80.55 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.

### 8.80.56 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.

### 8.80.57 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.

### 8.80.58 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 delete a virtual machine.

### 8.80.59 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 communicate 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.



### 8.80.60 VM\_insertBootISO

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

**Parameter:**

- type: VM\_SW\_VBOX for VirtualBox OSE
- vmName: Name of the VM.
- iso: ISO file with full path.

**Returns:**

- BASH code to insert a bootable ISO into a VM.

### 8.80.61 VM\_startVMInExistingXSession

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

**Parameter:**

- type: VM\_SW\_VBOX for VirtualBox OSE
- vmName: Name of the VM.

**Returns:**

- BASH code to start a virtual machine and finding the DISPLAY number of the user who runs this script.

### 8.80.62 VM\_startVM

**Description:** Starts a virtual machine.

**Parameter:**

- type: VM\_SW\_VBOX for VirtualBox OSE
- vmName: Name of the VM.
- vnc: Set to true if the VM should be accessible since the booting via VNC.

**Returns:**

- BASH code to start a virtual machine.

### 8.80.63 VM\_startVMCommandFile

**Description:** Writes a command file with the command(s) to start the VM.

**Parameter:**

- vmName: Name of the VM.
- cmd: Bash code to start the VM.

### 8.80.64 VM\_stopVMCommandFile

**Description:** Removes automatical starting of a VM by removing the command file.

**Parameter:**

- vmName: Name of the VM.

### 8.80.65 VM\_status

**Description:** Gets the current status of a virtual machine.

**Parameter:**

- type: VM\_SW\_VBOX for VirtualBox OSE
- vmName: Name of the VM.

**Returns:**

- BASH code to get the current status of a virtual machine or array containing the status of the VM.

### 8.80.66 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 if the medium is empty.

### 8.80.67 VM\_parseVBOXstate

**Description:** Parses the status (on, off, paused) line of VirtualBox.

**Parameter:**

- param: Parameter line that contains the status string of the VM.

**Returns:**

- VM\_STATE\_OFF, VM\_STATE\_PAUSE, VM\_STATE\_ON or false if the line could not be parsed.

### 8.80.68 VM\_parseVBOXNic

**Description:** Parses the status line of a virtual network card.

**Parameter:**

- param: Parameter line that contains the status string of the VM.

**Returns:**

- Array with the current state of the network device.

### 8.80.69 VM\_parseStatus

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

**Returns:**

- Array with the current state of the VM.