



Version 6.1

## Programers Guide

2016-05-30

# Table of Contents

1	Introduction.....	1
2	hyperCMS XML-Content-Repository .....	1
2.1	hyperCMS specific information .....	3
2.2	Meta-Information .....	4
2.3	Text .....	4
2.4	Media .....	5
2.5	Links .....	5
2.6	Components .....	6
2.7	Articles .....	6
3	Function libraries .....	7
3.1	Including a library .....	7
3.2	Loading the configuration .....	7
3.2.1	Content Management Server.....	7
3.2.2	Publication Server .....	8
3.3	Global variables .....	8
3.4	Object operation library .....	11
3.4.1	createfolder.....	11
3.4.2	deletefolder .....	12
3.4.3	renamefolder .....	13
3.4.4	createobject .....	14
3.4.5	deleteobject .....	15
3.4.6	renameobject .....	16
3.4.7	cutobject .....	17
3.4.8	copyobject .....	18
3.4.9	copyconnectedobject .....	19
3.4.10	pasteobject .....	20
3.4.11	lockobject .....	21
3.4.12	unlockobject.....	22
3.4.13	publishobject .....	23
3.4.14	unpublishobject .....	24
3.4.15	getlinkedobject .....	25
3.4.16	getconnectedobject .....	26
3.4.17	getobjectcontainer .....	27
3.4.18	loadcontainer.....	27
3.4.19	savecontainer .....	28
3.5	File Pointer library .....	29
3.5.1	getfilename .....	29
3.5.2	setfilename .....	30
3.6	File operation library.....	31
3.6.1	loadfile .....	31
3.6.2	savefile.....	31
3.6.3	loadlockfile .....	32
3.6.4	savelockfile .....	32
3.6.5	lockfile.....	33
3.6.6	unlockfile .....	33
3.6.7	deletefile.....	34
3.6.8	appendfile .....	34
3.7	Edit content library.....	35
3.7.1	setxmlparameter.....	35
3.7.2	getcontent .....	36
3.7.3	getxmlcontent .....	37
3.7.4	selectcontent .....	38
3.7.5	selectxmlcontent.....	39
3.7.6	deletecontent .....	40
3.7.7	setcontent.....	41

3.7.8	updatecontent .....	42
3.7.9	insertcontent .....	43
3.7.10	addcontent .....	44
3.8	Meta Data Generator library .....	45
3.8.1	getkeywords .....	45
3.8.2	getdescription .....	45
3.8.3	injectmetadata .....	46
3.9	Notifications library .....	47
3.9.1	licensenotification .....	47
4	Components and applications .....	48
5	Database Connectivity .....	49
5.1	Creating a Database Connectivity .....	49
6	Event System .....	51
7	hyperCMS API Function Reference .....	52
7.1	Main API Functions .....	52
7.1.1	correctnumber .....	52
7.1.2	cleancontent .....	52
7.1.3	convertchars .....	52
7.1.4	specialchr .....	53
7.1.5	specialchr_encode .....	53
7.1.6	specialchr_decode .....	53
7.1.7	is_utf8 .....	54
7.1.8	is_latin1 .....	54
7.1.9	makestring .....	54
7.1.10	splitstring .....	54
7.1.11	is_emptyfolder .....	54
7.1.12	is_supported .....	55
7.1.13	is_cloudstorage .....	55
7.1.14	is_cloudobject .....	55
7.1.15	is_date .....	56
7.1.16	is_tempfile .....	56
7.1.17	is_thumbnail .....	56
7.1.18	is_config .....	57
7.1.19	is_encryptedfile .....	57
7.1.20	is_document .....	57
7.1.21	is_image .....	58
7.1.22	is_rawimage .....	58
7.1.23	is_aiimage .....	58
7.1.24	is_video .....	59
7.1.25	is_audio .....	59
7.1.26	is_mobilebrowser .....	59
7.1.27	copyrecursive .....	60
7.1.28	correctfile .....	60
7.1.29	correctpath .....	60
7.1.30	convertpath .....	60
7.1.31	convertlink .....	61
7.1.32	deconvertpath .....	61
7.1.33	deconvertlink .....	62
7.1.34	createviewlink .....	62
7.1.35	createaccesslink .....	62
7.1.36	createwrapperlink .....	63
7.1.37	createdownloadlink .....	63
7.1.38	createmultidownloadlink .....	64
7.1.39	cleandomain .....	64
7.1.40	fileversion .....	65
7.1.41	createversion .....	65
7.1.42	rollbackversion .....	65
7.1.43	deleteversion .....	66

7.1.44	deleteversions .....	66
7.1.45	loadfile_header .....	66
7.1.46	loadfile_fast .....	67
7.1.47	loadfile .....	67
7.1.48	loadlockfile .....	68
7.1.49	savefile .....	68
7.1.50	savelockfile .....	68
7.1.51	appendfile .....	69
7.1.52	lockfile .....	69
7.1.53	unlockfile .....	70
7.1.54	deletefile .....	70
7.1.55	restoremediafile .....	70
7.1.56	preparemediafile .....	71
7.1.57	deletemediafiles .....	71
7.1.58	avoidfilecollision .....	71
7.1.59	substr_in_array .....	72
7.1.60	downloadobject .....	72
7.1.61	downloadfile .....	72
7.1.62	loadcontainer .....	73
7.1.63	savecontainer .....	73
7.1.64	getworkflowitem .....	74
7.1.65	checkworkflow .....	74
7.1.66	inherit_db_load .....	74
7.1.67	inherit_db_read .....	75
7.1.68	inherit_db_close .....	75
7.1.69	inherit_db_save .....	76
7.1.70	inherit_db_getparent .....	76
7.1.71	inherit_db_getchild .....	76
7.1.72	inherit_db_setparent .....	76
7.1.73	inherit_db_insertparent .....	77
7.1.74	inherit_db_deleteparent .....	77
7.1.75	getconfigvalue .....	77
7.1.76	createinstance .....	77
7.1.77	editinstance .....	78
7.1.78	deleteinstance .....	78
7.1.79	createpublication .....	78
7.1.80	editpublication .....	79
7.1.81	editpublicationsetting .....	79
7.1.82	deletepublication .....	80
7.1.83	createpersonalization .....	80
7.1.84	deletepersonalization .....	80
7.1.85	createtemplate .....	81
7.1.86	gettemplates .....	81
7.1.87	loadtemplate .....	82
7.1.88	edittemplate .....	82
7.1.89	deletetemplate .....	82
7.1.90	createuser .....	83
7.1.91	edituser .....	83
7.1.92	deleteuser .....	84
7.1.93	creategroup .....	84
7.1.94	editgroup .....	85
7.1.95	deletegroup .....	85
7.1.96	renamegroupfolder .....	86
7.1.97	deletegroupfolder .....	86
7.1.98	renameworkflowfolder .....	86
7.1.99	deleteworkflowfolder .....	87
7.1.100	createmediacat .....	87
7.1.101	renamemediacat .....	87

7.1.102	deletemediacat .....	88
7.1.103	uploadtomediacat .....	88
7.1.104	deletefrommediacat .....	89
7.1.105	createfolder .....	89
7.1.106	createfolders .....	89
7.1.107	collectfolders .....	90
7.1.108	copyfolders .....	90
7.1.109	deletefolder .....	91
7.1.110	renamefolder .....	91
7.1.111	createobject .....	92
7.1.112	uploadfile .....	92
7.1.113	createmediaobject .....	93
7.1.114	createmediaobjects .....	93
7.1.115	editmediaobject .....	94
7.1.116	manipulateobject .....	94
7.1.117	deleteobject .....	95
7.1.118	renameobject .....	95
7.1.119	renamefile .....	96
7.1.120	cutobject .....	96
7.1.121	copyobject .....	97
7.1.122	copyconnectedobject .....	97
7.1.123	pasteobject .....	98
7.1.124	lockobject .....	98
7.1.125	unlockobject .....	98
7.1.126	publishobject .....	99
7.1.127	processobjects .....	99
7.1.128	publishlinkedobject .....	100
7.1.129	unpublishobject .....	100
7.1.130	collectobjects .....	101
7.1.131	manipulateallobjects .....	101
7.1.132	remoteclient .....	102
7.1.133	HTTP_Post .....	102
7.1.134	HTTP_Get .....	102
7.1.135	HTTP_Proxy .....	103
7.1.136	loadbalancer .....	103
7.1.137	savelog .....	103
7.1.138	deletelog .....	104
7.1.139	debuglog .....	104
7.1.140	notifyusers .....	105
7.1.141	sendlicensenotification .....	105
7.1.142	licensenotification .....	106
7.1.143	html_diff .....	106
7.1.144	createfavorite .....	106
7.1.145	getfavorites .....	107
7.1.146	deletefavorite .....	107
7.1.147	setboxes .....	107
7.1.148	getboxes .....	107
7.1.149	rewrite_targetURI .....	108
7.1.150	rewrite_homepage .....	108
7.1.151	sendmessage .....	108
7.2	Get API Functions .....	109
7.2.1	getserverload .....	109
7.2.2	getSession .....	109
7.2.3	getRequest .....	109
7.2.4	getRequest_esc .....	109
7.2.5	getUserip .....	110
7.2.6	getlanguageoptions .....	110
7.2.7	getlanguagefile .....	110

7.2.8	getcodepage .....	111
7.2.9	getcalendarlang .....	111
7.2.10	getescapedtext .....	111
7.2.11	getsearchhistory .....	111
7.2.12	gettaxonomy_sublevel .....	112
7.2.13	gettaxonomy_childs .....	112
7.2.14	getkeywords .....	113
7.2.15	getdescription .....	113
7.2.16	getgooglesitemap .....	113
7.2.17	getmetadata .....	114
7.2.18	getobjectcontainer .....	114
7.2.19	getcontainer .....	114
7.2.20	getcontainername .....	115
7.2.21	getlocationname .....	115
7.2.22	getthemelocation .....	115
7.2.23	getcategory .....	116
7.2.24	getpublication .....	116
7.2.25	getlocation .....	116
7.2.26	getobject .....	116
7.2.27	getmediacontainername .....	117
7.2.28	getmediafileversion .....	117
7.2.29	getobjectid .....	117
7.2.30	getobjectlink .....	117
7.2.31	getcontainerversions .....	118
7.2.32	gettemplateversions .....	118
7.2.33	getfileinfo .....	118
7.2.34	getobjectinfo .....	119
7.2.35	getfilesize .....	119
7.2.36	getmimetype .....	119
7.2.37	getfiletype .....	120
7.2.38	getvideoinfo .....	120
7.2.39	getbrowserinfo .....	120
7.2.40	getcontentlocation .....	120
7.2.41	getmedialocation .....	121
7.2.42	getlockedfileinfo .....	121
7.2.43	getuseronline .....	121
7.2.44	getchatstate .....	122
7.2.45	getimagelib .....	122
7.2.46	getfilename .....	122
7.2.47	gethypertag .....	122
7.2.48	gethypertagname .....	123
7.2.49	gethtmltag .....	123
7.2.50	gethtmltags .....	123
7.2.51	getattribute .....	124
7.2.52	getoption .....	124
7.2.53	getcharset .....	124
7.2.54	getartid .....	124
7.2.55	getelementid .....	125
7.2.56	getfirstkey .....	125
7.2.57	getdirectoryfiles .....	125
7.2.58	getuserinformation .....	125
7.3	Set API Functions .....	126
7.3.1	setsession .....	126
7.3.2	settaxonomy .....	126
7.3.3	setarticle .....	126
7.3.4	settext .....	127
7.3.5	setmedia .....	127
7.3.6	setpagelink .....	128

7.3.7	setcomplink.....	128
7.3.8	sethead .....	129
7.3.9	setfilename .....	129
7.4	Connect API Functions .....	129
7.4.1	ftp_userlogin .....	129
7.4.2	ftp_userlogout .....	130
7.4.3	ftp_getfile .....	130
7.4.4	ftp_putfile .....	130
7.4.5	ftp_filelist.....	131
7.4.6	createsharelink_facebook .....	131
7.4.7	createsharelink_twitter .....	131
7.4.8	createsharelink_googleplus.....	131
7.4.9	createsharelink_linkedin.....	132
7.4.10	createsharelink_pinterest .....	132
7.5	Security API Functions .....	132
7.5.1	rootpermission.....	132
7.5.2	globalpermission .....	133
7.5.3	localpermission .....	133
7.5.4	accessgeneral .....	133
7.5.5	accesspermission .....	134
7.5.6	setlocalpermission .....	134
7.5.7	checkpublicationpermission.....	134
7.5.8	checkadminpermission .....	135
7.5.9	checkrootpermission .....	135
7.5.10	checkglobalpermission .....	135
7.5.11	checklocalpermission .....	136
7.5.12	userlogin.....	136
7.5.13	registerinstance .....	136
7.5.14	createchecksum .....	137
7.5.15	writesession .....	137
7.5.16	writesessiondata .....	137
7.5.17	createsession.....	138
7.5.18	killsession .....	138
7.5.19	checkdiskkey .....	138
7.5.20	checkpassword .....	138
7.5.21	loguserip.....	139
7.5.22	checkuserip .....	139
7.5.23	checkuserrequests.....	139
7.5.24	checkusersession .....	140
7.5.25	allowuserip.....	140
7.5.26	valid_objectname .....	140
7.5.27	valid_locationname.....	140
7.5.28	valid_publicationname .....	141
7.5.29	html_encode .....	141
7.5.30	html_decode .....	141
7.5.31	scriptcode_encode.....	142
7.5.32	scriptcode_extract.....	142
7.5.33	scriptcode_clean_functions .....	142
7.5.34	sql_clean_functions .....	143
7.5.35	url_encode .....	143
7.5.36	url_decode .....	143
7.5.37	shellcmd_encode.....	143
7.5.38	hcms_crypt .....	144
7.5.39	hcms_encrypt .....	144
7.5.40	hcms_decrypt .....	144
7.5.41	createtimetoken.....	145
7.5.42	checktimetoken .....	145
7.5.43	createtoken.....	145

7.5.44	checktoken.....	146
7.5.45	createunique token.....	146
7.5.46	rand_secure .....	146
7.6	Media API Functions .....	146
7.6.1	indexcontent .....	146
7.6.2	unindexcontent .....	147
7.6.3	createthumbnail_in design.....	147
7.6.4	createthumbnail_video.....	148
7.6.5	createmedia .....	148
7.6.6	convertmedia .....	149
7.6.7	convertimage .....	149
7.6.8	rotateimage .....	150
7.6.9	getimagecolors .....	150
7.6.10	getimagecolorkey.....	151
7.6.11	hex2rgb.....	151
7.6.12	rgb2hex.....	151
7.6.13	readmediaplayer_config .....	152
7.6.14	savemediaplayer_config.....	152
7.6.15	createdocument .....	152
7.6.16	unzipfile.....	153
7.6.17	zipfiles.....	153
7.6.18	px2mm.....	154
7.6.19	px2inch .....	154
7.6.20	inch2px .....	155
7.6.21	vtt2array .....	155
7.7	Metadata API Functions.....	155
7.7.1	createtaxonomy.....	155
7.7.2	splitkeywords .....	155
7.7.3	copymetadata.....	156
7.7.4	extractmetadata .....	156
7.7.5	xmlobject2array.....	156
7.7.6	id3_getdata.....	157
7.7.7	id3_writefile .....	157
7.7.8	id3_create.....	157
7.7.9	xmp_getdata .....	158
7.7.10	xmp_writefile .....	158
7.7.11	xmp_create.....	158
7.7.12	geo2decimal.....	159
7.7.13	exif_getdata .....	159
7.7.14	iptc_getdata .....	159
7.7.15	iptc_getcharset.....	159
7.7.16	iptc_maketag .....	160
7.7.17	iptc_writefile .....	160
7.7.18	iptc_create.....	161
7.7.19	createmapping.....	161
7.7.20	getmapping.....	161
7.7.21	setmetadata .....	161
7.8	Link API Functions.....	162
7.8.1	link_db_restore.....	162
7.8.2	link_db_load.....	162
7.8.3	link_db_read .....	163
7.8.4	link_db_close.....	163
7.8.5	link_db_save .....	163
7.8.6	link_db_update .....	164
7.8.7	link_db_insert.....	164
7.8.8	link_db_delete .....	165
7.8.9	link_db_getobject.....	165
7.8.10	link_update .....	165



7.8.11	getlinkedobject .....	166
7.8.12	getconnectedobject .....	166
7.8.13	extractlinks .....	166
7.9	Plugin API Functions .....	167
7.9.1	plugin_getdefaultconf .....	167
7.9.2	plugin_readmenu .....	167
7.9.3	plugin_parse .....	167
7.9.4	plugin_generatedefinition .....	168
7.9.5	plugin_saveconfig .....	168
7.9.6	plugin_generatelink .....	168
7.10	User Interface API Functions .....	169
7.10.1	toggleview .....	169
7.10.2	toggleidebar .....	169
7.10.3	setfilter .....	169
7.10.4	objectfilter .....	169
7.10.5	showshorttext .....	170
7.10.6	showtopbar .....	170
7.10.7	showtopmenubar .....	171
7.10.8	showmessage .....	171
7.10.9	showinfopage .....	171
7.10.10	showinfobox .....	172
7.10.11	showsharelinks .....	172
7.10.12	showmetadata .....	173
7.10.13	showobject .....	173
7.10.14	showmedia .....	173
7.10.15	showcompexplorer .....	174
7.10.16	showeditor .....	175
7.10.17	showinlineeditor_head .....	175
7.10.18	showinlinedatepicker_head .....	176
7.10.19	showinlineeditor .....	176
7.10.20	showvideoplayer .....	176
7.10.21	showvideoplayer_head .....	177
7.10.22	showaudioplayer .....	177
7.10.23	showaudioplayer_head .....	178
7.10.24	debug_getbacktracestring .....	178
7.10.25	showAPI docs .....	178
7.10.26	readnavigation .....	179
7.10.27	createnavigation .....	179
7.10.28	shownavigation .....	179
7.10.29	showselect .....	180
7.10.30	showtranslator .....	181
7.11	Template Engine API Functions .....	181
7.11.1	checklanguage .....	181
7.11.2	checkgroupaccess .....	181
7.11.3	transformlink .....	181
7.11.4	followlink .....	182
7.11.5	errorhandler .....	182
7.11.6	viewinclusions .....	182
7.11.7	buildview .....	183
7.11.8	buildsearchform .....	184
7.11.9	buildbarchart .....	184
7.12	XML API Functions .....	185
7.12.1	setxmlparameter .....	185
7.12.2	getcontent .....	185
7.12.3	geticontent .....	185
7.12.4	getxmlcontent .....	186
7.12.5	getxmlicontent .....	186
7.12.6	selectcontent .....	186

7.12.7	selectcontent .....	187
7.12.8	selectxmlcontent .....	187
7.12.9	selectxmlicontent .....	188
7.12.10	deletecontent .....	188
7.12.11	deleteicontent .....	188
7.12.12	setcontent .....	189
7.12.13	seticontent .....	189
7.12.14	setcontent_fast .....	190
7.12.15	updatecontent .....	190
7.12.16	insertcontent .....	191
7.12.17	inserticontent .....	191
7.12.18	addcontent .....	192
7.12.19	addicontent .....	192
7.13	Report API Functions .....	193
7.13.1	createreport .....	193
7.13.2	editreport .....	193
7.13.3	loadreport .....	194
7.13.4	deletereport .....	194
7.13.5	analyzeSQLselect .....	194
7.13.6	showpiechart .....	195
7.13.7	showcolumnchart .....	195
7.13.8	showtimelinechart .....	196
7.13.9	showgeolocationchart .....	196
7.14	Project API Functions .....	197
7.14.1	createproject .....	197
7.14.2	setproject .....	197
7.14.3	deleteproject .....	198
7.14.4	getprojectstructure .....	198
7.14.5	showprojectrecord .....	199
7.14.6	showganttchart .....	199
7.15	Task API Functions .....	200
7.15.1	createtask .....	200
7.15.2	settask .....	200
7.15.3	deletetask .....	201
7.15.4	tasknotification .....	201
7.16	Workflow API Functions .....	202
7.16.1	deleteworkflow .....	202
7.16.2	buildworkflow .....	202
7.16.3	workflowaccept .....	202
7.16.4	acceptobject .....	203
7.16.5	workflowreject .....	203
7.16.6	rejectobject .....	204
7.17	Cloud Storage API Functions .....	204
7.17.1	loadcloudobject .....	204
7.17.2	savecloudobject .....	204
7.17.3	copycloudobject .....	205
7.17.4	renamecloudobject .....	205
7.17.5	deletecloudobject .....	206
7.17.6	synccloudobjects .....	206
7.18	Import/Export API Functions .....	206
7.18.1	exportobjects .....	206
7.18.2	importobjects .....	207
8	Legal reference / flag .....	208
8.1	Questions and suggestions .....	208
8.2	Imprint .....	208
8.3	Legal information .....	208

# 1 Introduction

The following chapters deal with the function libraries of the hyper Content & Digital Asset Management Server and thus provide the documentation of the API (Application Programming Interface).

All libraries are located within the hyperCMS installation in the folder "function" and can be integrated and used in the respective scripts or templates. This can be used, for example, to create dynamic pages (applications) using the XML content repository.

If you run your application on a physically separated server, it is important that the function libraries are also available on the publication server. In this case you need to have access to the corresponding files on the publication server as well.

## 2 hyperCMS XML-Content-Repository

The XML content repository includes all XML Content Container and thus provides all content in native XML. The structure (schema) within an XML content container is dynamically generated based on the template used and has the following appearance:

```
<?xml version="1.0" encoding="UTF-8" ?>
<container>
  <hyperCMS>
    <contentcontainer>0000023.xml</contentcontainer>
    <contentxmlschema>object/page</contentxmlschema>
    <contentorigin>%page%/Publication/testpage.php</contentorigin>
    <contentobjects>%page%/Publication/testpage.php| %page%/ Publication/linkedcopy_of_testpage.php
  |</contentobjects>
    <contentuser>demouser</contentuser>
    <contentdate>2002-11-26</contentdate>
    <contentpublished>2002-11-26</contentpublished>
    <contentstatus>active</contentstatus>
  </hyperCMS>
  <head>
    <pagetitle>test</pagetitle>
    <pageauthor>Mr. Content</pageauthor>
    <pagedescription>just a small demonstration</pagedescription>
    <pagekeywords>demo of XML</pagekeywords>
    <pagecontenttype>text/html; charset=UTF-8</pagecontenttype>
    <pagelanguage>de</pagelanguage>
    <pagerevisit></pagerevisit>
  </head>
  <textcollection>
    <text>
      <text_id>headline</text_id>
      <textuser>demouser</textuser>
      <textcontent>fgfdgfdg</textcontent>
    </text>
    <text>
      <text_id>summary</text_id>
      <textuser>demouser</textuser>
      <textcontent><![CDATA[This is a
      <STRONG><EM>summary</EM></STRONG>]]></textcontent>
    </text>
  </textcollection>
  <mediacollection>
    <media>
      <media_id>logo</media_id>
      <mediauser>otheruser</mediauser>
      <mediafile>Publication/demo_hcms0000033.jpg</mediafile>
      <mediaobject>%page%/Publication/Multimedia/demo.jpg</mediaobject>
      <mediaalttext>demoimage</mediaalttext>
      <mediaalign></mediaalign>
      <mediawidth>200</mediawidth>
      <mediaheight>100</mediaheight>
    </media>
  </mediacollection>
```

```

<linkcollection>
  <link>
    <link_id>verweis</link_id>
    <linkuser>demouser</linkuser>
    <linkhref>http://localhost/index.php</linkhref>
    <linktarget>_blank</linktarget>
    <linktext>click me</linktext>
  </link>
</linkcollection>
<componentcollection>
  <component>
    <component_id>teasers</component_id>
    <componentuser>otheruser</componentuser>
    <componentcond>$customer == "private"</componentcond>
    <componentfiles>%comp%/Publication/teaser_1.php|%comp%/Publication/teaser_2.php|</componentfiles>
  </component>
  <component>
    <component_id>banner</component_id>
    <componentuser>demouser</componentuser>
    <componentcond></componentcond>
    <componentfiles>%comp%/banner.php</componentfiles>
  </component>
</componentcollection>
<articlecollection>
  <article>
    <article_id>news</article_id>
    <articletitle>Top News</articletitle>
    <articledatefrom>2002-10-01</articledatefrom>
    <articledateto>2002-11-01</articledateto>
    <articlestatus>active</articlestatus>
    <articleuser>demouser</articleuser>
    <articletextcollection>
      <text>
        <text_id>news:headline</text_id>
        <textuser>demouser</textuser>
        <textcontent>News from Scene</textcontent>
      </text>
    </articletextcollection>
    <articlemediacollection>
    </articlemediacollection>
    <articlelinkcollection>
    </articlelinkcollection>
    <articlecomponentcollection>
    </articlecomponentcollection>
  </article>
  <article>
    <article_id>special</article_id>
    <articletitle>Special Info</articletitle>
    <articledatefrom>2002-01-01</articledatefrom>
    <articledateto>2002-01-01</articledateto>
    <articlestatus>inactive</articlestatus>
    <articleuser>otheruser</articleuser>
    <articletextcollection>
      <text>
        <text_id>special:informations</text_id>
        <textuser>otheruser</textuser>
        <textcontent><![CDATA[<STRONG><FONT color=#cc0033>What is really going on behind the
Scene</FONT></STRONG>... find it out]]></textcontent>
      </text>
    </articletextcollection>
    <articlemediacollection>
    </articlemediacollection>
    <articlelinkcollection>
    </articlelinkcollection>
    <articlecomponentcollection>
    </articlecomponentcollection>
  </article>
</articlecollection>
</container>

```

After a review of the content container, a structure can be seen, which is composed of the following main elements for content storage:

- hyperCMS specific information
- Meta-information
- Text
- Media (images or other multimedia files)
- Links
- Components
- Articles

The entire content is made up of these basic elements whose information is stored within XML tags.

Articles include the elements text, media and links as well. The entire contents of a page or component can be obtained from the associated content containers.

## 2.1 hyperCMS specific information

The data collected in this XML node represent primarily relevant information for the management of the container.

```
<hyperCMS>
  <contentcontainer>0000023.xml</contentcontainer>
  <contentxmlschema>object/page</contentxmlschema>
  <contentorigin>%page%/testpage.php</contentorigin>
  <contentobjects>%page%/testpage.php| %page%/linkedcopy_of_testpage.php |</contentobjects>
  <contentuser>demouser</contentuser>
  <contentdate>2002-11-26</contentdate>
  <contentpublished>2002-11-26</contentpublished>
  <contentstatus>active</contentstatus>
</hyperCMS>
```

### Description:

contentcontainer	Name of the Content Container (unique for all publications)
contentxmlschema	Schema of the object: page (page) or component (comp)
contentorigin	Object (page or component) that led to the creation of the Content Containers
contentobjects	All objects which use the Container
contentuser	Object owner (user)
contentdate	Date of the last changes of the Containers
contentpublished	Date of the last publishing of the object based on the Content Container
contentstatus	Status can be "active" if an object using the Container exists. If all objects using the Container have been removed, the status will be set to "deleted". The Container therefore holds the last published information, but it can not be used anymore.

## 2.2 Meta-Information

The standard meta-information of a HTML page is described in this XML node.

```
<head>
  <pagetitle>test</pagetitle>
  <pageauthor>Mr. Content</pageauthor>
  <pagedescription>just a small demonstration</pagedescription>
  <pagekeywords>demo of XML</pagekeywords>
  <pagecontenttype>text/html; charset=UTF-8</pagecontenttype>
  <pagelanguage>de</pagelanguage>
  <pagerevisit></pagerevisit>
</head>
```

### Description:

pagetitle	Page title
pageauthor	Author
pagedescription	Description of the content of a page
pagekeywords	List of keywords of the page
pagecontenttype	Content-Type (character set) of the page or component
pagelanguage	Language shortcut of the page
pagerevisit	Search engine revisit of the page

## 2.3 Text

This XML-node stores the text.

```
<text>
  <text_id>headline</text_id>
  <textuser>demouser</textuser>
  <textcontent>fgfdgfdg</textcontent>
</text>
```

### Description:

text_id	Text identification
textuser	Text owner (last changes of the Text by a user)
textcontent	text content

## 2.4 Media

This XML-node describes an included media file.

```
<media>
  <media_id>logo</media_id>
  <mediauser>otheruser</mediauser>
  <mediafile>Publication/demo_hcms0000033.jpg</mediafile>
  <mediaobject>%page%/Publication/Multimedia/demo.jpg</mediaobject>
  <mediaalttext>demoimage</mediaalttext>
  <mediaalign></mediaalign>
  <mediawidth>200</mediawidth>
  <mediaheight>100</mediaheight>
</media>
```

### Description:

media_id	Media identification
mediauser	Media owner (last changes of the Media by a user)
mediafile	included multimedia file and declaration of the Publication
mediaobject	Location of the multimedia file
mediaalttext	Alternative text of the multimedia file
mediaalign	Alignment of the multimedia file
mediawidth	Displayed width of the multimedia file
mediaheight	Displayed height of the multimedia file

## 2.5 Links

This XML-node describes the link to a page or file.

```
<link>
  <link_id>link</link_id>
  <linkuser>demouser</linkuser>
  <linkhref>http://localhost/index.php</linkhref>
  <linktarget>_blank</linktarget>
  <linktext>click me</linktext>
</link>
```

### Description:

link_id	Link identification
linkuser	Link owner (last changes of the Link by a user)
linkhref	Reference (Link) to a page or file
linktarget	Target of the reference (name of the target frame)
linktext	Text describing the link

## 2.6 Components

This XML-node describes the reference to Components.

```
<component>
  <component_id>teasers</component_id>
  <componentuser>otheruser</componentuser>
  <componentcond>$customer == "private"</componentcond>
  <componentfiles>%comp%/teaser_1.php|%comp%/teaser_2.php|</componentfiles>
</component>
```

### Description:

component_id	Component identification
componentuser	Component owner (last changes of the Component reference by a user)
componentcond	assigned customer Profile to the Component
componentfiles	Reference (Component-link) to a single or multiple Components

## 2.7 Articles

This XML-node describes the article information.

```
<article>
  <article_id>news</article_id>
  <articletitle>Top News</articletitle>
  <articledatefrom>2002-10-01</articledatefrom>
  <articledateto>2002-11-01</articledateto>
  <articlestatus>active</articlestatus>
  <articleuser>demouser</articleuser>
  <articletextcollection>
  </articletextcollection>
</article>
```

### Description:

article_id	Article identification
articletitle	Title of the Article
articledatefrom	Publishing date of the Article
articledateto	End date of publishing the Article
articlestatus	Publishing settings of the Article: active = always published/displayed inactive = never published/displayed timeswitched = scheduled publishing/display
articleuser	Article owner (last changes of the Article by a user)
articlecollection	Holds all content of the Article



## 3 Function libraries

### 3.1 Including a library

The inclusion of a configuration or library requires that you know the absolute or relative path to the library. By using the function "require" or "require\_once" and specifying the path to the library file all functions contained in the library will be available. Once the library is included, all functions can be used in the script.

To use the hyperCMS functions, the file "hypercms\_api.inc.php" needs to be included. This file contains all functions required for programming.

```
// absolute path on MS Windows
require_once ("C:/inetpub/wwwroot/hypercms/function/hypercms_api.inc.php");

// relative path on MS Windows or Linux/UNIX
require_once ("function/hypercms_api.inc.php");
```

### 3.2 Loading the configuration

#### 3.2.1 Content Management Server

To use the main configuration of hyperCMS the appropriate configuration file must be loaded. The main configuration will be loaded when including the hyperCMS API. However you can also load it in your script.

Using the variable \$site for the identification of a publication, the publication can be loaded as well. The hyperCMS config file is located in "hypercms/config" and is named "config.inc.php". The publication config files are located in hyperCMS Data directory in the directory "data/config". Its filename holds the name of the publication as well as the ending "inc.php.", example: site.inc.php.

```
// Include the main config file (please set the correct path):
require_once ("C:/inetpub/wwwroot/hypercms/config.inc.php");

// Include publication management config file
// Attention: Please use valid_publicationname to verify the name before including the file
if (valid_publicationname ($site))
{
    require_once ($mgmt_config['abs_path_data']."config/".$site.".conf.php");
}
```

The config files can be opened and read. Each parameter is described therein and is available for use in programs. Therefore, please take a look at the configuration to learn more about the parameters and their names.

If you want to set a specific language, the variable \$lang need to be set. \$lang contains the language code, which is defined in the main configuration file "hypercms/config/config.inc.php".

```
// Set the language for messages in functions, German (de)
$lang = "de";
```

Since you want to use the hyperCMS API you need to include the hyperCMS API loader.

```
// Include the hyperCMS API:
require_once ($mgmt_config['abs_path_cms']."/function/hypercms_api.inc.php");
```

Now you can start using the API functions. For instance loading the content container of an object using various methods:

```
// Loading the page
$pagedata = loadfile ("%page%/MyPublication/home/", "index.php");

// Reading the name of the content container
$contentcontainer = filepointer ($pagedata, "content");

// Loading the live content container from the content repository
$containerdata = loadcontainer ($contentcontainer, "published", $user);

// Or even more simple by using the direct path to the object
$containerdata = getobjectcontainer ("MyPublication", "%page%/MyPublication/home/",
"index.php", $user);
```

The functions will also load the publication specific configuration in case it is not provided. Since many features require the settings of a publication, it is advisable to include the configuration before you plan any actions.

### 3.2.2 Publication Server

Note that the configuration of the publication server (publication target) is stored separately in an INI file. If you will need the publication target settings, you must load and parse the INI file. After that you can access the settings as an array.

The INI file of the publication target is located in the external repository in the directory "repository/config". The file name corresponds to the name of the publication with the file extension ".ini".

```
// Load and parse the INI file using PHP
$publ_config = parse_ini_file ("C:/inetpub/wwwroot/repository/config/Mandant_1.ini");

// Access the settings of the publication target
echo "This is the document root of the publication:". $publ_config[abs_publ_page];
```

## 3.3 Global variables

Many functions use global variables that are stored in the configuration and are available to functions as global. You should therefore take care that those global variable names of hyperCMS are not changed in your scripts.

The following list shows all global variables of hyperCMS, which must not be changed with in your own scripts:

```
$mgmt_config
$lang
$lang_name
$lang_shortcut
$lang_codepage
$lang_shortcut_default
```

Many global variables of hyperCMS are useful for use in hyperCMS scripts and PHP scripts, these are only available if the corresponding configuration has been loaded, or a hyperCMS script (used only during the publication process) is in use . Since this happened in the preview as well when publishing pages and components, these variables can be used in hyperCMS scripts. For dynamic applications that are executed each time a visitor accesses a page or component, the configuration must be integrated directly in the template, if hyperCMS variables are required.

#### Content Management Server:

<b>\$lang</b>	language shortcut according to config.inc.php
<b>\$mgmt_config['url_path_cms']</b>	URL of the hyperCMS root directory according to config.inc.php
<b>\$mgmt_config['abs_path_cms']</b>	absolute path to the hyperCMS root directory according to config.inc.php
<b>\$mgmt_config['url_path_page']</b>	URL of the document root of the publication in the management system
<b>\$mgmt_config['abs_path_page']</b>	absolute of the document root of the publication in the management system
<b>\$mgmt_config['url_path_comp']</b>	URL of the component root directory of the publication in the management system
<b>\$mgmt_config['abs_path_comp']</b>	absolute path of the component root directory of the publication in the management system

#### Publication Server:

hyperCMS scripts can access variables at any time. The values are stored in the array \$publ\_config, but are also optionally available without the array. If the script/application will be executed at each access of a page or component on the publication target, the configuration file must be loaded separately.

<b>\$publ_config['url_publ_page']</b>	URL of the document root of the publication target
<b>\$publ_config['abs_publ_page']</b>	absolute path of the document root of the publication target
<b>\$publ_config['url_publ_comp']</b>	URL of the document root of the publication target
<b>\$publ_config['abs_publ_comp']</b>	absolute path of the document root of the publication target

Optional (deprected):

<b>\$url_publ_page</b>	URL of the document root of the publication target
<b>\$abs_publ_page</b>	absolute path of the document root of the publication target
<b>\$url_publ_comp</b>	URL of the document root of the publication target
<b>\$abs_publ_comp</b>	absolute path of the document root of the publication target

## Vorlagenvariablen

There is also the possibility to use hyperCMS template variables in templates. These variables are a special feature, since they don not need to be used in hyperCMS script. Rather, they are placeholder for the value of a variable and can be used in any template.

This neutral form of the variables should primarily be used in templates, providing a more technology-neutral usage.

Please pay attention to the lower case of all variables!

**%container%** provides the name of the content containers of an object.

**%template%** provides the file name of the used template of the object.

**%object%** provides the name of the object.

**%date%** provides the actual date in the format YYYY-MM-DD.

For the integration of media files a path variable can be used. This path variable will be replaced by the URL (address) of publication target when publishing a page or component:

**%media%** provides the URL of the content media repository.

**%tplmedia%** provides the URL of the template media repository.

Also the document roots of pages and components of the publication target can be provided:

**%url\_page%** provides the root URL of the pages document root.

**%abs\_page%** provides the path to root directory of the pages document root.

**%url\_comp%** provides the root URL of the components document root.

**%abs\_comp%** provides the path to root directory of the components document root.

If you are using the hyperCMS APIs, it is often advisable to use the placeholders **%page%** and **%comp%** to access the document root of pages and components. This path variables can be used only on the management side.

It should be noted that the variable is always paired with the publication name to form the root directory, eg:

**%page%/besttrade/** .... Pages document root of the publication "besttrade"

**%page%/Publikationsname/** provides the path to root directory of the pages document root.

**%comp%/Publikationsname/** provides the path to root directory of the components document root.

## 3.4 Object operation library

This library contains all functions for the manipulation of objects (pages, components or files). You should only use these functions to access objects that are managed by the system.

### 3.4.1 createfolder

**Syntax:**

createfolder (\$site, \$location, \$foldernew, \$user)

**Description:**

Creates a new folder.

**Example:**

```
$result = createfolder ("besttrade", "%page%/besttrade/", "company", "brown");
```

**Input-Parameters:**

\$site	Name of the publication
\$location	absolute path (location of the new folder)
\$foldernew	Name of the new folder
\$user	User name

**globale Input-Parameters:**

The following global input parameters need to be passed to the function:

\$lang	Language setting or language shortcut, e.g. „en“, „de“
--------	--

**Output:**

Array	Array \$result holds the following information:
-------	---

\$result[result]	True/False (has the folder been created successfully)
\$result[add_onload]	JavaScript code for the onLoad event
\$result[message]	Message regarding the result of the action or error message
\$result[folder]	Name of the folder

### 3.4.2 deletefolder

**Syntax:**

deletefolder (\$site, \$location, \$folder, \$user)

**Description:**

Removes an existing folder. The folder is removed only if it contains no more objects. All objects must therefore be removed by using the function deleteobject.

**Example:**

```
$result = deletefolder ("besttrade", "%page%/besttrade/", "company", "brown");
```

**Input parameters:**

\$site	Name of the publication
\$location	absolute path (location of the folder)
\$folder	Name of the folder
\$user	User name

**global input parameters:**

The following global input parameters need to be passed to the function:

\$lang	Language setting or language shortcut, e.g. „en“, „de“
--------	--

**Output:**

Array	Array \$result holds the following information:
\$result[result]	True/False (has the folder been removed successfully)
\$result[add_onload]	JavaScript code for the onLoad event
\$result[message]	Message regarding the result of the action or error message
\$result[folder]	Name of the existing folder is not successful, otherwise empty

### 3.4.3 renamefolder

**Syntax:**

renamefolder (\$site, \$location, \$folder, \$foldernew, \$user)

**Description:**

Renames an existing folder.

Example:

```
$result = renamefolder ("besttrade", "%page%/besttrade/", "company", "news", "Miller");
```

**Input parameters:**

\$site	Name of the publication
\$location	absolute path (location of the folder)
\$folder	old folder name
\$foldernew	new folder name
\$user	User name

**global input parameters:**

The following global input parameters need to be passed to the function:

\$lang	Language setting or language shortcut, e.g. „en“, „de“
--------	--

**Output:**

Array	Array \$result holds the following information:
\$result[result]	True/False (Could the folder be renamed successfully)
\$result[add_onload]	JavaScript code for the onLoad event
\$result[message]	Message regarding the result of the action or error message
\$result[folder]	Name of the folder

### 3.4.4 createobject

**Syntax:**

createobject (\$site, \$location, \$object, \$template, \$user)

**Description:**

Creates a new page or component based on a template. Please note that the location (\$location) defines the category of the object (page/component) as well. This implies further that the value of the parameter \$template must provide a valid page or component template.

**Example:**

```
$result = createobject ("besttrade", "%page%/besttrade/", "index", "page_main", "Miller");
```

**Input parameters:**

\$site	Name of the publication
\$location	absolute path (location of the object)
\$object	Name of the new object (page or component)
\$template	Name of the page or component template (name of the template or template file name)
\$user	User name

**global input parameters:**

The following global input parameters need to be passed to the function:

\$lang	Language setting or language shortcut, e.g. „en“, „de“
--------	--

**Output:**

Array                      Array \$result holds the following information:

\$result[result]	True/False (result of the action)
\$result[add_onload]	JavaScript code for the onLoad event
\$result[message]	Message regarding the result of the action or error message
\$result[object]	File name of the page or component
\$result[objectname]	Name of the page or component
\$result[objecttype]	File-type or file extension of the file



### 3.4.5 deleteobject

**Syntax:**

deleteobject (\$site, \$location, \$object, \$user)

**Description:**

Removes an existing page, file or component.

Example:

```
$result = deleteobject ("besttrade", "%page%/besttrade/", "sales.doc", "Miller");
```

**Input parameters:**

\$site	Name of the publication
\$location	absolute path (location of the Object)
\$object	Name of the object
\$user	User name

**global input parameters:**

The following global input parameters need to be passed to the function:

\$lang	Language setting or language shortcut, e.g. „en“, „de“
--------	--

**Output:**

Array	Array \$result holds the following information:
\$result[result]	True/False (result of the action)
\$result[add_onload]	JavaScript code for the onLoad event
\$result[message]	Message regarding the result of the action or error message
\$result[publication]	Name of the publication where the object is located
\$result[location]	absolute path (location of the Object)
\$result[object]	File name of the page, file or component
\$result[objectname]	Name of the page, file or component
\$result[objecttype]	File-type or file extension of the file

### 3.4.6 renameobject

**Syntax:**

renameobject (\$site, \$location, \$object, \$objectnew, \$user)

**Description:**

Renames an existing page, file or component.

Example:

```
$result = renameobject ("besttrade", "%page%/besttrade/", "sales.doc", "best.doc",  
"Miller");
```

**Input parameters:**

\$site	Name of the publication
\$location	absolute path (location of the Object)
\$object	old name of the object
\$objectnew	new name of the object (without file extension)
\$user	User name

**global input parameters:**

The following global input parameters need to be passed to the function:

\$lang	Language setting or language shortcut, e.g. „en“, „de“
--------	--

**Output:**

Array	Array \$result holds the following information:
\$result[result]	True/False (result of the action)
\$result[add_onload]	JavaScript code for the onLoad event
\$result[message]	Message regarding the result of the action or error message
\$result[publication]	Name of the publication where the object is located
\$result[location]	absolute path (location of the Object)
\$result[object]	File name of the page, file or component
\$result[objectname]	Name of the page, file or component
\$result[objecttype]	File-type or file extension of the file

### 3.4.7 cutobject

**Syntax:**

cutobject (\$site, \$location, \$object, \$user)

**Description:**

Cut an existing page, file or component.

Example:

```
$result = cutobject ("besttrade", "%page%/besttrade/", "index.php", "Miller");
```

**Input parameters:**

\$site	Name of the publication
\$location	absolute path (location of the Object)
\$object	Name of the object
\$user	User name

**global input parameters:**

The following global input parameters need to be passed to the function:

\$lang	Language setting or language shortcut, e.g. „en“, „de“
--------	--

**Output:**

Array	Array \$result holds the following information:
\$result[result]	True/False (result of the action)
\$result[add_onload]	JavaScript code for the onLoad event
\$result[message]	Message regarding the result of the action or error message
\$result[object]	File Name of the page, file or component
\$result[objectname]	Name of the page, file or component
\$result[objecttype]	File-type or file extension of the file
\$result[clipboard]	temporary entry in the clipboard (can be passed as global variable \$clipboard to the function pasteobject, so reading the temporary file is not necessary)

### 3.4.8 copyobject

**Syntax:**

copyobject (\$site, \$location, \$object, \$user)

**Description:**

Copy an existing page, file or component.

Example:

```
$result = copyobject ("besttrade", "%page%/besttrade/", "index.php", "Miller");
```

**Input parameters:**

\$site	Name of the publication
\$location	absolute path (location of the Object)
\$object	Name of the object
\$user	User name

**global input parameters:**

The following global input parameters need to be passed to the function:

\$lang	Language setting or language shortcut, e.g. „en“, „de“
--------	--

**Output:**

Array	Array \$result holds the following information:
\$result[result]	True/False (result of the action)
\$result[add_onload]	JavaScript code for the onLoad event
\$result[message]	Message regarding the result of the action or error message
\$result[object]	File name of the page, file or component
\$result[objectname]	Name of the page, file or component
\$result[objecttype]	File-type or file extension of the file
\$result[clipboard]	temporary entry in the clipboard (can be passed as global variable \$clipboard to the function pasteobject, so reading the temporary file is not necessary)

### 3.4.9 copyconnectedobject

**Syntax:**

copyconnectedobject (\$site, \$location, \$object, \$user)

**Description:**

Connected copy an existing page, file or component sharing the same content container.

Example:

```
$result = copyconnectedobject ("besttrade", "%page%/besttrade/", "index.php", "Miller");
```

**Input parameters:**

\$site	Name of the publication
\$location	absolute path (location of the Object)
\$object	Name of the object
\$user	User name

**global input parameters:**

The following global input parameters need to be passed to the function:

\$lang	Language setting or language shortcut, e.g. „en“, „de“
--------	--

**Output:**

Array	Array \$result holds the following information:
-------	---

\$result[result]	True/False (result of the action)
\$result[add_onload]	JavaScript code for the onLoad event
\$result[message]	Message regarding the result of the action or error message
\$result[object]	File name of the page, file or component
\$result[objectname]	Name of the page, file or component
\$result[objecttype]	File-type or file extension of the file
\$result[clipboard]	temporary entry in the clipboard (can be passed as global variable \$clipboard to the function pasteobject, so reading the temporary file is not necessary)

### 3.4.10 pasteobject

**Syntax:**

pasteobject (\$site, \$location, \$user)

**Description:**

Paste Einfügen an existing page, file or component.

Example:

```
$result = pasteobject ("besttrade", "%page%/besttrade/", "Miller");
```

**Input parameters:**

\$site	Name of the publication
\$location	absolute path (location of the Object)
\$user	User name
\$clipboard	temporary entry in the clipboard (can be passed as global variable \$clipboard to the function pasteobject, so reading the temporary file is not necessary)

**global input parameters:**

The following global input parameters need to be passed to the function:

\$lang	Language setting or language shortcut, e.g. „en“, „de“
--------	--

**Output:**

Array                      Array \$result holds the following information:

\$result[result]	True/False (result of the action)
\$result[add_onload]	JavaScript code for the onLoad event
\$result[message]	Message regarding the result of the action or error message
\$result[publication]	Name of the publication where the object is located
\$result[location]	absolute path (location of the Object)
\$result[object]	File name of the page, file or component
\$result[objectname]	Name of the page, file or component
\$result[objecttype]	File-type or file extension of the file

### 3.4.11 lockobject

**Syntax:**

lockobject (\$site, \$location, \$object, \$user)

**Description:**

Locking of one or more existing pages or components based on the same content containers for the exclusive use of a user.

Example:

```
$result = lockobject ("besttrade", "%page%/besttrade/", "index.php", "Miller");
```

**Input parameters:**

\$site	Name of the publication
\$location	absolute path (location of the Object)
\$object	Name of the object
\$user	User name

**global input parameters:**

The following global input parameters need to be passed to the function:

\$lang	Language setting or language shortcut, e.g. „en“, „de“
--------	--

**Output:**

Array	Array \$result holds the following information:
-------	---

\$result[result]	True/False (result of the action)
\$result[add_onload]	JavaScript code for the onLoad event
\$result[message]	Message regarding the result of the action or error message
\$result[object]	File name of the page, file or component
\$result[objectname]	Name of the page, file or component
\$result[objecttype]	File-type or file extension of the file

### 3.4.12 unlockobject

**Syntax:**

unlockobject (\$site, \$location, \$object, \$user)

**Description:**

Unlocking of one or more existing pages or components based on the same content containers for the exclusive use of a user.

Example:

```
$result = unlockobject ("besttrade", "%page%/besttrade/", "index.php", "Miller");
```

**Input parameters:**

\$site	Name of the publication
\$location	absolute path (location of the Object)
\$object	Name of the object
\$user	User name

**global input parameters:**

The following global input parameters need to be passed to the function:

\$lang	Language setting or language shortcut, e.g. „en“, „de“
--------	--

**Output:**

Array	Array \$result holds the following information:
-------	---

\$result[result]	True/False (result of the action)
\$result[add_onload]	JavaScript code for the onLoad event
\$result[message]	Message regarding the result of the action or error message
\$result[object]	File name of the page, file or component
\$result[objectname]	Name of the page, file or component
\$result[objecttype]	File-type or file extension of the file



### 3.4.13 publishobject

**Syntax:**

publishobject (\$site, \$location, \$object, \$user)

**Description:**

Publishing a page or component. All connected copies of the object and its content container will be published as well. If a workflow is in use and does not permit the publishing, the object will not be published.

Example:

```
$result = publishobject ("besttrade", "%page%/besttrade/", "index.php", "Miller");
```

**Input parameters:**

\$site	Name of the publication
\$location	absolute path (location of the Object)
\$object	Name of the object
\$user	User name

**global input parameters:**

The following global input parameters need to be passed to the function:

\$lang	Language setting or language shortcut, e.g. „en“, „de“
--------	--

**Output:**

Array	Array \$result holds the following information:
\$result[result]	True/False (result of the action)
\$result[add_onload]	JavaScript code for the onLoad event
\$result[message]	Message regarding the result of the action or error message

### 3.4.14 unpublishobject

**Syntax:**

unpublishobject (\$site, \$location, \$object, \$user)

**Description:**

Unpublishing a page or component. Link and task management will be executed automatically. All connected copies of the object and its content containers will be unpublished as well.

**Example:**

```
$result = unpublishobject ("besttrade", "%page%/besttrade/", "index.php", "Miller");
```

**Input parameters:**

\$site	Name of the publication
\$location	absolute path (location of the Object)
\$object	Name of the object
\$user	User name

**global input parameters:**

The following global input parameters need to be passed to the function:

\$lang	Language setting or language shortcut, e.g. „en“, „de“
--------	--

**Output:**

Array	Array \$result holds the following information:
\$result[result]	True/False (result of the action)
\$result[add_onload]	JavaScript code for the onLoad event
\$result[message]	Message regarding the result of the action or error message

### 3.4.15 getlinkedobject

**Syntax:**

getlinkedobject (\$site, \$location, \$object, \$cat)

**Description:**

This function extracts all objects that have a reference to the given object. This may be page or component links. If the object is a page, then all objects which have a page link to the object will be determined. If the object is a component, all objects which have a component link to the object will be determined.

Example:

```
$result = getlinkedobject ("besttrade", "%page%/besttrade/", "index.php", "page");
```

**Input parameters:**

\$site	Name of the publication
\$location	absolute path (location of the Object)
\$object	Name of the object
\$cat	optional: Objekt category [page, comp]

**Output:**

Array	Array \$result holds the following information:
\$result	False (action was not successful)
\$result[publication]	Name of the publication where the object is located
\$result[location]	absolute path (location of the Object)
\$result[object]	Name of the object
\$result[category]	Object category [page, comp]

### 3.4.16      getconnectedobject

**Syntax:**

getconnectedobject (\$site, \$container)

**Description:**

This function determines all objects that are based on the same content container. The name of the content container of an object can be by extracted by the function "getfilename".

**Example:**

```
$result = getconnectedobject ("besttrade", "0000127.xml");
```

**Input parameters:**

\$site	Name of the publication
\$container	Name of the content container

**Output:**

Array	Array \$result holds the following information:
\$result	False (action was not successful)
\$result[publication]	Name of the publication where the object is located
\$result[location]	absolute path (location of the Object)
\$result[object]	Name of the object
\$result[category]	Object category [page, comp]

### 3.4.17      getObjectcontainer

**Syntax:**

getObjectcontainer (\$site, \$location, \$object, \$user)

**Description:**

This function loads the content containers (XML string) of a particular object. The object can be a page, file, component or folder.

The desired content can be extracted from the XML-based container using the function "getContent" or "selectcontent".

Example:

```
$xmldata = getObjectcontainer ("besttrade", "%page%/Home/", "index.php", "demouser");
```

**Input parameters:**

\$site	Name of the publication
\$location	absolute path (location of the Object)
\$object	Name of the object
\$user	User name

**Output:**

XML-String	Content of the content container
False	An error occurred

### 3.4.18      loadcontainer

**Syntax:**

loadcontainer (\$container)

**Description:**

This function loads the content container (XML string) by its name or by its ID (in this case, the current working container will be loaded by default).

The desired content can be extracted from the XML-based container using the function "getContent" or "selectcontent".

Example:

```
// Load the published container
$xmldata1 = loadcontainer "00012345.xml");
// Load the working container
$xmldata2 = loadcontainer ("00012345");
```

**Input parameters:**

\$container	Name or ID (including zeros) of the container
-------------	---

**Output:**

XML-String	Content of the content container
False	An error occurred

### 3.4.19      savecontainer

**Syntax:**

savecontainer (\$container, \$xmldata)

**Description:**

This function stores the content containers (XML string) by its name or by its ID (in this case, the current working container will be loaded by default).

Example:

```
// Save the published container
$result = savecontainer ("00012345.xml", $xmldata);
// Save the working container
$result = savecontainer ("00012345", $xmldata);
```

**Input parameters:**

\$container	Name or ID (including zeros) of the container
\$xmldata	Content of the content containers as XML string

**Output:**

True	The function was executed successfully
False	An error occurred

## 3.5 File Pointer library

This function library allows you to determine or set the XML content container of an. You must have this object (page or component) loaded previously, e.g. via http or via the file system using the function "loadfile".

### 3.5.1 getfilename

**Syntax:**

getfilename (\$filedata, \$tagname)

**Description:**

The input of the function is the content of a page, file object or component and the desired tag name, either content, template or media. After that the file name of the associated content container, template, or multimedia file will be returned.

Example:

```
// Load a page
```

```
$filedata = loadfile ("%page%/myPublication/home/", "index.php");
```

```
// Load the content container
```

```
$contentcontainer = filepointer ($filedata, "content");
```

**Input parameters:**

\$filedata      Content

\$tagname      Tag name [content, template, media]

**Output:**

Filename      The function was executed successfully and returns the file name of the content Container, template or multimedia file

False          An error occurred or the object is not managed by the system

### 3.5.2 setfilename

**Syntax:**

setfilename (\$filedata, \$tagname, \$value)

**Description:**

The input of the function is the content of a page, file object or component, the tag name and a value for the file parameter of the tag. The return value of the function will be True if writing the value has been successful or False otherwise.

Example:

```
$result = setfilename ($filedata, "template", "fullpage.page.tpl");
```

**Input parameters:**

\$filedata	Content of the page, file object or component
\$tagname	Tag name [content, template, media]
\$value	(File)name of the content container, multimedia file or template

**Output:**

True	The function was executed successfully
False	An error occurred



## 3.6 File operation library

The following functions for file operations should never be used to load or save objects (pages, components or files).

However you can use them to load and save XML content container, if you intend to develop extensions or applications.

### 3.6.1 loadfile

**Syntax:**

loadfile (\$abs\_path, \$filename)

**Description:**

This function loads the content of a file. The absolute path and the filename itself must be provided as input parameters. The function waits usually up to 3 seconds to load locked files. If the user parameter \$user is set, the function can also read locked files of the given user.

Example:

```
$data = loadfile ("%page%/myPublication/home/", "index.php");
```

**Input parameters:**

\$abs_path	absolute path of the file, %page% and %comp% can be used as the root elements of the path
\$filename	file name

**Output:**

File content	The function was executed successfully and returns the content of the file
False	An error occurred

### 3.6.2 savefile

**Syntax:**

savefile (\$abs\_path, \$filename, \$filedata)

**Description:**

This function saved content in files. The absolute path of the file name, and the content that will be written to the file needs to be passed as parameters. If the file is locked, it will not be saved and False will be returned.

Example:

```
$result = savefile ("%page%/myPublication/home/", "index.php", "text content");
```

**Input parameters:**

\$abs_path	absolute path of the file, %page% and %comp% can be used as the root elements of the path
\$filename	file name
\$filedata	Content that will be saved in the file

**Output:**

True	The function was executed successfully
False	An error occurred

### 3.6.3 loadlockfile

**Syntax:**

loadlockfile (\$user, \$abs\_path, \$filename)

**Description:**

This function allows to load the content of a file like the function "loadfile", but it is also triggers a locking mechanism for the file.

The function should only be used when the data will be saved again using the function "savelockfile". This ensures that no other write access by other users can take place.

The user, the absolute path and the filename itself must be passed as a parameter to load and lock the file. To save and unlock the file the function "savelockfile" must be used.

Example:

```
$data = loadlockfile ("Miller", "%page%/myPublication/home/", "index.php");
```

**Input parameters:**

\$user	User name of the user who locked the file
\$abs_path	absolute path of the file, %page% and %comp% can be used as the root elements of the path
\$filename	file name

**Output:**

File content	The function was executed successfully und liefert den Inhalt der Datei
False	An error occurred

### 3.6.4 savelockfile

**Syntax:**

savelockfile (\$user, \$abs\_path, \$filename, \$filedata)

**Description:**

The function "savefile" saved data and unlocks previously opened files using "loadlockfile".

For this purpose, the user, the absolute path, the file name, and the content that needs to be written to the file must be passed as parameters.

Example:

```
savelockfile ("Miller", "%page%/myPublication/home/", "index.php", "file content");
```

**Input parameters:**

\$user	User name of the user who locked the file
\$abs_path	absolute path of the file, %page% and %comp% can be used as the root elements of the path
\$filename	file name
\$filedata	Content that will be saved in the file

**Output:**

True	The function was executed successfully
False	An error occurred

### 3.6.5 lockfile

**Syntax:**

lockfile (\$user, \$abs\_path, \$filename)

**Description:**

The function "lockfile" locks a file for a specific user, so its available for the exclusive use. For this purpose, the user, the absolute path and the file name must be passed as a parameters.

Example:

```
lockfile ("Miller", "%page%/myPublication/home/", "index.php");
```

**Input parameters:**

\$user	User name of the user who locked the file
\$abs_path	absolute path of the file, %page% and %comp% can be used as the root elements of the path
\$filename	file name

**Output:**

True	The function was executed successfully
False	An error occurred

### 3.6.6 unlockfile

**Syntax:**

unlockfile (\$user, \$abs\_path, \$filename)

**Description:**

The function "unlockfile" unlocks files that have been previously locked by "lockfile" or opened by "loadlockfile". For this purpose, the user, the absolute path and the file name must be passed as a parameters.

Example:

```
unlockfile ("Miller", "%page%/myPublication/home/", "index.php");
```

**Input parameters:**

\$user	User name of the user who locked the file
\$abs_path	absolute path of the file, %page% and %comp% can be used as the root elements of the path
\$filename	file name

**Output:**

True	The function was executed successfully
False	An error occurred

### 3.6.7 deletefile

**Syntax:**

deletefile (\$location, \$file, \$recursive)

**Description:**

With "deletefile" files and (empty) folders can be deleted. The absolute path, the file or directory name, and a parameter "recursive", which is either (0) or (1), need to be passed. If recursive is set to 1 the entire contents of the directory will be processed, including subdirectories and their files, using the value 0 only the file or directory (if empty) will be removed.

Example:

```
deletefile ("%page%/myPublication/home/", "index.php", 0);
```

**Input parameters:**

\$abs_path	absolute path of the file, %page% and %comp% can be used as the root elements of the path
\$file	file name
\$recursive	0 or 1, if subdirectories should removed recursively as well

**Output:**

True	The function was executed successfully
False	An error occurred

### 3.6.8 appendfile

**Syntax:**

append (\$abs\_path, \$filename, \$filedata)

**Description:**

With "appendfile" content can be added to a file. The function does not overwrite existing data of a file, it appends the data at the file end. For this the absolute path, the file name, and the content that needs to be written to the file must be passed as parameters.

Example:

```
appendfile ("%page%/myPublication/home/", "index.php", "© 2003 ...");
```

**Input parameters:**

\$abs_path	absolute path of the file, %page% and %comp% can be used as the root elements of the path
\$filename	file name
\$filedata	Content that will be appended to the file

**Output:**

True	The function was executed successfully
False	An error occurred

## 3.7 Edit content library

The following functions allow you to read and write content from XML content container. You can optionally query the contents of the container with other technologies that can deal with XML. However, the Edit Content library offers a very simple and performant way of doing this.

### 3.7.1 setxmlparameter

**Syntax:**

setxmlparameter (\$xmldata, \$parameter, \$value)

**Description:**

Set a specific value of the XML declaration (1.row).

Example:

```
$xmldata = setxmlparameter ($xmldata, "encoding", "UTF-8");
```

**Input parameters:**

\$xmldata	XML string that should be manipulated
\$parameter	Name of the tag that should be manipulated
\$value	Value saved in the tag

**Output:**

XML-String	Return of the manipulated XML string
False	An error occurred

### 3.7.2 getcontent

**Syntax:**

getcontent (\$xmldata, \$tag)

**Description:**

Retrieves the XML content from the content container that is located inside the tags \$tag. An array containing all content or childs found will be returned.

Example:

```
// Get all text-childs from the content container  
$text_array = getcontent ($xmldata, "<text>");
```

```
// Show all text-childs  
foreach ($text_array as $text) echo $text;
```

**Input parameters:**

\$xmldata	XML string holding the content
\$tag	Name of the tag holding the information or child nodes

**Output:**

Array	Array holding all found values, the first value can be accessed using the first array element (Array[0])
False	An error occurred

### 3.7.3 getxmlcontent

**Syntax:**

getxmlcontent (\$xmldata, \$tag)

**Description:**

Retrieves the XML content from a content container that is located inside the tags \$tag and leaves in contrast to the function "getcontent" the XML tags in the return value (array). An entire node (well-formed) will therefore be returned.

An array containing all content and childs found will be returned and can be stored and used in a variable of type array.

Example:

```
$text_array = getxmlcontent ($xmldata, "<text>");  
foreach ($text_array as $text) echo $text;
```

**Input parameters:**

\$xmldata	XML string holding the content
\$tag	Name of the tag holding the information or child nodes

**Output:**

Array	Array holding all found values, the first value can be accessed using the first array element (Array[0])
False	An error occurred

### 3.7.4 selectcontent

**Syntax:**

selectcontent (\$xmldata, \$parenttag, \$childtag, \$childvalue)

**Description:**

Retrieves the XML content defined by the tag \$parenttag from the content container, where the childtag \$childtag has a certain value \$value.

An array with all items found will be returned and can be stored and used in a variable of type array.

Example:

Extract of a content container:

```
.....
<text>
  <text_id>summary</text_id>
  <textuser>editor1</textuser>
  <textcontent>This is my summary!</textcontent>
</text>
.....
```

```
// Get all text-children with id=summary
$text_array = selectcontent ($xmldata, "<text>", "<text_id>", "summary");

// Extract the summary from the found content
foreach ($text_array as $text)
{
  $summary = getcontent ($text, "<textcontent>");
}
```

**Input parameters:**

\$xmldata	XML string holding the content
\$parenttag	Name of the tag holding the information or child nodes
\$childtag	optional: XML tag that encloses the information that must be of a certain value
\$childvalue	optional: Value of the condition, the wildcard character * can be used at the beginning and/or end of the term and is a wildcard for any further characters.

**Output:**

Array	Array holding all found values, the first value can be accessed using the first array element (Array[0])
False	An error occurred



### 3.7.5 selectxmlcontent

**Syntax:**

selectxmlcontent (\$xmldata, \$parenttag, \$childtag, \$childvalue)

**Description:**

Retrieves the XML content defined by the tag \$parenttag from the content container, where the childtag \$childtag has a certain value \$value. In contrast to the function "getcontent" the parent tags are included in the return value (array).

An array with all items found will be returned and can be stored and used in a variable of type array.

Example:

Extract of a content container:

```
.....
<text>
  <text_id>summary</text_id>
  <textuser>editor1</textuser>
  <textcontent>This is my summary!</textcontent>
</text>
.....
```

```
// Get all text-children with id=summary
```

```
$text_array = selectxmlcontent ($xmldata, "<text>", "<text_id>", "summary");
```

```
// Extract the summary from the found content
```

```
foreach ($text_array as $text)
```

```
{
  $summary = getcontent ($text, "<textcontent>");
}
```

**Input parameters:**

\$xmldata	XML string holding the content
\$parenttag	Name of the tag holding the information or child nodes
\$childtag	optional: XML tag that encloses the information that must be of a certain value
\$childvalue	optional: Value of the condition, the wildcard character * can be used at the beginning and/or end of the term and is a wildcard for any further characters.

**Output:**

Array	Array holding all found values, the first value can be accessed using the first array element (Array[0])
False	An error occurred

### 3.7.6 deletecontent

**Syntax:**

deletecontent (\$xmldata, \$tagname, \$condtag, \$condvalue)

**Description:**

Removes the entire XML content defined by the tag \$tagname. For the selection of a certain child the appropriate XML childtag \$condtag and the enclosed information as \$condvalue as condition can be passed.

**Example:**

Extract of a content container:

```
.....  
<text>  
  <text_id>condition</text_id>  
  <textuser>editor1</textuser>  
  <textcontent>This is my summary!</textcontent>  
</text>  
.....
```

```
$xmldata = deletecontent ($xmldata, "<text>", "<text_id>", "bedingung");
```

**Input parameters:**

\$xmldata	XML string holding the content
\$parenttag	Name of the tag holding the information or child nodes that should be removed
\$condtag	optional: XML tag that encloses the information that must be of a certain value
\$condvalue	optional: Value of the condition

**Output:**

XML-String	Return of the manipulated XML string
False	An error occurred

### 3.7.7 setcontent

**Syntax:**

setcontent (\$xmldata, \$parenttagname, \$tagname, \$contentnew, \$condtag, \$condvalue)

**Description:**

An XML string is passed and within a certain parent node (\$parenttagname) a certain parameter (\$condtag) must exist and must have a certain value (\$condvalue). If the condition is satisfied, the value of the parameter \$tagname will be replaced by the new value \$contentnew.

**Example:**

Extract of a content container:

```
.....
<text>
  <text_id>condition</text_id>
  <textuser>editor1</textuser>
  <textcontent>This is should set!</textcontent>
</text>
.....
```

```
$xmldata = setcontent ($xmldata, "<text>", "<textcontent>", "This is my new value!",
"<text_id>", "condition");
```

**Input parameters:**

\$xmldata	XML string holding the content
\$parenttagname	optional: XML parent tag
\$tagname	optional: XML child tag, that value should be replaced (if the condition is met)
\$contentnew	new value for the XML child tag \$tagname
\$condtag	optional: XML tag that encloses the information that must be of a certain value
\$condvalue	optional: Value for the condition

**Output:**

XML-String	Return of the manipulated XML string
False	An error occurred

### 3.7.8 updatecontent

**Syntax:**

updatecontent (\$xmldata, \$xmlnode, \$xmlnodenew)

**Description:**

All XML strings \$xmlnode will be replaced by a new XML string \$xmlnodenew in \$xmldata. This method is faster than „setcontent“ when the updated XML node has already been extracted from the container.

**Example:**

Extract of a content container:

```
.....  
<text>  
  <text_id>condition</text_id>  
  <textuser>editor1</textuser>  
  <textcontent>This is old content!<textcontent>  
</text>  
.....
```

```
$xmldata = updatecontent ($xmldata, "<textcontent>This is old content!<textcontent> ",  
"<textcontent>This is my new content!<textcontent>");
```

**Input parameters:**

\$xmldata	XML string holding the content
\$xmlnode	XML string to be replaced (node or substring of \$xmldata)
\$xmlnodenew	optional: new XML string, if empty, the existing XML string will be removed.

**Output:**

XML-String	Return of the manipulated XML string
False	An error occurred

### 3.7.9 insertcontent

**Syntax:**

insertcontent (\$xmldata, \$insertxmldata, \$tagname)

**Description:**

Inserts an XML string (child node) before the end tag of the given XML parent tag. The modified XML string will be returned.

Example:

Extract of a content container:

```
.....
<articletextlist>
  <text>
    <text_id>art1:summary</text_id>
    <textuser>editor1</textuser>
    <textcontent>This is my summary!</textcontent>
  </text>
----- The new child node will be inserted here -----
  <text>
    <text_id>art1:longtext</text_id>
    <textuser>editor1</textuser>
    <textcontent>This is my summary!</textcontent>
  </text>
-----
</articletextlist>
.....
```

```
$xmldata = insertcontent ($xmldata, $insertxmldata, "<articletextlist>");
```

**Input parameters:**

\$xmldata	XML string holding the content
\$insertxmldata	XML string that will be inserted
\$tagname	optional: Include xml string before the end tag of the given XML parent tag

**Output:**

XML-String	Return of the manipulated XML string
False	An error occurred

### 3.7.10 addcontent

#### Syntax:

addcontent (\$xmldata, \$sub\_xmldata, \$grandtagname, \$condtag, \$condvalue, \$parenttagname, \$tagname, \$contentnew)

#### Description:

Within a parent node a child node will be added, provided that a certain value in the overlying grandparent node meets the condition. In the child node a value can be set as well. The modified XML string will be returned.

Example:

Extract of a content container:

```
.....
<article>
  <article_id>art1</article_id>
  <articletitle></articletitle>
  <articledatefrom></articledatefrom>
  <articledateto></articledateto>
  <articlestatus>active</articlestatus>
  <articleuser></articleuser>
  <articletextlist>
    <text>
      <text_id>art1:summary</text_id>
      <textuser>editor1</textuser>
      <textcontent>This is my summary!</textcontent>
    </text>
    ----- The new child node will be inserted here -----
    <text>
      <text_id>art1:longtext</text_id>
      <textuser>editor1</textuser>
      <textcontent>This is my summary!</textcontent>
    </text>
    -----
  </articletextlist>
</article>
.....
```

```
$xmldata = addcontent ($xmldata, $sub_xmldata, "<article>", "<article_id>", "art1",
"<articletextlist>", "<text_id>", "art1:longtext");
```

#### Input parameters:

\$xmldata	XML string holding the content
\$sub_xmldata	XML string that will be inserted
\$grandtagname	XML child tag where the \$sub_xmldata should be embedded
\$condtag	optional: XML tag that encloses the information that must be of a certain value
\$condvalue	optional: Value for the condition
\$parenttagname	optional: XML child tag, where \$sub_xmldata should be embedded
\$tagname	optional: Child tag of the embedded XML string where a value will be set
\$contentnew	optional: Value for the \$tagname

#### Output:

XML-String	Return of the manipulated XML string
False	An error occurred

## 3.8 Meta Data Generator library

This function library allows you to automatically create keyword lists and a description from a given content. This can be used to automatically generate and fill in metadata for pages. Also meta data from multimedia files can be extracted and stored in the container of an object.

### 3.8.1 getkeywords

**Syntax:**

getkeywords (\$text, \$language, \$charset)

**Description:**

The function requires the text content to be passed as input. All keywords are determined from the text and returned as a keyword list.

Example:

```
$keywords = getkeywords ("This is just a short text.", "en", "UTF-8");
```

**Input parameters:**

\$text	Content als String
\$language	optional: Language [en, de], default is "en"
\$charset	optional: Character set, default is "UTF-8"

**Output:**

Keywords	Comma seperated list of keywords
False	An error occured

### 3.8.2 getdescription

**Syntax:**

getdescription (\$text, \$charset)

**Description:**

The function requires the text content to be passed as input. A brief description from the given text will be extracted and returned.

Example:

```
$keywords = getdescription ("This is just a short text.", "UTF-8");
```

**Input parameters:**

\$text	Content as string
\$charset	optional: Character set, default is "UTF-8"

**Output:**

Keywords	Short description of the content
False	An error occured

### 3.8.3 injectmetadata

**Syntax:**

injectmetadata (\$site, \$location, \$object, \$mediafile, \$mapping, \$user)

**Description:**

The function requires the path to the object and, optionally, the filename of the object or the multimedia file, and a mapping in order to save the meta data. The content from the meta data will be saved in the in the appropriate text ID of the container of the object, based on the given mapping.

WARNING: Existing data of the container will be overwritten!

Example:

```
// Mapping Definition (Meta Data Name -> Text-ID)
```

```
// Dublin Core
```

```
$mapping['dc:title'] = "Title";
```

```
$mapping['dc:subject'] = "Keywords";
```

```
$mapping['dc:description'] = "Description";
```

```
$mapping['dc:creator'] = "Creator";
```

```
$mapping['dc:rights'] = "Copyright";
```

```
// Adobe PhotoShop
```

```
$mapping['photoshop:SupplementalCategories'] = "Categories";
```

```
// Image Resolution defines Quality [Print, Web]
```

```
$mapping['hcms:quality'] = "Quality";
```

```
$result = injectmetadata ("Publication", "%comp%/test/", "image.jpg", "", $mapping,  
"Miller");
```

**Input parameters:**

\$site            Name of the publication

\$location       Absolute path to the object (Location of the object)

\$object         optional: Name of the object (multimedia object file)

\$mediafile      or optional: Name of the multimedia file

\$mapping        Mapping array (Meta Data Name -> Text-ID)

\$user           User name

**Output:**

True            Meta data has been saved successfully

False           An error occurred



## 3.9 Notifications library

This function library sends automated messages to a user based on limits of a certain value of a particular field.

The user receives a pre-formatted message with information (links) to all objects that are within the search area (date upper and lower limit).

### 3.9.1 licensenotification

#### **Syntax:**

licensenotification (\$site, \$cat, \$folderpath, \$text\_id, \$date\_begin, \$date\_end, \$user)

#### **Description:**

The function returns all objects due to the specified search range (location and date limits) and sends an e-mail to a specific user with links to all the affected objects.

Example:

```
// set language for mail message
```

```
$lang = "en";
```

```
// send mail to Miller
```

```
$result = licensenotification ("Demo-DAM", "%comp%/images/", "comp", "valid_date",  
"2012-09-01", "2012-09-30", "Miller");
```

#### **Input parameters:**

\$site	Name of the publication
\$cat	Object category [page, comp]
\$folderpath	Location for the definition of the search area
\$text_id	Text ID of the XML node that need to be analyzed
\$date_begin	Start date for the seach (YYYY-MM-DD)
\$date_end	End date for the seach (YYYY-MM-DD)
\$user	User name

#### **Output:**

True	Mail wurdwas send successfully
False	An error occured

## 4 Components and applications

If applications are integrated into components and variables need to be passed from a page to a component, you need to pay attention to the following:

The components must be integrated via the file system (not via HTTP).

All variables to be passed to the component need to be defined in the component as global.

Example:

A page passes a variable to a component.

Code example of a page:

```
<html>
<head>
<title>page</title>
</head>
<body>
<?php $test="This is just a test!"; ?>
[hyperCMS:components id='component']
</body>
</html>
```

The code of the component must be as followed:

```
<p>
<?php
global $test;
echo $test;
?>
</p>
```

In the example, the variable `$test` and its value "This is just a test!" will be passed to the component and will be displayed in the presentation of the component.

## 5 Database Connectivity

The Database Connectivity of the hyper Content Management Server allows the connection of different databases for the storage and retrieval of content. Relational databases are widely used as an external content repository.

For this purpose, a corresponding hyperCMS tag for the Database Connectivity need to be present in each template, which then refers to a DB-Connect file.

In this file functions are stored that hyperCMS will call, provided that the template points to the function file.

The contents are read from the database and will be displayed to the editor. If the editor modifies the content, it will be written to the database again. For read and write access different databases can be accessed as well. The functions in the DB Connect File offer only the shell or standardized interface to hyperCMS that needs to be filled by the programmer.

The subject of database integration is complex and need to be treated individual, since existing databases and their information must be integrated. hyperCMS does not provide any ER model or commits itself to specific database vendors. In general it can be said that all the possibilities of PHP can be exploited in order to connect to various data sources.

Besides the necessary parameters for queries to relational databases, the entire content container is passed as an XML string. This allows documents or content from the content repository to be stored as a node in XML databases as well.

You therefore decide from which sources you read and save data. With PHP you have a powerful language that gives you access to all major databases.

More information regarding the functions of PHP can be found here: <http://www.php.net>

### 5.1 Creating a Database Connectivity

If you want to create a Database Connectivity, you need to make a copy of the the file "db\_connect\_default.php". This file can be found in the root directory for storing the management data under the following path: /data /db\_connect/

The copy of the file should be named according to the database you want to connect with.

Open the file and gain insight into the functions. In the source code you will also find a description of the functions and parameters passed as well as the output parameters.

The following example will show a read access to a MySQL database in order to extract a text based content. We assume that in a table "TextContent" the content will be presented by the primary key "container\_id" and "text\_id" as well as the text content "text" itself and the text-type "type". The user and the article ID is not stored separately, this is also not necessary for the uniqueness of the content, because the ID of the content container as well as the ID of the element already provides the primary key.

```

// ===== db connect =====
// this file allows you to access a database using the full PHP functionality.
// you can read or write data from or into a database:

// ===== read from database =====
// the following parameter values are passed to each function for
// retrieving data from the database:
// name of the site: $site [string]
// name of the content container: $container_id [string] (is unique
// inside hyperCMS over all sites)
// content container: $container_content [XML-string]
// identification name: $id [string]

// ----- text -----
// if content is text
function db_read_text ($site, $content_id, $container_content, $id, $art_id, $user)
{
    //-----
    // input variables: $id [string], optional: $artid [string], $user [string]
    // return value: $text [array]
    //         the array must exactly look like this:
    //         $text[text], optional: $text[type]
    //         constraints/accepted values for article type, see note below
    // note: special characters in $text are escaped into
    //       their html/xml equivalents.
    //       you can decide between unformatted, formatted and
    //       optional text using $type:
    //       unformatted text: $text[type] = textu
    //       formatted text: $text[type] = textf
    //       text option: $text[type] = textl
    //-----

    $user = "username";
    $password = "password";
    $database = "database";

    // connect to database
    mysql_connect ("localhost", $user, $password);
    @mysql_select_db ($database) or die ("Unable to select database");

    // fire SQL-query
    $result = mysql_query ("SELECT Text, Type FROM TextContent WHERE
        container_id=$container_id AND text_id=$id");

    // count returned rows, must be 1 if unique
    $num_of_rows = mysql_num_rows ($result);

    // get the result into an array named $row
    if ($num_of_rows == 1)
    {
        $row = mysql_fetch_row ($result);

        // set values
        $text[text] = $row[0];
        $text[type] = $row[1];
    }
    else $text = false;

    // close connection
    mysql_close ();

    // return result
    return $text;
}

```

## 6 Event System

The hyper Content & Digital Asset Management Server provides an event system that allows automated execution of actions when events in the system will be executed. This can be used, for example, to automate manual processes.

Events are usually started by the user by selecting an action, e.g. publishing a page. If the corresponding event is enabled, the event "onpublishobject" will be called after successful execution of the publication process of the page. The functions defined therein will therefore be executed.

The events of the event system can be defined in the "hypercms\_eventsys.inc.php" file. This file is located in the internal repository in the folder "data/event system". In this file there are also other important instructions that must be followed during the execution of events.

The event system is valid within the management system for all publications. The system is part of the hyperCMS API and is thus performed on each invocation of the API functions.

Events can be activated and deactivated, so that the use of its defined events can be easily controlled in the "hypercms\_eventsys.inc.php" file.

There is a distinction between PRE and POST events. The PRE event will be fired before the actual execution of the called action, while the POST event is called after the successful execution of the action.

Example:

When publishing an object the page "index.php" located at the same position should be automatically published as well, since the page "index.php" is used to generate an overview using a hyperCMS script all objects of the same folder.

```
// ----- on publish object POST event -----  
function onpublishobject_post ($site, $cat, $location, $object, $user)  
{  
    // load configuration  
    include_once ("config.inc.php");  
  
    // hide the event used in your action (1) otherwise execute event (0)  
    $eventsystem[hide] = 1;  
  
    // insert your program code here  
    $result = publishobject ($site, $location, "index.php", $user);  
  
    // return true if successful  
    if ($result[result] == true) return true;  
    else return false;  
}
```

## 7 hyperCMS API Function Reference

The documentation of all API functions in the current version is also available on our website [hypercms.com](http://hypercms.com). You can view the documentation in its current version directly in the browser. Click on the ?-Icon in the template editor to access the help with all hyperCMS tags and API functions.

### 7.1 Main API Functions

#### 7.1.1 correctnumber

**Syntax:**

correctnumber (\$number)

**Input parameters:**

\$number ... formatted number

**global input parameters:**

\$mgmt\_config

**Output:**

correct mathematical number / false on error

#### 7.1.2 cleancontent

**Syntax:**

cleancontent (\$text, \$charset="UTF-8")

**Input parameters:**

\$text ... expression (mixed)

\$charset ... character set (optional)

**global input parameters:**

\$mgmt\_config

**Output:**

converted expression / false on error

#### 7.1.3 convertchars

**Syntax:**

convertchars (\$expression, \$charset\_from="UTF-8", \$charset\_to="UTF-8")

**Input parameters:**

\$expression ... expression as string or array

\$charset\_from ... input character set (optional)

\$charset\_to ... output character set (optional)

**global input parameters:**

\$mgmt\_config

**Output:**

converted expression / false on error

### 7.1.4 specialchr

**Syntax:**

specialchr (\$expression, \$accept="")

**Input parameters:**

\$expression ... expression

\$accept ... list of characters to be excluded from search (optional)

**Output:**

true/false

**Description:**

Tests if an expression includes special characters (true) or does not (false).

Allows you to accept characters through including it into \$accept (e.g. #\$.)

### 7.1.5 specialchr\_encode

**Syntax:**

specialchr\_encode (\$expression, \$remove="no")

**Input parameters:**

\$expression ... expression

\$remove ... remove all special characters [yes, no]

**global input parameters:**

\$user

\$mgmt\_config

\$hcms\_lang

\$lang

**Output:**

expression without special characters (for file names)

**Description:**

Renames all special characters for file names to an expression according to given rules

### 7.1.6 specialchr\_decode

**Syntax:**

specialchr\_decode (\$expression)

**Input parameters:**

\$expression ... expression

**global input parameters:**

\$user

\$mgmt\_config

\$hcms\_lang

\$lang

**Output:**

expression with special characters (for file names) / false

**Description:**

This is the decode function for function specialchr\_encode

### 7.1.7 is\_utf8

**Syntax:**

is\_utf8 (\$str)

**Input parameters:**

\$str ... expression

**Output:**

if string is utf-8 encoded true / false otherwise

**Description:**

This function is an alternative to mb\_check\_encoding (which requires an extra PHP module). It is not failsave!

### 7.1.8 is\_latin1

**Syntax:**

is\_latin1 (\$str)

**Input parameters:**

\$str ... expression

**Output:**

if string is latin 1 encoded true / false otherwise

**Description:**

This function is an alternative to mb\_check\_encoding (which requires an extra PHP module). It is not failsave!

### 7.1.9 makestring

**Syntax:**

makestring (\$array)

**Input parameters:**

\$array ... (multidimensional) array

**Output:**

string including all array values / false on error

### 7.1.10 splitstring

**Syntax:**

splitstring (\$string)

**Input parameters:**

\$string ... string with ";" or ", " as separator

**Output:**

array with string splitted into array / false on error

### 7.1.11 is\_emptyfolder

**Syntax:**

is\_emptyfolder (\$dir)



**Input parameters:**

\$dir ... path to folder

**global input parameters:**

\$mgmt\_config

**Output:**

true / false

**Description:**

Checks if a directory/folder is empty (has no published objects or other files)

### 7.1.12 is\_supported

**Syntax:**

is\_supported (\$preview\_array, \$file)

**Input parameters:**

\$preview\_array ... preview array holding the supported file extensions as key and references to executables as value

\$file ... file name or file extension

**Output:**

true / false

**Description:**

This function determines if a certain file type by its file extension is supported by the systems media conversion

### 7.1.13 is\_cloudstorage

**Syntax:**

is\_cloudstorage (\$site="")

**Input parameters:**

\$site ... publication name (optional)

**global input parameters:**

\$mgmt\_config

**Output:**

true / false

**Description:**

This function determines if a cloud storage has been defined in the main configuration or for a specific publication

### 7.1.14 is\_cloudobject

**Syntax:**

is\_cloudobject (\$file)

**Input parameters:**

\$file ... path to media file or media file name

**global input parameters:**

\$mgmt\_config

**Output:**

true / false

**Description:**

This functions verifies if an object/file is available in the cloud storage

### 7.1.15 is\_date

**Syntax:**

is\_date (\$date, \$format="Y-m-d")

**Input parameters:**

\$date ... date as string

\$format ... date format (optional)

**Output:**

true / false

**Description:**

This function determines if a string represents a valid date format

### 7.1.16 is\_tempfile

**Syntax:**

is\_tempfile (\$path)

**Input parameters:**

\$path ... file name or path [string]

**global input parameters:**

\$user

\$mgmt\_config

\$hcms\_lang

\$lang

**Output:**

if file is a temp file true / false on error

**Description:**

This functions checks if the provided file name is a temporary file

### 7.1.17 is\_thumbnail

**Syntax:**

is\_thumbnail (\$media, \$images\_only=true)

**Input parameters:**

\$media ... file name or path [string]

\$images\_only ... only thumbnail images should be considered as thumbnail [true, false]

**global input parameters:**

\$user

\$mgmt\_config

\$hcms\_lang

\$lang

**Output:**

if file is a thumbnail file true / false on error

**Description:**

This functions checks if the provided file name is a thumbnail file

### 7.1.18 is\_config

**Syntax:**

is\_config (\$media)

**Input parameters:**

\$media ... file name or path [string]

**global input parameters:**

\$user

\$mgmt\_config

\$hcms\_lang

\$lang

**Output:**

if file is a config file true / false if not

**Description:**

This functions checks if the provided file name is a config file

### 7.1.19 is\_encryptedfile

**Syntax:**

is\_encryptedfile (\$location, \$file)

**Input parameters:**

\$location ... path to file [string]

\$file ... file name [string]

**global input parameters:**

\$user

\$mgmt\_config

\$hcms\_lang

\$lang

**Output:**

true / false

**Description:**

This functions checks if the provided file is encrypted

### 7.1.20 is\_document

**Syntax:**

is\_document (\$file)

**Input parameters:**

\$file ... file name or file extension

**global input parameters:**

\$mgmt\_config

\$hcms\_ext

**Output:**

true / false

**Description:**

This function determines if a certain file is a document (binary and text based)

### 7.1.21 is\_image

**Syntax:**

is\_image (\$file)

**Input parameters:**

\$file ... file name or file extension

**global input parameters:**

\$mgmt\_config

\$hcms\_ext

**Output:**

true / false

**Description:**

This function determines if a certain file is an image

### 7.1.22 is\_rawimage

**Syntax:**

is\_rawimage (\$file)

**Input parameters:**

\$file ... file name or file extension

**global input parameters:**

\$mgmt\_config

\$hcms\_ext

**Output:**

true / false

**Description:**

This function determines if a certain file is a raw image

### 7.1.23 is\_aiimage

**Syntax:**

is\_aiimage (\$file)

**Input parameters:**

\$file ... file name or file extension

**Output:**

true / false

**Description:**

This function determines if a certain file is a vector-based Adobe Illustrator (AI) or AI-compatible EPS file

### 7.1.24 is\_video

**Syntax:**

is\_video (\$file)

**Input parameters:**

\$file ... file name or file extension

**global input parameters:**

\$mgmt\_config

\$hcms\_ext

**Output:**

true / false

**Description:**

This function determines if a certain file is a video

### 7.1.25 is\_audio

**Syntax:**

is\_audio (\$file)

**Input parameters:**

\$file ... file name or file extension

**global input parameters:**

\$mgmt\_config

\$hcms\_ext

**Output:**

true / false

**Description:**

This function determines if a certain file is an audio file

### 7.1.26 is\_mobilebrowser

**Syntax:**

is\_mobilebrowser ()

**Input parameters:****global input parameters:**

\$user

\$mgmt\_config

\$hcms\_lang

\$lang

**Output:**

true / false

**Description:**

Detects if a mobile browser is used

### 7.1.27 copyrecursive

**Syntax:**

copyrecursive (\$src, \$dst)

**Input parameters:**

\$src ... source directory

\$dst ... destination directory

**Output:**

true/false

**Description:**

This function copies all directories and files from source to destination directory

### 7.1.28 correctfile

**Syntax:**

correctfile (\$abs\_path, \$filename, \$user="")

**Input parameters:**

\$abs\_path ... path to file or directory

\$filename ... file or directory name

\$user ... user name

**global input parameters:**

\$mgmt\_config

\$hcms\_lang

\$lang

**Output:**

correct filename/false

### 7.1.29 correctpath

**Syntax:**

correctpath (\$path, \$slash="/")

**Input parameters:**

\$path ... path to folder

\$slash ... directory separator (optional)

**Output:**

correct path/false

### 7.1.30 convertpath

**Syntax:**

convertpath (\$site, \$path, \$cat)

**Input parameters:**

\$site ... publication

\$path ... content management path to folder or object

\$cat ... object category ['page

**global input parameters:**

\$user

\$mgmt\_config  
\$hcms\_lang  
\$lang

**Output:**

converted path or URL / false on error

**Description:**

This function replaces object pathes of the content management config with %page% and %comp% path variables

### 7.1.31 convertlink

**Syntax:**

convertlink (\$site, \$path, \$cat)

**Input parameters:**

\$site ... publication  
\$path ... publication management path to folder or object  
\$cat ... object category ['page

**global input parameters:**

\$user  
\$mgmt\_config  
\$publ\_config  
\$hcms\_lang  
\$lang

**Output:**

converted path or URL / false on error

**Description:**

This function replaces pathes of the publication management config with %page% and %comp% path variables.

### 7.1.32 deconvertpath

**Syntax:**

deconvertpath (\$path, \$type="file", \$specialchars\_transform=false)

**Input parameters:**

\$path ... string including path to folder or object  
\$type ... convert to file system path or URL [file  
\$specialchars\_transform ... url] (optional)

**global input parameters:**

\$user  
\$mgmt\_config  
\$hcms\_lang  
\$lang

**Output:**

deconverted path/false

**Description:**

This function replaces all %page% and %comp% path variables with the path of the content management config.

It converts the path only on content management side not for the publication target.  
It optionally transform special characters as well.

### 7.1.33      deconvertlink

**Syntax:**

deconvertlink (\$path, \$type="url")

**Input parameters:**

\$path ... path to folder or object

\$type ... convert to file system path or URL [file

**global input parameters:**

\$user

\$mgmt\_config

\$publ\_config

\$hcms\_lang

\$lang

**Output:**

converted absolute link without host/false

**Description:**

This function deconverts the path only for the publication target.

It should be used for page linking, otherwise the function will return the absolute link including the host for component links.

### 7.1.34      createviewlink

**Syntax:**

createviewlink (\$site, \$mediafile, \$name="", \$force\_reload=false, \$type="wrapper")

**Input parameters:**

\$site ... publication

\$mediafile ... media file name

\$name ... media name (optional)

\$force\_reload ... force reload [true, false] (optional)

\$type ... link type [wrapper, download] (optional)

**global input parameters:**

\$user

\$mgmt\_config

**Output:**

URL for download of the multimedia file / false on error

**Description:**

The view link is mainly used inside the system in order to reference and load a multimedia file.

The database is not required since the object hash or ID is not needed to create the view link.

### 7.1.35      createaccesslink

**Syntax:**

createaccesslink (\$site, \$location="", \$object="", \$cat="", \$object\_id="", \$login, \$type="al", \$lifetime=0, \$formats="")

**Input parameters:**



\$site ... publication  
\$location ... location (optional)  
\$object ... object (optional)  
\$cat ... category [page, comp] (optional)  
\$object\_id ... object-ID (optional)  
\$login ... user login  
\$type ... link type [al, dl] (optional)  
\$lifetime ... token lifetime in seconds (optional)  
\$formats ... formats (optional)

**global input parameters:**

\$user  
\$mgmt\_config  
\$hcms\_lang  
\$lang

**Output:**

URL for access to given object / false on error

### 7.1.36 createwrapperlink

**Syntax:**

createwrapperlink (\$site="", \$location="", \$object="", \$cat="", \$object\_id="",  
\$container\_id="")

**Input parameters:**

\$site ... publication (optional)  
\$location ... location (optional)  
\$object ... object (optional)  
\$cat ... category [page, comp] (optional)  
\$object\_id ... object ID (optional)  
\$container\_id ... container-ID (optional)

**global input parameters:**

\$user  
\$mgmt\_config  
\$hcms\_lang  
\$lang

**Output:**

URL for download of the multimedia file of the given object or folder / false on error

### 7.1.37 createdownloadlink

**Syntax:**

createdownloadlink (\$site="", \$location="", \$object="", \$cat="", \$object\_id="",  
\$container\_id="")

**Input parameters:**

\$site ... publication name (optional)  
\$location ... location (optional)  
\$object ... object (optional)  
\$cat ... category [page, comp] (optional)  
\$object\_id ... object ID (optional)  
\$container\_id ... container-ID (optional)

**global input parameters:**

\$user

\$mgmt\_config  
\$hcms\_lang  
\$lang

**Output:**

URL for download of the multimedia file of the given object or folder / false on error

### 7.1.38 createmultidownloadlink

**Syntax:**

createmultidownloadlink (\$site, \$multiobject="", \$media="", \$location="", \$name="", \$user, \$type="", \$mediacfg="", \$linktype="download")

**Input parameters:**

\$site ... publication name  
\$multiobject ... multiobject string (optional)  
\$media ... media file name (optional)  
\$location ... location (optional)  
\$name ... presentation name (optional)  
\$user ... user name  
\$type ... conversion type (format  
\$mediacfg ... e.g: jpg)  
\$linktype ... media configuration used for conversion (e.g.: 1024x768px)

**global input parameters:**

\$mgmt\_config  
\$mgmt\_compress  
\$pageaccess  
\$compaccess  
\$hiddenfolder  
\$hcms\_linking  
\$globalpermission  
\$setlocalpermission  
\$hcms\_lang  
\$lang

**Output:**

download link / false on error

**Description:**

Generates a download link of a single media file, folder or multi objects.  
Priority if multiple input parameters for media file, folder or multi objects are given:  
1st...multi objects  
2nd...media file  
3rd...folder

### 7.1.39 cleandomain

**Syntax:**

cleandomain (\$path)

**Input parameters:**

\$path ... string to clean from http(s)://domain

**global input parameters:**

\$mgmt\_config  
\$hcms\_lang  
\$lang

**Output:**

cleanded string / false on error

**Description:**

Returns the URL notation without the protocoll://domain.

#### 7.1.40 fileversion

**Syntax:**

fileversion (\$file)

**Input parameters:**

\$file ... file name [string]

**Output:**

versioned file name [string] / false on error

**Description:**

Creates a version file name

#### 7.1.41 createversion

**Syntax:**

createversion (\$site, \$file, \$user="sys")

**Input parameters:**

\$site ... publication name

\$file ... media file name or container name

\$user ... user name (optional)

**global input parameters:**

\$mgmt\_config

\$mgmt\_mediaoptions

\$mgmt\_docoptions

\$hcms\_ext

\$user

**Output:**

true / false

**Description:**

Creates a new version of a multimedia file and container

#### 7.1.42 rollbackversion

**Syntax:**

rollbackversion (\$site, \$location, \$page, \$container\_version, \$user="sys")

**Input parameters:**

\$site ... publication name

\$location ... location

\$page ... object name

\$container\_version ... container version name

\$user ... user name (optional)

**global input parameters:**

\$mgmt\_config

\$mgmt\_mediaoptions  
\$mgmt\_dcoptions  
\$hcms\_ext  
\$hcms\_lang  
\$lang

**Output:**

result\_array

**Description:**

Makes an older object version to the current version

### 7.1.43 deleteversion

**Syntax:**

deleteversion (\$site, \$container\_version, \$user="sys")

**Input parameters:**

\$site ... publication name  
\$container\_version ... location  
\$user ... object name

**global input parameters:**

\$mgmt\_config

**Output:**

true / false

**Description:**

Removes the version of an object

### 7.1.44 deleteversions

**Syntax:**

deleteversions (\$type, \$report, \$user="sys")

**Input parameters:**

\$type ... type [content, template] or valid path in filesystem  
\$report ... report [yes, no]  
\$user ... user name (optional)

**global input parameters:**

\$user  
\$mgmt\_config  
\$hcms\_lang  
\$lang

**Output:**

true [report=no] or report [report=yes]  
false on error

**Description:**

Removes all versions of all objects or templates

### 7.1.45 loadfile\_header

**Syntax:**

loadfile\_header (\$abs\_path, \$filename)

**Input parameters:**

\$abs\_path ... path to file  
\$filename ... file name

**global input parameters:**

\$user  
\$mgmt\_config  
\$hcms\_lang  
\$lang

**Output:**

file content

**Description:**

Loads the file header, represented by a defined header size.

### 7.1.46 loadfile\_fast

**Syntax:**

loadfile\_fast (\$abs\_path, \$filename)

**Input parameters:**

\$abs\_path ... path to file  
\$filename ... file name

**global input parameters:**

\$user  
\$mgmt\_config  
\$hcms\_lang  
\$lang

**Output:**

file content

**Description:**

This functions is identical to loadfile, but it does not wait for locked files to be unlocked again. It should only be used on files that won't be locked by the system. It is therefore recommended to use loadfile.

### 7.1.47 loadfile

**Syntax:**

loadfile (\$abs\_path, \$filename)

**Input parameters:**

\$abs\_path ... path to file  
\$filename ... file name

**global input parameters:**

\$user  
\$mgmt\_config  
\$hcms\_lang  
\$lang

**Output:**

file content

**Description:**

This function loads a file and waits up to 3 seconds for locked files to be unlocked

### 7.1.48 loadlockfile

**Syntax:**

loadlockfile (\$user, \$abs\_path, \$filename, \$force\_unlock=3)

**Input parameters:**

\$user ... user

\$abs\_path ... path to file

\$filename ... file name

\$force\_unlock ... force unlock of file after x seconds [integer]

**global input parameters:**

\$mgmt\_config

\$hcms\_lang

\$lang

**Output:**

file content

**Description:**

This function loads and locks a file for a sepecific user. It waits up to 3 seconds for locked files to be unlocked.

Function loadlockfile and savelockfile includes a locking mechanismen for files.

Every time you want to lock a file during your operations use loadlockfile.

It is important to use savelockfile to save and unlock the file again.

savelockfile requires the file to be opened by loadlockfile before.

### 7.1.49 savefile

**Syntax:**

savefile (\$abs\_path, \$filename, \$filedata)

**Input parameters:**

\$abs\_path ... path to file

\$filename ... file name

\$filedata ... file content

**global input parameters:**

\$user

\$mgmt\_config

\$hcms\_lang

\$lang

**Output:**

true/false

**Description:**

This function saves content to a file

### 7.1.50 savelockfile

**Syntax:**

savelockfile (\$user, \$abs\_path, \$filename, \$filedata)

**Input parameters:**

\$user ... user  
\$abs\_path ... path to file  
\$filename ... file name  
\$filedata ... file content

**global input parameters:**

\$mgmt\_config  
\$hcms\_lang  
\$lang

**Output:**

true/false

**Description:**

Saves content to a locked file. It requires the file to be opened by loadlockfile. Function loadlockfile and savelockfile includes a locking mechanism for files. Every time you want to lock a file during your operations use loadlockfile. It is important to use savelockfile to save and unlock the file again. savelockfile requires the file to be opened by loadlockfile before.

### 7.1.51      appendfile

**Syntax:**

appendfile (\$abs\_path, \$filename, \$filedata)

**Input parameters:**

\$abs\_path ... path to file  
\$filename ... file name  
\$filedata ... file content

**global input parameters:**

\$user  
\$mgmt\_config  
\$hcms\_lang  
\$lang

**Output:**

true/false

**Description:**

Appends data to a file but cannot create a new file!  
Waits up to 3 seconds for locked files to be unlocked again.  
Files won't be unlocked if the file is already locked.

### 7.1.52      lockfile

**Syntax:**

lockfile (\$user, \$abs\_path, \$filename)

**Input parameters:**

\$user ... user  
\$abs\_path ... path to file  
\$filename ... file name

**global input parameters:**

\$mgmt\_config  
\$hcms\_lang  
\$lang

**Output:**

true/false

**Description:**

This functions lockes a file for a specific user

### 7.1.53      unlockfile

**Syntax:**

unlockfile (\$user, \$abs\_path, \$filename)

**Input parameters:**

\$user ... user

\$abs\_path ... path to file

\$filename ... file name

**global input parameters:**

\$mgmt\_config

\$hcms\_lang

\$lang

**Output:**

true/false

**Description:**

This functions unlocks a file for a specific user

### 7.1.54      deletefile

**Syntax:**

deletefile (\$abs\_path, \$filename, \$recursive=false)

**Input parameters:**

\$abs\_path ... path to file

\$filename ... file or directory name

\$recursive ... delete all files in directory recursively including symbolic links [true, false]

**global input parameters:**

\$user

\$mgmt\_config

\$hcms\_lang

\$lang

**Output:**

true/false

### 7.1.55      restoremediafile

**Syntax:**

restoremediafile (\$site, \$mediafile)

**Input parameters:**

\$site ... publication name

\$mediafile ... media file name

**global input parameters:**

\$mgmt\_config



**Output:**

result array

**Description:**

Moves an exported media file back to the media repository

### 7.1.56      preparemediafile

**Syntax:**

preparemediafile (\$site, \$medialocation, \$mediafile, \$user="")

**Input parameters:**

\$site ... publication name

\$medialocation ... media file location

\$mediafile ... media file name

\$user ... user name (optional)

**global input parameters:**

\$mgmt\_config

**Output:**

result array / false on error

**Description:**

Prepares a media file for use in the system (load from cloud, decrypt content)

### 7.1.57      deletemediainfiles

**Syntax:**

deletemediainfiles (\$site, \$mediafile, \$delete\_original=false)

**Input parameters:**

\$site ... publication name

\$mediafile ... mediafile name

\$delete\_original ... delete original media file [true, false] (optional)

**global input parameters:**

\$user

\$mgmt\_config

\$mgmt\_mediaoptions

\$mgmt\_docoptions

\$hcms\_ext

\$hcms\_lang

\$lang

**Output:**

true/false

**Description:**

Deletes all derivatives (thumbnails, config files, converted versions of the file) of a specific media file resource. Deletes the original media file optionally .

### 7.1.58      avoidfilecollision

**Syntax:**

avoidfilecollision (\$data="tempdata")

**Input parameters:**

\$data ... data string (optional)

**global input parameters:**

\$mgmt\_config

\$site

**Output:**

true / false on error

**Description:**

Appending data to a file ensures that the previous write process is finished (required due to issue when editing encrypted files)

### 7.1.59      substr\_in\_array

**Syntax:**

substr\_in\_array (\$search, \$array)

**Input parameters:**

\$search ... search-string

\$array ... array

**Output:**

array with found values / false

**Description:**

Searches for substring in array

### 7.1.60      downloadobject

**Syntax:**

downloadobject (\$location, \$object, \$container="", \$lang="en", \$user="")

**Input parameters:**

\$location ... location

\$object ... object name

\$container ... content container

\$lang ... language (optional)

\$user ... user name (optional)

**global input parameters:**

\$mgmt\_config

\$eventsystem

\$hcms\_lang

\$lang

**Output:**

stream of file content / false on error

**Description:**

This functions provides an object via http for viewing, not suitable for multimedia objects!

### 7.1.61      downloadfile

**Syntax:**

downloadfile (\$filepath, \$name, \$force="wrapper", \$user="")

**Input parameters:**

\$filepath ... path to file [string]  
\$name ... file name to show for download via http  
\$force ... force file wrapper or download or no file headers for WebDAV [download, wrapper, noheader]  
\$user ... user name (optional)

**global input parameters:**

\$mgmt\_config  
\$eventsystem  
\$hcms\_lang  
\$lang  
\$is\_iphone

**Output:**

stream of file content / false on error

**Description:**

This functions provides a file via http for view or download

### 7.1.62 loadcontainer

**Syntax:**

loadcontainer (\$container, \$type="work", \$user)

**Input parameters:**

\$container ... container file name or container id (working container will be loaded by default)  
\$type ... optional container type [published  
\$user ... work

**global input parameters:**

\$mgmt\_config  
\$hcms\_lang  
\$lang

**Output:**

XML content of container / false on error

**Description:**

This functions loads a content container

### 7.1.63 savecontainer

**Syntax:**

savecontainer (\$container, \$type="work", \$data, \$user, \$init=false)

**Input parameters:**

\$container ... container file name or container id (working container will be loaded by default)  
\$type ... optional container type [published, work, version]  
\$data ... container content  
\$user ... user  
\$init ... save container initially [true, false] (optional)

**global input parameters:**

\$mgmt\_config  
\$hcms\_lang  
\$lang

**Output:**

true / false on error

**Description:**

Saves data into existing content container by default. Only if \$init is set to true it will initially save a non existing container.

### 7.1.64      getworkflowitem

**Syntax:**

getworkflowitem (\$site, \$workflow\_file, \$workflow, \$user)

**Input parameters:**

\$site ... publication name [string]

\$workflow\_file ... location name [string]

\$workflow ... object name [string]

\$user ... workflow file name [string]

**global input parameters:**

\$mgmt\_config

\$hcms\_lang

\$lang

**Output:**

workflow item [XML-string]

### 7.1.65      checkworkflow

**Syntax:**

checkworkflow (\$site, \$location, \$page, \$cat="", \$contentfile="", \$contentdata="", \$buildview="cmsview", \$viewstore="", \$user)

**Input parameters:**

\$site ... publication name [string]

\$location ... location [string]

\$page ... object name [string]

\$cat ... category [page, comp] (optional)

\$contentfile ... container name [string] (optional)

\$contentdata ... container [XML string] (optional)

\$buildview ... view name [string] (optional)

\$viewstore ... view store [string] (optional)

\$user ... user name [string]

**global input parameters:**

\$mgmt\_config

**Output:**

result array

**Description:**

Help function for function buildview to evaluate the workflow of an object and return the manipulated view store, view name, workflow ID, workflow role and the encrypted workflow token.

### 7.1.66      inherit\_db\_load

**Syntax:**

inherit\_db\_load (\$user)

**Input parameters:**

\$user ... nothing

**global input parameters:**

\$siteaccess  
\$mgmt\_config  
\$hcms\_lang  
\$lang

**Output:**

inheritance database [2 dim. array]/false

**Description:**

This function loads and locks the inheritance database.

Each record of the inherit management database has the following design:

xml-content container : | absolute path to 1-n objects : | 1-m inherits used by 1-n objects

Important: The inherit management database needs to be saved or closed after loading it

### 7.1.67 inherit\_db\_read

**Syntax:**

inherit\_db\_read ()

**Input parameters:****global input parameters:**

\$user  
\$mgmt\_config  
\$hcms\_lang  
\$lang

**Output:**

inheritance database [2 dim. array]/false

**Description:**

This function loads the inheritance database for reading

### 7.1.68 inherit\_db\_close

**Syntax:**

inherit\_db\_close (\$user)

**Input parameters:**

\$user ... %

**global input parameters:**

\$mgmt\_config  
\$hcms\_lang  
\$lang

**Output:**

true/false

**Description:**

Closes and unlocks the inheritance management database

### 7.1.69 inherit\_db\_save

**Syntax:**

inherit\_db\_save (\$inherit\_db, \$user)

**Input parameters:**

\$inherit\_db ... inherit database array

\$user

**global input parameters:**

\$mgmt\_config

\$hcms\_lang

\$lang

**Output:**

true/false

**Description:**

This function saves und unlocks the inheritance management database

### 7.1.70 inherit\_db\_getparent

**Syntax:**

inherit\_db\_getparent (\$inherit\_db, \$child)

**Input parameters:**

\$inherit\_db ... inherit database [2 dim. array]

\$child ... child [string]

**Output:**

all parents of given child [1 dim. array] / false

### 7.1.71 inherit\_db\_getchild

**Syntax:**

inherit\_db\_getchild (\$inherit\_db, \$parent)

**Input parameters:**

\$inherit\_db ... parent [string]

\$parent

**Output:**

all childs of given parent [1 dim. array] / false

### 7.1.72 inherit\_db\_setparent

**Syntax:**

inherit\_db\_setparent (\$inherit\_db, \$child, \$parent\_array)

**Input parameters:**

\$inherit\_db ... inherit database [2 dim. array]

\$child ... child [string]

\$parent\_array ... parents [Array]

**Output:**

inherit database [2 dim. array]

**Description:**

This function updates and insert all references from a child an its parents

### 7.1.73 inherit\_db\_insertparent

**Syntax:**

inherit\_db\_insertparent (\$inherit\_db, \$parent, \$child\_array)

**Input parameters:**

\$inherit\_db ... inherit database [2 dim. array]

\$parent ... parent [string]

\$child\_array ... childs [Array]

**Output:**

inherit database [2 dim. array]

### 7.1.74 inherit\_db\_deleteparent

**Syntax:**

inherit\_db\_deleteparent (\$inherit\_db, \$parent)

**Input parameters:**

\$inherit\_db ... inherit database [2 dim. array]

\$parent ... parent [string]

**Output:**

inherit database [2 dim. array]

### 7.1.75 getconfigvalue

**Syntax:**

getconfigvalue (\$config, \$in\_key="")

**Input parameters:**

\$config ... settings array

\$in\_key ... value/substring in array key (optional)

**Output:**

value of setting

**Description:**

Help function for createinstance

### 7.1.76 createinstance

**Syntax:**

createinstance (\$instance\_name, \$settings, \$user="sys")

**Input parameters:**

\$instance\_name ... instance name

\$settings ... settings array

\$user ... user name

**global input parameters:**

\$eventsystem

\$mgmt\_config

\$hcms\_lang

\$lang

**Output:**

result array

**Description:**

This function creates a new instance with all its files and the mySQL database

### 7.1.77      editinstance

**Syntax:**

editinstance (\$instance\_name, \$content, \$user="sys")

**Input parameters:**

\$instance\_name ... instance name

\$content ... content as string

\$user ... user name

**global input parameters:**

\$eventsystem

\$mgmt\_config

\$hcms\_lang

\$lang

**Output:**

result array

**Description:**

This function saves the instance configuration in the config file

### 7.1.78      deleteinstance

**Syntax:**

deleteinstance (\$instance\_name, \$user="sys")

**Input parameters:**

\$instance\_name ... instance name

\$user ... settings array

**global input parameters:**

\$eventsystem

\$mgmt\_config

\$hcms\_lang

\$lang

**Output:**

result array

**Description:**

This function creates a new instance with all its files and the mySQL database

### 7.1.79      createpublication

**Syntax:**

createpublication (\$site\_name, \$user="sys")

**Input parameters:**

\$site\_name ... publication name



\$user ... user name

**global input parameters:**

\$eventsystem  
\$mgmt\_config  
\$hcms\_lang  
\$lang

**Output:**

result array

**Description:**

This function creates a new publication with all its files

## 7.1.80      editpublication

**Syntax:**

editpublication (\$site\_name, \$setting, \$user="sys")

**Input parameters:**

\$site\_name ... publication name  
\$setting ... publication settings array  
\$user ... user name

**global input parameters:**

\$eventsystem  
\$mgmt\_config  
\$hcms\_lang  
\$lang

**Output:**

result array

**Description:**

This function saves the settings of a publication

## 7.1.81      editpublicationsetting

**Syntax:**

editpublicationsetting (\$site\_name, \$setting, \$value, \$user="sys")

**Input parameters:**

\$site\_name ... publication name  
\$setting ... publication settings name ['site\_admin', 'linkengine', 'sendmail', 'webdav',  
'http\_incl', 'inherit\_obj', 'inherit\_comp', 'inherit\_tpl', 'specialchr\_disable', 'default\_codepage',  
'exclude\_folders']  
\$value ... value  
\$user ... user name

**global input parameters:**

\$eventsystem  
\$mgmt\_config  
\$hcms\_lang  
\$lang

**Output:**

true/false

**Description:**

This function can be used to edit a single settings of a publication

### 7.1.82      deletepublication

**Syntax:**

deletepublication (\$site\_name, \$user="sys")

**Input parameters:**

\$site\_name ... publication name

\$user ... user name

**global input parameters:**

\$mgmt\_config

\$eventsystem

\$hcms\_lang

\$lang

**Output:**

result array

**Description:**

This function deletes a publication with all its files

### 7.1.83      createpersonalization

**Syntax:**

createpersonalization (\$site, \$pers\_name, \$cat)

**Input parameters:**

\$site ... site

\$pers\_name ... personalization profile or tracking name

\$cat ... category [profile, tracking]

**global input parameters:**

\$user

\$eventsystem

\$mgmt\_config

\$hcms\_lang

\$lang

**Output:**

result array

**Description:**

This function creates a new customer personalization tracking or profile

### 7.1.84      deletepersonalization

**Syntax:**

deletepersonalization (\$site, \$pers\_name, \$cat)

**Input parameters:**

\$site ... site

\$pers\_name ... personalization profile or tracking name

\$cat ... category [profile, tracking]

**global input parameters:**

\$user  
\$eventssystem  
\$mgmt\_config  
\$hcms\_lang  
\$lang

**Output:**

result array

**Description:**

This function deletes a customer personalization tracking or profile

## 7.1.85 createtemplate

**Syntax:**

createtemplate (\$site, \$template, \$cat)

**Input parameters:**

\$site ... site  
\$template ... template name  
\$cat ... category [page, comp, meta, inc]

**global input parameters:**

\$user  
\$eventssystem  
\$mgmt\_config  
\$hcms\_lang  
\$lang

**Output:**

result array

**Description:**

This function creates a new template

## 7.1.86 gettemplates

**Syntax:**

gettemplates (\$site, \$cat)

**Input parameters:**

\$site ... publication name  
\$cat ... object category [page, comp, meta]

**global input parameters:**

\$user  
\$mgmt\_config  
\$hcms\_lang  
\$lang

**Output:**

template file name list as array / false on error

**Description:**

This function returns a list of all templates for pages or components.  
Based on the inheritance settings of the publication the template will be loaded with highest priority from the own publication and if not available from a parent publication.

### 7.1.87 loadtemplate

**Syntax:**

loadtemplate (\$site, \$template)

**Input parameters:**

\$site ... publication name

\$template ... template file name

**global input parameters:**

\$user

\$mgmt\_config

\$hcms\_lang

\$lang

**Output:**

array (template content [XML string])

publication

result[true/false] / false on error

**Description:**

This function loads templates by given name.

Based on the inheritance settings of the publication the template will be loaded with highest priority from the own publication and if not available from a parent publication. If the parent publications have double entries the sort mechanism will define the priority. First priority have numbers, second are upper case letters and last priority have lower case letters.

### 7.1.88 edittemplate

**Syntax:**

edittemplate (\$site, \$template, \$cat, \$user, \$content="", \$extension="", \$application="")

**Input parameters:**

\$site ... site

\$template ... template file name

\$cat ... category [page, comp, meta, inc]

\$user ... template content (optional)

\$content ... template extension (optional)

\$extension ... template application (optional)

\$application

**global input parameters:**

\$eventsystem

\$mgmt\_config

\$hcms\_lang

\$lang

**Output:**

result array

**Description:**

This function edits a template

### 7.1.89 deletetemplate

**Syntax:**

deletetemplate (\$site, \$template, \$cat)

**Input parameters:**

\$site ... site  
\$template ... template file name  
\$cat ... category [page, comp, meta, inc]

**global input parameters:**

\$user  
\$eventsystem  
\$mgmt\_config  
\$hcms\_lang  
\$lang

**Output:**

result array

**Description:**

This function deletes a template

### 7.1.90 createuser

**Syntax:**

createuser (\$site, \$login, \$password, \$confirm\_password, \$user="sys")

**Input parameters:**

\$site ... publication name (optional)  
\$login ... login name  
\$password ... password  
\$confirm\_password ... confirmed password  
\$user ... user name

**global input parameters:**

\$eventsystem  
\$mgmt\_config  
\$mgmt\_lang\_shortcut\_default  
\$hcms\_lang  
\$lang

**Output:**

array

**Description:**

This function creates a new user

### 7.1.91 edituser

**Syntax:**

edituser (\$site, \$login, \$old\_password="", \$password="", \$confirm\_password="",  
\$superadmin="0", \$realname="", \$language="en", \$theme="", \$email="", \$signature="",  
\$usergroup="", \$usersite="", \$user="sys")

**Input parameters:**

\$site ... publication name  
\$login ... login name  
\$old\_password ... new login name  
\$password ... password  
\$confirm\_password ... confirmed password  
\$superadmin ... super administrator [0, 1]

\$realname ... real name  
\$language ... language setting [de, en]  
\$theme ... theme name (optional)  
\$email ... email  
\$signature ... usergroup string [group1|group2]  
\$usergroup ... member of site(s) string [site1|site2]]  
\$usersite ... user name  
\$user

**global input parameters:**

\$eventsystem  
\$login\_cat  
\$group  
\$mgmt\_config  
\$hcms\_lang  
\$lang

**Output:**

array

**Description:**

This function edits a user

7.1.92      deleteuser

**Syntax:**

deleteuser (\$site, \$login, \$user="sys")

**Input parameters:**

\$site ... publication where the user should be removed [\*Null\*] for all publications  
\$login ... login name  
\$user ... user name

**global input parameters:**

\$eventsystem  
\$mgmt\_config  
\$hcms\_lang  
\$lang

**Output:**

array

**Description:**

This function removes a user

7.1.93      creategroup

**Syntax:**

creategroup (\$site, \$group\_name, \$user="sys")

**Input parameters:**

\$site ... publication anem  
\$group\_name ... group name  
\$user ... user name

**global input parameters:**

\$eventsystem  
\$mgmt\_config

\$hcms\_lang  
\$lang

**Output:**

array

**Description:**

This function creates a new user group

## 7.1.94      editgroup

**Syntax:**

editgroup (\$site, \$group\_name, \$pageaccess, \$compaccess, \$permission, \$user)

**Input parameters:**

\$site ... publication name  
\$group\_name ... group name  
\$pageaccess ... page folder access array  
\$compaccess ... component folder access array  
\$permission ... permissions array  
\$user ... user name

**global input parameters:**

\$eventsystem  
\$mgmt\_config  
\$hcms\_lang  
\$lang

**Output:**

array

**Description:**

This function removes a user group

## 7.1.95      deletegroup

**Syntax:**

deletegroup (\$site, \$group\_name, \$user)

**Input parameters:**

\$site ... publication name  
\$group\_name ... group name  
\$user ... user name

**global input parameters:**

\$eventsystem  
\$mgmt\_config  
\$hcms\_lang  
\$lang

**Output:**

array

**Description:**

This function removes a user group

### 7.1.96      renamegroupfolder

**Syntax:**

renamegroupfolder (\$site, \$cat, \$folder\_curr, \$folder\_new, \$user)

**Input parameters:**

\$site ... site

\$cat ... cat[page, comp]

\$folder\_curr ... old location

\$folder\_new ... new location

\$user ... user

**global input parameters:**

\$mgmt\_config

\$hcms\_lang

\$lang

**Output:**

true / false on error

**Description:**

This function renames a workgroup folder

### 7.1.97      deletegroupfolder

**Syntax:**

deletegroupfolder (\$site, \$cat, \$folderpath, \$user)

**Input parameters:**

\$site ... site

\$cat ... cat[page, comp]

\$folderpath ... path to the folder

\$user ... user

**global input parameters:**

\$mgmt\_config

\$hcms\_lang

\$lang

**Output:**

true / false on error

**Description:**

This function removes a group folder

### 7.1.98      renameworkflowfolder

**Syntax:**

renameworkflowfolder (\$site, \$cat, \$folder\_curr, \$folder\_new, \$user)

**Input parameters:**

\$site ... site

\$cat ... cat[page, comp]

\$folder\_curr ... old location

\$folder\_new ... new location

\$user ... user

**global input parameters:**



\$mgmt\_config  
\$hcms\_lang  
\$lang

**Output:**

true / false on error

**Description:**

This function renames a workgroup folder

## 7.1.99      deleteworkflowfolder

**Syntax:**

deleteworkflowfolder (\$site, \$cat, \$folderpath, \$user)

**Input parameters:**

\$site ... site  
\$cat ... cat[page, comp]  
\$folderpath ... location of folder  
\$user ... user

**global input parameters:**

\$mgmt\_config  
\$hcms\_lang  
\$lang

**Output:**

true / false on error

**Description:**

This function removes a workgroup folder

## 7.1.100      createmediacat

**Syntax:**

createmediacat (\$site, \$mediacat\_name)

**Input parameters:**

\$site ... publication  
\$mediacat\_name ... media category name

**global input parameters:**

\$user  
\$eventsystem  
\$mgmt\_config  
\$hcms\_lang  
\$lang

**Output:**

Array with onload JS-code and message

**Description:**

Creates a new media category in the template media index file. Only used for template media.

## 7.1.101      renamemediacat

**Syntax:**

renamemediacat (\$site, \$mediacat\_name\_curr, \$mediacat\_name)

**Input parameters:**

\$site ... publication

\$mediacat\_name\_curr ... old media category name

\$mediacat\_name ... new media category name

**global input parameters:**

\$user

\$eventsystem

\$mgmt\_config

\$hcms\_lang

\$lang

**Output:**

Array with onload JS-code and message

**Description:**

Renames a new media category in the template media index file

### 7.1.102 deletemediacat

**Syntax:**

deletemediacat (\$site, \$mediacat\_name)

**Input parameters:**

\$site ... publication

\$mediacat\_name ... media category name

**global input parameters:**

\$user

\$eventsystem

\$mgmt\_config

\$hcms\_lang

\$lang

**Output:**

Array with onload JS-code and message

**Description:**

Deletes a new media category in the template media index file

### 7.1.103 uploadtomediacat

**Syntax:**

uploadtomediacat (\$site, \$mediacat\_name, \$global\_files)

**Input parameters:**

\$site ... publication

\$mediacat\_name ... media category name

\$global\_files ... PHP FILES array

**global input parameters:**

\$user

\$eventsystem

\$mgmt\_config

\$hcms\_lang

\$lang

**Output:**

Array with onload JS-code and message

**Description:**

Uploads a media file into a given template media category

### 7.1.104      deletefrommediacat

**Syntax:**

deletefrommediacat (\$site, \$mediafile)

**Input parameters:**

\$site ... publication

\$mediafile ... media file name

**global input parameters:**

\$user

\$eventsystem

\$mgmt\_config

\$hcms\_lang

\$lang

**Output:**

Array with onload JS-code and message

**Description:**

Deletes a media file from the template media category index

### 7.1.105      createfolder

**Syntax:**

createfolder (\$site, \$location, \$foldernew, \$user)

**Input parameters:**

\$site ... site

\$location ... location

\$foldernew ... folder

\$user ... user

**global input parameters:**

\$eventsystem

\$mgmt\_config

\$cat

\$hcms\_lang

\$lang

**Output:**

array

**Description:**

This function creates a new folder

### 7.1.106      createfolders

**Syntax:**

createfolders (\$site, \$location, \$foldernew, \$user)

**Input parameters:**

\$site ... site  
\$location ... location  
\$foldernew ... folder  
\$user ... user

**global input parameters:**

\$eventsystem  
\$mgmt\_config  
\$cat  
\$hcms\_lang  
\$lang

**Output:**

array

**Description:**

This function creates all folders recursively

### 7.1.107 collectfolders

**Syntax:**

collectfolders (\$site, \$location, \$folder)

**Input parameters:**

\$site ... site  
\$location ... location  
\$folder ... folder name

**global input parameters:**

\$user  
\$mgmt\_config  
\$hcms\_lang  
\$lang

**Output:**

result array / false

**Description:**

Help function to create the collection of folders

### 7.1.108 copyfolders

**Syntax:**

copyfolders (\$site, \$location, \$locationnew, \$folder, \$user)

**Input parameters:**

\$site ... site  
\$location ... location (source)  
\$locationnew ... new location (destination)  
\$folder ... folder  
\$user

**global input parameters:**

\$mgmt\_config  
\$cat  
\$hcms\_lang  
\$lang

**Output:**

result array equal to createfolder

**Description:**

This function copies/creates all folders of the source location using mkdir (only directories will be created!). used for pasteobject function.

### 7.1.109      deletefolder

**Syntax:**

deletefolder (\$site, \$location, \$folder, \$user)

**Input parameters:**

\$site ... site  
\$location ... location  
\$folder ... folder  
\$user

**global input parameters:**

\$eventsystem  
\$mgmt\_config  
\$cat  
\$hcms\_lang  
\$lang

**Output:**

array

**Description:**

This function removes a folder

### 7.1.110      renamefolder

**Syntax:**

renamefolder (\$site, \$location, \$folder, \$foldernew, \$user)

**Input parameters:**

\$site ... site  
\$location ... location  
\$folder ... folder  
\$foldernew ... new folder name  
\$user ... user

**global input parameters:**

\$eventsystem  
\$mgmt\_config  
\$cat  
\$hcms\_lang  
\$lang

**Output:**

array

**Description:**

This function renames a folder

### 7.1.111 createobject

**Syntax:**

createobject (\$site, \$location, \$page, \$template, \$user)

**Input parameters:**

\$site ... site  
\$location ... location  
\$page ... object  
\$template ... template  
\$user

**global input parameters:**

\$eventsystem  
\$mgmt\_config  
\$hcms\_lang  
\$lang

**Output:**

result array

**Description:**

This function creates a new page or component

### 7.1.112 uploadfile

**Syntax:**

uploadfile (\$site, \$location, \$cat, \$global\_files, \$page="", \$unzip=0, \$createthumbnail=0, \$imageresize="", \$imagepercentage="", \$user="sys", \$checkduplicates=true, \$versioning=false)

**Input parameters:**

\$site ... publication name  
\$location ... destination location  
\$cat ... category [page/comp]  
\$global\_files ... uploaded file (array as defined by PHP autoglobale \$\_FILES)  
\$page ... unzip [1/0]  
\$unzip ... object name (only for media file update of existing object)  
\$createthumbnail ... create only a new thumbnail [1/0]  
\$imageresize ... imageresize [percentage  
\$imagepercentage ... null]  
\$user ... imagepercentage (%-value as integer)  
\$checkduplicates ... user name  
\$versioning ... check for duplicates [true, false]

**global input parameters:**

\$mgmt\_config  
\$mgmt\_uncompress  
\$mgmt\_imagepreview  
\$mgmt\_mediapreview  
\$mgmt\_mediaoptions  
\$mgmt\_imageoptions  
\$mgmt\_maxsizepreview  
\$mgmt\_parser  
\$eventsystem  
\$pageaccess  
\$compaccess  
\$hiddenfolder

\$localpermission  
\$hcms\_lang  
\$lang

**Output:**

result array

**Description:**

This function manages all file uploads, like unzip files, create media objects and resize images.

The container name will be extracted from the media file name for updating an existing multimedia file.

### 7.1.113 createmediaobject

**Syntax:**

createmediaobject (\$site, \$location, \$file, \$path\_source\_file, \$user, \$imagepercentage=0)

**Input parameters:**

\$site ... site  
\$location ... destination location  
\$file ... file name  
\$path\_source\_file ... path to source multimedia file (uploaded file in temp directory)  
\$user ... user  
\$imagepercentage ... resize original image (100%) by percentage (optional)

**global input parameters:**

\$mgmt\_config  
\$mgmt\_imageoptions  
\$eventsystem  
\$hcms\_lang  
\$lang

**Output:**

Array

**Description:**

This function creates an asset (multimedia object) by reading a given source file

### 7.1.114 createmediaobjects

**Syntax:**

createmediaobjects (\$site, \$location\_source, \$location\_destination, \$user)

**Input parameters:**

\$site ... site  
\$location\_source ... source location  
\$location\_destination ... destination location  
\$user ... user

**global input parameters:**

\$mgmt\_config  
\$mgmt\_imageoptions  
\$eventsystem  
\$hcms\_lang  
\$lang

**Output:**

result array with all objects created / false

**Description:**

This function creates media objects by reading all media files from a given source location (used after unzipfile)

### 7.1.115 editmediaobject

**Syntax:**

editmediaobject (\$site, \$location, \$page, \$format="jpg", \$type="thumbnail", \$user)

**Input parameters:**

\$site ... publication

\$location ... locationr

\$page ... object name

\$format ... format (file extension w/o dot) (optional)

\$type ... type of image/video/audio file [thumbnail, origthumb(thumbnail made from original video/audio), original, any other string present in \$mgmt\_imageoptions] (optional)

\$user

**global input parameters:**

\$wf\_token

\$mgmt\_config

\$mgmt\_imagepreview

\$mgmt\_mediapreview

\$mgmt\_mediaoptions

\$mgmt\_imageoptions

\$mgmt\_maxsizepreview

\$mgmt\_mediametadata

\$hcms\_ext

\$hcms\_lang

\$lang

**Output:**

result array / false on error (saves original or thumbnail media file of an object

for thumbnail only jpeg format is supported as output)

user name

**Description:**

This function mainly uses function createmedia to render the objects media, but at the same time takes care of versioning and the object name, if the file extension has been changed

### 7.1.116 manipulateobject

**Syntax:**

manipulateobject (\$site, \$location, \$page, \$pagenew, \$user, \$action)

**Input parameters:**

\$site ... site

\$location ... location

\$page ... object name

\$pagenew ... new object name (exkl. extension except for action "file\_rename")

\$user ... user

\$action ... action [page\_delete

**global input parameters:**

\$wf\_token

\$eventsystem



\$mgmt\_config  
\$mgmt\_mediaoptions  
\$mgmt\_docoptions  
\$hcms\_ext  
\$pageaccess  
\$compaccess  
\$hiddenfolder  
\$cat  
\$temp\_clipboard  
\$hcms\_lang  
\$lang

**Output:**

array

**Description:**

This function removes, unpublishes, renames and pastes objects and is used by other functions which works as a shell for this function

### 7.1.117 deleteobject

**Syntax:**

deleteobject (\$site, \$location, \$page, \$user)

**Input parameters:**

\$site ... site  
\$location ... location  
\$page ... object  
\$user

**global input parameters:**

\$eventsystem  
\$mgmt\_config  
\$cat  
\$hcms\_lang  
\$lang

**Output:**

array

**Description:**

This function removes page or component and calls the function manipulateobject

### 7.1.118 renameobject

**Syntax:**

renameobject (\$site, \$location, \$page, \$pagenew, \$user)

**Input parameters:**

\$site ... site  
\$location ... location  
\$page ... object  
\$pagenew ... new object name exkl. file extension  
\$user ... user

**global input parameters:**

\$eventsystem  
\$mgmt\_config

\$cat  
\$hcms\_lang  
\$lang

**Output:**

array

**Description:**

This function renames a page or component and calls the function manipulateobject

### 7.1.119      renamefile

**Syntax:**

renamefile (\$site, \$location, \$page, \$pagenew, \$user)

**Input parameters:**

\$site ... publication name  
\$location ... location  
\$page ... object  
\$pagenew ... new object including file extension  
\$user ... user name

**global input parameters:**

\$eventsystem  
\$mgmt\_config  
\$cat  
\$hcms\_lang  
\$lang

**Output:**

array

**Description:**

This function renames a file (NOT a page or component) and calls the function manipulateobject.

This function renames the file name including the extension and not only the name of an object.

The event that will be executed in the event system is the same as renameobject.

### 7.1.120      cutobject

**Syntax:**

cutobject (\$site, \$location, \$page, \$user, \$clipboard\_add=false)

**Input parameters:**

\$site ... publication name[string]  
\$location ... location[string]  
\$page ... object[string]  
\$user ... user[string]  
\$clipboard\_add ... add to clipboard to save more entries (optional)

**global input parameters:**

\$eventsystem  
\$mgmt\_config  
\$cat  
\$temp\_clipboard  
\$hcms\_lang  
\$lang

**Output:**

array

**Description:**

This function cuts a page or component

### 7.1.121      copyobject

**Syntax:**

copyobject (\$site, \$location, \$page, \$user, \$clipboard\_add=false)

**Input parameters:**

\$site ... publication name[string]

\$location ... location[string]

\$page ... object[string]

\$user ... user[string]

\$clipboard\_add ... add to clipboard to save more entries (optional)

**global input parameters:**

\$eventsystem

\$mgmt\_config

\$cat

\$temp\_clipboard

\$hcms\_lang

\$lang

**Output:**

array

**Description:**

This function copies a page or component

### 7.1.122      copyconnectedobject

**Syntax:**

copyconnectedobject (\$site, \$location, \$page, \$user, \$clipboard\_add=false)

**Input parameters:**

\$site ... publication name[string]

\$location ... location[string]

\$page ... object[string]

\$user ... user[string]

\$clipboard\_add ... add to clipboard to save more entries (optional)

**global input parameters:**

\$eventsystem

\$mgmt\_config

\$cat

\$temp\_clipboard

\$hcms\_lang

\$lang

**Output:**

array

**Description:**

This function makes a connected copy of a page or component

### 7.1.123     pasteobject

**Syntax:**

pasteobject (\$site, \$location, \$user)

**Input parameters:**

\$site ... publication name[string]

\$location ... location[string]

\$user ... user[string]

**global input parameters:**

\$eventsystem

\$mgmt\_config

\$cat

\$temp\_clipboard

\$hcms\_lang

\$lang

**Output:**

array

**Description:**

This function pastes a page or component and calls the function manipulateobject

### 7.1.124     lockobject

**Syntax:**

lockobject (\$site, \$location, \$page, \$user)

**Input parameters:**

\$site ... site

\$location ... location

\$page ... object

\$user ... user

**global input parameters:**

\$eventsystem

\$mgmt\_config

\$cat

\$temp\_clipboard

\$hcms\_lang

\$lang

**Output:**

array

**Description:**

This function locks a page, component or asset

### 7.1.125     unlockobject

**Syntax:**

unlockobject (\$site, \$location, \$page, \$user)

**Input parameters:**

\$site ... site

\$location ... location

\$page ... object

\$user ... user

**global input parameters:**

\$eventsystem  
\$mgmt\_config  
\$cat  
\$temp\_clipboard  
\$hcms\_lang  
\$lang

**Output:**

array

**Description:**

This function unlocks a page or component

### 7.1.126      publishobject

**Syntax:**

publishobject (\$site, \$location, \$page, \$user)

**Input parameters:**

\$site ... site  
\$location ... location  
\$page ... object (full name incl. extension)  
\$user

**global input parameters:**

\$eventsystem  
\$mgmt\_config  
\$cat  
\$ctrlreload  
\$hcms\_lang  
\$lang

**Output:**

array

**Description:**

This function publishes a page, component or asset

### 7.1.127      processobjects

**Syntax:**

processobjects (\$action, \$site, \$location, \$file, \$published\_only="0", \$user)

**Input parameters:**

\$action ... action [publish  
\$site ... unpublish  
\$location ... delete]  
\$file ... publication  
\$published\_only ... location  
\$user ... object

**global input parameters:**

\$eventsystem  
\$mgmt\_config  
\$hcms\_lang

\$lang

**Output:**

true/false on error

**Description:**

Publish, unpublish or delete all objects recursively.

Should not be used in CMS GUI, only for queue processing, since it does not provide feedback about the process state!

### 7.1.128      publishlinkedobject

**Syntax:**

publishlinkedobject (\$site, \$location, \$page, \$user)

**Input parameters:**

\$site ... site

\$location ... location

\$page ... object

\$user ... user name

**global input parameters:**

\$eventsystem

\$mgmt\_config

\$cat

\$ctrlreload

\$hcms\_lang

\$lang

**Output:**

array

**Description:**

This function publishes all linked objects of a given object.

All objects with component links (references) to the given object will be published.

This function is only used by publishobject.

### 7.1.129      unpublishobject

**Syntax:**

unpublishobject (\$site, \$location, \$page, \$user)

**Input parameters:**

\$site ... site

\$location ... location

\$page ... object

\$user

**global input parameters:**

\$eventsystem

\$mgmt\_config

\$cat

\$ctrlreload

\$hcms\_lang

\$lang

**Output:**

array

**Description:**

This function unpublishes a page or component and calls the function manipulateobject

### 7.1.130 collectobjects

**Syntax:**

collectobjects (\$root\_id, \$site, \$cat, \$location, \$published\_only="0")

**Input parameters:**

\$root\_id ... root ID

\$site ... publication name

\$cat ... category [page, comp]

\$location ... location

\$published\_only ... collect only published objects [0, 1]

**global input parameters:**

\$user

\$pageaccess

\$compaccess

\$mgmt\_config

\$hiddenfolder

\$hcms\_lang

\$lang

**Output:**

result array / false

**Description:**

Help function used to create a list of all objects inside a folder

### 7.1.131 manipulateallobjects

**Syntax:**

manipulateallobjects (\$action, \$objectpath\_array, \$method, \$force, \$published\_only, \$user, \$tempfile="")

**Input parameters:**

\$action ... action [publish

\$objectpath\_array ... unpublish

\$method ... delete

\$force ... paste]

\$published\_only ... objectpath (array)

\$user ... method (only for paste action) [copy

\$tempfile ... linkcopy

**global input parameters:**

\$eventsystem

\$mgmt\_config

\$pageaccess

\$compaccess

\$hiddenfolder

\$cat

\$hcms\_lang

\$lang

**Output:**

true/false

**Description:**

This function is used to perform actions on folders with several items. the function will be called by popup\_status.php. To work correctly the functions needs several variables to be passed. please take a look at the \$result array of this function.

### 7.1.132      remoteclient

**Syntax:**

remoteclient (\$action, \$root, \$site, \$location, \$locationnew, \$page, \$pagenew)

**Input parameters:**

\$action ... action [save  
\$root ... copy  
\$site ... delete  
\$location ... rename  
\$locationnew ... get]  
\$page ... root [abs\_path\_link  
\$pagenew ... abs\_path\_media

**global input parameters:**

\$user  
\$mgmt\_config  
\$hcms\_lang  
\$lang

**Output:**

http answer [string] or false

**Description:**

Sends data to remote client via http post

### 7.1.133      HTTP\_Post

**Syntax:**

HTTP\_Post (\$URL, \$data, \$contenttype="application/x-www-form-urlencoded",  
\$charset="UTF-8", \$referrer="")

**Input parameters:**

\$URL ... URL[string]  
\$data ... \$data[array] (raw data)  
\$contenttype ... content-type [application/x-www-form-urlencoded  
\$charset ... multipart/form-data]  
\$referrer ... character set [string]

**Output:**

http response [string] / false on error

**Description:**

Sends data via http post and returns response

### 7.1.134      HTTP\_Get

**Syntax:**

HTTP\_Get (\$URL, \$data="", \$contenttype="application/x-www-form-urlencoded",  
\$charset="UTF-8")



**Input parameters:**

\$URL ... URL[string]  
\$data ... \$data[array] (raw data) (optional)  
\$contenttype ... content-type[string excl. charset] (optional)  
\$charset ... character set[string] (optional)

**Output:**

http response [string] / false on error

**Description:**

Sends data via http get and returns response

### 7.1.135 HTTP\_Proxy

**Syntax:**

HTTP\_Proxy (\$URL, \$enable\_file=false)

**Input parameters:**

\$URL ... URL[string]  
\$enable\_file ... enable post of files [true, false] (optional)

**global input parameters:**

\$mgmt\_config

**Output:**

http response [string] / false on error

**Description:**

Sends all global POST/GET and FILES data via http post and returns response

### 7.1.136 loadbalancer

**Syntax:**

loadbalancer (\$type)

**Input parameters:**

\$type ... type [renderimage, rendervideo, uploadfile]

**global input parameters:**

\$mgmt\_config

**Output:**

http response [string] / false on error or if disabled

**Description:**

Balances the load by sending all global POST/GET and FILES to one service resource of a given array of service resources.  
Don't define and use the same server resources in \$mgmt\_config['url\_path\_service'], this can lead to an infinite loop.

### 7.1.137 savelog

**Syntax:**

savelog (\$error, \$logfile="event")

**Input parameters:**

\$error ... error messages array

\$logfile ... name of log file without extension (optional)

**global input parameters:**

\$user  
\$eventsystem  
\$mgmt\_config  
\$hcms\_lang  
\$lang

**Output:**

true / false on error

**Description:**

adds new entries to log file

An error entry must be formed like:

date[YYYY-MM-DD hh:mm]|name of script file|error type: "error", "warning" or "information"|unique error code in script file|error message

### 7.1.138 deletelog

**Syntax:**

deletelog (\$logname="")

**Input parameters:**

\$logname ... logname (optional)

**global input parameters:**

\$user  
\$eventsystem  
\$mgmt\_config  
\$hcms\_lang  
\$lang

**Output:**

result array

**Description:**

Deletes the log file

### 7.1.139 debuglog

**Syntax:**

debuglog (\$code)

**Input parameters:**

\$code ... code to write to debug file

**global input parameters:**

\$user  
\$mgmt\_config  
\$hcms\_lang  
\$lang

**Output:**

true / false

**Description:**

Writes code lines into debug file in data/log/debug.log

### 7.1.140     notifyusers

**Syntax:**

notifyusers (\$site, \$location, \$object, \$event, \$user\_from)

**Input parameters:**

\$site ... publication name

\$location ... location

\$object ... object name

\$event ... event name [oncreate, onedit, onmove, ondelete]

\$user\_from ... user name

**global input parameters:**

\$user

\$mgmt\_config

\$hcms\_lang\_codepage

\$hcms\_lang

\$lang

**Output:**

true / false on error

**Description:**

Notifies all users based on the given event and location

### 7.1.141     sendlicensenotification

**Syntax:**

sendlicensenotification (\$site, \$cat, \$folderpath, \$text\_id, \$date\_begin, \$date\_end, \$user, \$format="%Y-%m-%d")

**Input parameters:**

\$site ... publication name

\$cat ... category [page, comp]

\$folderpath ... folder path

\$text\_id ... text ID for text field

\$date\_begin ... search from date (YYYY-MM-DD)

\$date\_end ... search till date (YYYY-MM-DD)

\$user ... user name string or array (optional)

\$format ... date format (optional)

**global input parameters:**

\$eventsystem

\$mgmt\_config

\$hcms\_lang\_codepage

\$hcms\_lang

\$lang

**Output:**

true / false on error

**Description:**

Searches for objects with a date in a defined text field that has to be between the defined date limits and sends a message to the defined users.

This is a helper function for function licensenotification.

### 7.1.142 licensenotification

**Syntax:**

licsenotification ()

**Input parameters:****global input parameters:**

\$eventsystem

\$mgmt\_config

\$hcms\_lang\_codepage

\$hcms\_lang

\$lang

**Output:**

true / false on error

**Description:**

This function reads the license notification configuration and looks up all objects with a date in a defined text field that has to be between the defined date limits and sends a message to the defined users.

### 7.1.143 html\_diff

**Syntax:**

html\_diff (\$old, \$new)

**Input parameters:**

\$old ... old text

\$new ... new text

**Output:**

result text showing deleted and inserted words/differences / false on error

**Description:**

Paul's Simple Diff Algorithm v 0.1

Function html\_diff is a wrapper for the diff command, it takes two strings and returns the differences in HTML. The tags used are <ins> and <del>, which can easily be styled with CSS.

### 7.1.144 createfavorite

**Syntax:**

createfavorite (\$site="", \$location="", \$page="", \$id="", \$user)

**Input parameters:**

\$site ... publication name (optional)

\$location ... location (optional)

\$page ... object name (optional)

\$id ... identifier (object ID

\$user ... object hash) (optional)

**global input parameters:**

\$mgmt\_config

**Output:**

true / false

### 7.1.145 getfavorites

**Syntax:**

getfavorites (\$user, \$output="path")

**Input parameters:**

\$user ... user name

\$output ... output [path, id] (optional)

**global input parameters:**

\$mgmt\_config

**Output:**

object path or id array of users favorites / false

### 7.1.146 deletefavorite

**Syntax:**

deletefavorite (\$site="", \$location="", \$page="", \$id="", \$user)

**Input parameters:**

\$site ... publication name (optional)

\$location ... location (optional)

\$page ... object name (optional)

\$id ... identifier (object ID)

\$user ... object hash) (optional)

**global input parameters:**

\$mgmt\_config

**Output:**

true / false

### 7.1.147 setboxes

**Syntax:**

setboxes (\$name\_array, \$user)

**Input parameters:**

\$name\_array ... home box names as array or string

\$user ... user name

**global input parameters:**

\$mgmt\_config

**Output:**

true / false

### 7.1.148 getboxes

**Syntax:**

getboxes (\$user)

**Input parameters:**

\$user ... user name

**global input parameters:**

\$mgmt\_config

**Output:**

selected home box names of user as array / false

### 7.1.149      rewrite\_targetURI

**Syntax:**

rewrite\_targetURI (\$site, \$text\_id, \$uri, \$exclude\_dir\_esc="", \$rewrite\_type="include")

**Input parameters:**

\$site ... publication name

\$text\_id ... text ID array (text-ID as key and URL paramaters as value)

\$uri ... requested URI as string

\$exclude\_dir\_esc ... exclude path as array (optional)

\$rewrite\_type ... rewrite type [none, forward, include] (optional)

**global input parameters:**

\$mgmt\_config

\$publ\_config

**Output:**

target URI / false on error

### 7.1.150      rewrite\_homepage

**Syntax:**

rewrite\_homepage (\$site, \$rewrite\_type="forward")

**Input parameters:**

\$site ... publication name

\$rewrite\_type ... rewrite type [none, forward] (optional)

**global input parameters:**

\$mgmt\_config

\$publ\_config

**Output:**

target URI / false on error

**Description:**

Uses the page root directory of the publication configuration and forwards to the default index page. No page include supported!

### 7.1.151      sendmessage

**Syntax:**

sendmessage (\$from\_user, \$to\_user, \$title, \$message, \$object\_id="")

**Input parameters:**

\$from\_user ... from\_user name

\$to\_user ... to\_user name

\$title ... title

\$message ... message

\$object\_id ... object ID or object path (optional)

**global input parameters:**

\$mgmt\_config

\$hcms\_lang\_codepage  
\$hcms\_lang  
\$lang

**Output:**

true/false

**Description:**

Sends a message via e-mail to a user

## 7.2 Get API Functions

### 7.2.1 getserverload

**Syntax:**

getserverload ()

**Input parameters:**

**Output:**

Returns the average system load (the number of processes in the system run queue) over the last minute and the number of CPU cores as array

### 7.2.2 getsession

**Syntax:**

getsession (\$variable, \$default="")

**Input parameters:**

\$variable ... session variable name

\$default ... default session value (optional)

**Output:**

session value

### 7.2.3 getrequest

**Syntax:**

getrequest (\$variable, \$force\_type=false, \$default="")

**Input parameters:**

\$variable ... request variable name

\$force\_type ... must be of certain type [numeric, array, publicationname, locationname, objectname, url, bool] (optional)

\$default ... default value (optional)

**Output:**

request value

**Description:**

Returns the value of a POST, GET or COOKIE request, or a default value if not valid.

### 7.2.4 getrequest\_esc

**Syntax:**

getrequest\_esc (\$variable, \$force\_type=false, \$default="", \$js\_protection=false)

**Input parameters:**

\$variable ... request variable name

\$force\_type ... must be of certain type [numeric, array, publicationname, locationname, objectname] (optional)

\$default ... default value (optional)

\$js\_protection ... remove characters to avoid JS injection [true, false] (optional)

**Output:**

request value

**Description:**

Returns the escaped value in order to prevent XSS from POST, GET or COOKIE variables.  
Returns a default value if not valid.

### 7.2.5 getuserip

**Syntax:**

getuserip ()

**Input parameters:****Output:**

IP address of client / false on error

**Description:**

Retrieves the IP address of the client/user.

### 7.2.6 getlanguageoptions

**Syntax:**

getlanguageoptions ()

**Input parameters:****global input parameters:**

\$mgmt\_config

**Output:**

array with 2-digit language code as key and language name in English as value / false on error

### 7.2.7 getlanguagefile

**Syntax:**

getlanguagefile (\$lang="en")

**Input parameters:**

\$lang ... language code (optional)

**global input parameters:**

\$mgmt\_config

**Output:**

language file name



### 7.2.8 getcodepage

**Syntax:**

getcodepage (\$lang="en")

**Input parameters:**

\$lang ... language code (optional)

**global input parameters:**

\$mgmt\_config

\$hcms\_lang\_codepage

**Output:**

code page (character set)

### 7.2.9 getcalendarlang

**Syntax:**

getcalendarlang (\$lang="en")

**Input parameters:**

\$lang ... language code (optional)

**global input parameters:**

\$mgmt\_config

**Output:**

supported language code for calendar

### 7.2.10 getescapedtext

**Syntax:**

getescapedtext (\$text, \$charset="", \$lang="")

**Input parameters:**

\$text ... word as string

\$charset ... 2-digit language code (optional)

\$lang

**global input parameters:**

\$mgmt\_config

**Output:**

array holding all synonyms including the provided word / false on error

**Description:**

Returns the synonyms of a word.

### 7.2.11 getsearchhistory

**Syntax:**

getsearchhistory ()

**Input parameters:**

**global input parameters:**

\$mgmt\_config

**Output:**

array holding all expressions of the search history / false on error

## 7.2.12      gettaxonomy\_sublevel

**Syntax:**

gettaxonomy\_sublevel (\$site, \$lang="en", \$tax\_id="0")

**Input parameters:**

\$site ... publication name

\$lang ... language code (optional)

\$tax\_id ... taxonomy parent ID (optional)

**global input parameters:**

\$mgmt\_config

\$mgmt\_lang\_shortcut\_default

\$taxonomy

**Output:**

array holding all keywords of the next taxonomy level / false on error

**Description:**

Returns sorted keywords of a taxonomy level (multilingual support based on taxonomies).  
Global variable \$taxonomy can be used to pass the taxonomy as array.

## 7.2.13      gettaxonomy\_childs

**Syntax:**

gettaxonomy\_childs (\$site="", \$lang="", \$expression, \$childlevels=1, \$id\_only=true)

**Input parameters:**

\$site ... publication name (optional)

\$lang ... language code (optional)

\$expression ... taxonomy ID or expression or taxonomy path in the form

%taxonomy%/publication-name or 'default'/language-code/taxonomy-ID/taxonomy-child-levels as string

\$childlevels ... taxonomy child levels as integer (optional)

\$id\_only ... only return taxonomy IDs without language and keyword information [true, false] (optional)

**global input parameters:**

\$mgmt\_config

\$mgmt\_lang\_shortcut\_default

\$taxonomy

**Output:**

array holding all taxonomy IDs / false on error

**Description:**

Returns keywords based on taxonomy definition and synonyms if expression is a keyword (multilingual support based on taxonomies and synonyms).

The expression can be a taxonomy path in the form of %taxonomy%/site/language-code/taxonomy-ID/taxonomy-child-levels (use "all" for all languages and "0" for all taxonomy-IDs on first level).

Global variable \$taxonomy can be used to pass the taxonomy as array.

### 7.2.14 getkeywords

**Syntax:**

getkeywords (\$text, \$language="en", \$charset="UTF-8")

**Input parameters:**

\$text ... text as string

\$language ... language to be used for stop word list [de, en, ...] (optional)

\$charset ... character set (optional)

**global input parameters:**

\$mgmt\_config

**Output:**

keywords as array /false on error

**Description:**

Generates a keyword list from a plain text. Stop word lists are defined in data/include/stopwords.inc.php

### 7.2.15 getdescription

**Syntax:**

getdescription (\$text, \$charset="UTF-8")

**Input parameters:**

\$text ... text as string

\$charset

**Output:**

cleanded description of provided text /false on error

**Description:**

Generates a description from a text, to be used as meta information.

### 7.2.16 getgooglesitemap

**Syntax:**

getgooglesitemap (\$site, \$dir, \$url, \$getpara=array(), \$permlink=array(),  
\$chfreq="weekly", \$prio="", \$ignore=array(), \$filetypes=array('cfm', 'htm', 'html', 'xhtml',  
'asp', 'aspx', 'jsp', 'php', 'pdf'), \$show\_freq=true, \$show\_prio=true)

**Input parameters:**

\$site ... publication anme

\$dir ... directory path

\$url ... URL to directory

\$getpara ... GET parameters to use for new versions of the URL as array (optional)

\$permlink ... permanent links text-ID to use for location as array (optional)

\$chfreq ... frequency of google scrawler [never, weekly, daily] (optional)

\$prio ... priority [1 or less] (optional)

\$ignore ... ignore file names as array (optional)

\$filetypes ... allowed file types as array (optional)

\$show\_freq ... include frequency tag [true, false] (optional)

\$show\_prio ... include priority tag [true, false] (optional)

**global input parameters:**

\$mgmt\_config

\$publ\_config

**Output:**

xml sitemap / false on error

**Description:**

Generates a google sitemap xml-output

## 7.2.17 getmetadata

**Syntax:**

getmetadata (\$location, \$object, \$container="", \$separator="\n", \$template="")

**Input parameters:**

\$location ... location

\$object ... object (both optional if container is given)

\$container ... container name or container content (optional)

\$separator ... separator of meta data fields [any string, array] (optional)

\$template ... publication name/template name to extract label names (optional)

**global input parameters:**

\$mgmt\_config

**Output:**

string with all meta data from given object based on container / false

## 7.2.18 getobjectcontainer

**Syntax:**

getobjectcontainer (\$site, \$location, \$object, \$user)

**Input parameters:**

\$site ... publication [string]

\$location ... location [string]

\$object ... object [string]

\$user ... user [string]

**global input parameters:**

\$mgmt\_config

**Output:**

Content Container [XML]/false

**Description:**

Loads the content container of a given object (page, component, folder)

## 7.2.19 getcontainer

**Syntax:**

getcontainer (\$containerid, \$type)

**Input parameters:**

\$containerid ... container name or container ID

\$type ... container type [published]

**global input parameters:**

\$mgmt\_config

**Output:**

Contant Container [XML]/false

**Description:**

Obsolete function used as an alias for the loadcontainer function without the possibility to load locked containers

## 7.2.20      getcontainername

**Syntax:**

getcontainername (\$container)

**Input parameters:**

\$container ... container name (e.g. 0000112.xml.wrk) or container ID

**global input parameters:**

\$mgmt\_config

**Output:**

Array with file name of the working content container (locked or unlocked!) and username if locked

## 7.2.21      getlocationname

**Syntax:**

getlocationname (\$site, \$location, \$cat, \$source="path")

**Input parameters:**

\$site ... publication name

\$location ... location path (as absolute path or converted path)

\$cat ... category [page, comp]

\$source ... source for name [path, name]

**global input parameters:**

\$mgmt\_config

\$lang

\$hcms\_lang\_codepage

**Output:**

location with readable names instead of file names / false on error

## 7.2.22      getthemelocation

**Syntax:**

getthemelocation (\$theme="")

**Input parameters:**

\$theme ... theme name (optional)

**global input parameters:**

\$mgmt\_config

**Output:**

path to theme / false

**Description:**

Returns the absolute path (URL) of the theme (css and images).

### 7.2.23 getcategory

**Syntax:**

getcategory (\$site="", \$location)

**Input parameters:**

\$site ... publication name (optional)

\$location ... location path

**global input parameters:**

\$mgmt\_config

\$publ\_config

**Output:**

category ['page  
comp'] / false on error

**Description:**

Evaluates the category ['page, comp'] of a location

### 7.2.24 getpublication

**Syntax:**

getpublication (\$path)

**Input parameters:**

\$path ... converted location path

**Output:**

publication name

**Description:**

Extract the publication name of a location path

### 7.2.25 getlocation

**Syntax:**

getlocation (\$path)

**Input parameters:**

\$path ... location path

**Output:**

location (without object or folder)

**Description:**

Extract the location excluding object or folder of a location path

### 7.2.26 getobject

**Syntax:**

getobject (\$path)

**Input parameters:**

\$path ... location path

**Output:**

object or folder name

**Description:**

Extract the object or folder of a location path

## 7.2.27 getmediacontainername

**Syntax:**

getmediacontainername (\$file)

**Input parameters:**

\$file ... file name

**Output:**

container name / false on error

**Description:**

Extract the container name from a multimedia file name by using the hcm-ID

## 7.2.28 getmediafileversion

**Syntax:**

getmediafileversion (\$container)

**Input parameters:**

\$container ... container name or container ID

**global input parameters:**

\$mgmt\_config

\$user

**Output:**

media file name / false on error

**Description:**

Extracts the name from the multimedia file by container name or ID in order to get the media file of older content versions.

if the result is false, there is no older media file version.

## 7.2.29 getobjectid

**Syntax:**

getobjectid (\$objectlink)

**Input parameters:**

\$objectlink ... converted object path or pathes separated by |

**Output:**

object ID

**Description:**

Converts the object path to the object ID of any object

## 7.2.30 getobjectlink

**Syntax:**

getobjectlink (\$objectid)

**Input parameters:**

\$objectid ... converted object ID or IDs separated by |

**Output:**

converted object link

**Description:**

Converts the object ID to the object path of any object

### 7.2.31      getcontainerversions

**Syntax:**

getcontainerversions (\$container)

**Input parameters:**

\$container ... container ID or container name

**global input parameters:**

\$mgmt\_config

**Output:**

array of all versions (array[version-extension] = file-name) / false

### 7.2.32      gettemplateversions

**Syntax:**

gettemplateversions (\$site, \$template)

**Input parameters:**

\$site ... publication name

\$template ... template name

**global input parameters:**

\$mgmt\_config

**Output:**

array of all versions (array['YYYY-MM-DD HH:MM:SS'] = file-name) / false

### 7.2.33      getfileinfo

**Syntax:**

getfileinfo (\$site, \$file, \$cat="comp")

**Input parameters:**

\$site ... publication name (optional)

\$file ... file name incl. extension

\$cat ... category [page, comp] (optional)

**global input parameters:**

\$mgmt\_config

**Output:**

array/false

**Description:**

defines file properties based on the file extension and returns file info as an array:

\$result['file']: file name without hypercms management extension



\$result['name']: readable file name without hypercms management extension  
\$result['filename']: file name without file extensions  
\$result['icon']: file name of the file icon  
\$result['icon\_large']: file name of the large file icon  
\$result['type']: file type  
\$result['ext']: file extension incl. dot in lower case  
\$result['published']: if file published = true else = false

### 7.2.34      getobjectinfo

**Syntax:**

getobjectinfo (\$site, \$location, \$object, \$user="sys", \$container\_version="")

**Input parameters:**

\$site ... publication name  
\$location ... location  
\$object ... object name  
\$user ... user name (optional)  
\$container\_version ... container version (optional)

**global input parameters:**

\$mgmt\_config

**Output:**

result array / false on error

**Description:**

Get all file pointers (container, media, template) and object name from object file and collect info from container version, if provided

### 7.2.35      getfilesize

**Syntax:**

getfilesize (\$file)

**Input parameters:**

\$file ... converted path to file or directory

**global input parameters:**

\$mgmt\_config

**Output:**

result array with file size in kB and file count / false on error

### 7.2.36      getmimetype

**Syntax:**

getmimetype (\$file)

**Input parameters:**

\$file ... file name incl. extension

**global input parameters:**

\$mgmt\_config

**Output:**

mime\_type

**Description:**

Gets the mime-type of the file by its extension.

If file has a version file extension the next file extension will be used.

### 7.2.37      getfiletype

**Syntax:**

getfiletype (\$file\_ext)

**Input parameters:**

\$file\_ext ... file extension or file name

**global input parameters:**

\$mgmt\_config

\$hcms\_ext

**Output:**

file type to be saved in database based on file extension

### 7.2.38      getvideoinfo

**Syntax:**

getvideoinfo (\$mediafile)

**Input parameters:**

\$mediafile ... path to video file

**global input parameters:**

\$mgmt\_config

\$mgmt\_mediapreview

\$user

**Output:**

video file information as result array / false on error

### 7.2.39      getbrowserinfo

**Syntax:**

getbrowserinfo ()

**Input parameters:****Output:**

client browser + version as array

### 7.2.40      getcontentlocation

**Syntax:**

getcontentlocation (\$container\_id, \$type="abs\_path\_content")

**Input parameters:**

\$container\_id ... container id

\$type ... type [url\_path\_content]

**global input parameters:**

\$mgmt\_config

**Output:**

location of the container file / false on error

**Description:**

Gets the content location based on the given container id.

The use of various directories is necessary since the number of directories is limited by the filesystem, e.g. Linux ext3 is limited to 32000.

## 7.2.41 getmedialocation

**Syntax:**

getmedialocation (\$site, \$file, \$type)

**Input parameters:**

\$site ... publication name

\$file ... multimedia file name (including hcm-ID)

\$type ... type [url\_path\_media

**global input parameters:**

\$mgmt\_config

\$publ\_config

**Output:**

location of the multimedia file / false on error

**Description:**

Gets the media repository location from \$mgmtm\_config array.

The function supports up to 10 media repositories.

Any other rules for splitting the media files on multiple devices can be implemented as well by the function getmedialocation\_rule.

## 7.2.42 getlockedfileinfo

**Syntax:**

getlockedfileinfo (\$location, \$file)

**Input parameters:**

\$location ... location to file

\$file ... file name

**global input parameters:**

\$mgmt\_config

**Output:**

Array holding file name incl. lock extension and user name / false on error

**Description:**

Finds the locked file and returns the name and user as array

## 7.2.43 getuseronline

**Syntax:**

getuseronline ()

**Input parameters:****global input parameters:**

\$mgmt\_config

**Output:**

Array of online user names / false

#### 7.2.44      getchatstate

**Syntax:**

getchatstate (\$register=true)

**Input parameters:**

\$register ... register stat in session [true/false] (optional)

**global input parameters:**

\$mgmt\_config

**Output:**

state of chat / false on error

#### 7.2.45      getimagelib

**Syntax:**

getimagelib ()

**Input parameters:****global input parameters:**

\$mgmt\_imagepreview

**Output:**

name of image library used [GD  
ImageMagick] / false on error

#### 7.2.46      getfilename

**Syntax:**

getfilename (\$filedata, \$tagname)

**Input parameters:**

\$filedata ... file content

\$tagname ... hyperCMS tag name in page or component

**Output:**

file name

**Description:**

Extracts the file name of the content and template pointer tags of an object file

#### 7.2.47      gethypertag

**Syntax:**

gethypertag (\$filedata, \$tagname, \$offset=0)

**Input parameters:**

\$filedata ... file content [string]

\$tagname ... full/partly hyperCMS tag name (with or without hyperCMS:) [string]

\$offset ... offset position [integer]

**Output:**

full hyperCMS tag array [array]/false on error

**Description:**

Finds the hyperCMS tag start and end position and returns an array of the whole tags including all information.

Offset value must be integer value and is used to skip search for hyperCMS tag till offset position of filedata.

#### 7.2.48      gethypertagname

**Syntax:**

gethypertagname (\$tagdata)

**Input parameters:**

\$tagdata ... full hyperCMS tag

**Output:**

full hyperCMS tag name/false on error

**Description:**

Reads the name of the hyperCMS tag

#### 7.2.49      gethtmltag

**Syntax:**

gethtmltag (\$filedata, \$tag)

**Input parameters:**

\$filedata ... file content

\$tag ... full hyperCMS tag (or other identifier)

**Output:**

full html tag/false on error

**Description:**

Finds the first html tag start and end position of a nested hyperCMS tag and returns the whole tag including all information.

Works also if other script tags are nested in the HTML-tag.

This function is not case sensitive!

#### 7.2.50      gethtmltags

**Syntax:**

gethtmltags (\$filedata, \$tag)

**Input parameters:**

\$filedata ... file content

\$tag ... full hyperCMS tag or other identifier in html tag

**Output:**

string from html tag start to end tag/false on error

**Description:**

Finds the nearest html tag start and end position of a nested hyperCMS tag and returns the whole tag including all information.

This functions works also for html-tag pairs like <a href></a>, <div></div> and so on.

### 7.2.51      getattribute

**Syntax:**

getattribute (\$string, \$attribute, \$secure=true)

**Input parameters:**

\$string ... string including attributes

\$attribute ... attribute name

\$secure ... secure attribute value reg. XSS (optional)

**Output:**

attribute value/false on error

**Description:**

Get the value of a certain attribute out of a string (...attributname=value....)

### 7.2.52      getoption

**Syntax:**

getoption (\$string, \$option)

**Input parameters:**

\$string ... string including options

\$option ... option name

**Output:**

option value / false on error

**Description:**

Get the value of a certain option out of a string (-c:v value -ar 44100)

### 7.2.53      getcharset

**Syntax:**

getcharset (\$site, \$data)

**Input parameters:**

\$site ... publication

\$data ... data from template or content container [string]

**global input parameters:**

\$mgmt\_config

**Output:**

array with content-type and charset / false on error

**Description:**

Extract the content-type definition and the character set from the template (1st priority), content container (2nd priority) or publication settings (3rd priority)

### 7.2.54      getartid

**Syntax:**

getartid (\$id)

**Input parameters:**

\$id ... string including id

**Output:**

article id/false on error

**Description:**

Extract the article ID from the tag ID

## 7.2.55      getelementid

**Syntax:**

getelementid (\$id)

**Input parameters:**

\$id ... string including id

**Output:**

element id/false on error

**Description:**

Extract the element ID from the tag ID

## 7.2.56      getfirstkey

**Syntax:**

getfirstkey (\$array)

**Input parameters:**

\$array ... array

**Output:**

array key of first element in array if \$value is not empty / false on error

## 7.2.57      getdirectoryfiles

**Syntax:**

getdirectoryfiles (\$dir, \$pattern="")

**Input parameters:**

\$dir ... path to directory

\$pattern ... pattern as string (optional)

**Output:**

sorted array of all files matching the pattern / false on error

## 7.2.58      getuserinformation

**Syntax:**

getuserinformation ()

**Input parameters:****global input parameters:**

\$mgmt\_config

**Output:**

associative array with basic user information [publication->username->attribute] / false

**Description:**

This function creates an associative array with user information for the user select box

## 7.3 Set API Functions

### 7.3.1 setsession

**Syntax:**

```
setsession ($variable, $content="", $write=false)
```

**Input parameters:**

\$variable ... temporary hyperCMS variable name or array

\$content ... value as string or array (optional)

\$write ... write session data for load balancer [true, false] (optional)

**Output:**

true / false on error

### 7.3.2 settaxonomy

**Syntax:**

```
settaxonomy ($site, $container_id, $text_array, $langcode="")
```

**Input parameters:**

\$site ... publication name

\$container\_id ... container ID

\$text\_array ... content as array in form of array[text-ID]=text-content

\$langcode ... 2-digit language code (optional)

**global input parameters:**

\$mgmt\_config

\$taxonomy

**Output:**

result array / false on error

**Description:**

Analyzes the content regarding all taxonomy keywords and returns an array (multilingual support based on taxonomies).

Global variable \$taxonomy can be used to pass the taxonomy as array.

### 7.3.3 setarticle

**Syntax:**

```
setarticle ($site, $contentdata, $contentfile, $arttitle, $artstatus, $artdatefrom, $artdateto, $artuser, $user)
```

**Input parameters:**

\$site ... publication name

\$contentdata ... container (XML)

\$contentfile ... container name

\$arttitle ... article title array

\$artstatus ... article status array

\$artdatefrom ... article beginn date array

\$artdateto ... article end date array

\$artuser ... user array or string

\$user ... user name



**global input parameters:**

\$mgmt\_config

**Output:**

updated content container (XML)

false on error

### 7.3.4 settext

**Syntax:**

settext (\$site, \$contentdata, \$contentfile, \$text, \$type, \$art, \$textuser, \$user, \$charset="", \$addmicrotime=false)

**Input parameters:**

\$site ... publication name

\$contentdata ... container (XML)

\$contentfile ... container name

\$text ... text array

\$type ... type array or string of text [u, f, l, c, d]

\$art ... article array or string [yes, no]

\$textuser ... text user array or string

\$user ... user name

\$charset ... character set of text content

\$addmicrotime ... add microtime to ID [true, false] used for comments

**global input parameters:**

\$mgmt\_config

\$publ\_config

**Output:**

updated content container (XML)

false on error

### 7.3.5 setmedia

**Syntax:**

setmedia (\$site, \$contentdata, \$contentfile, \$mediafile, \$mediaobject\_curr, \$mediaobject, \$mediaalltext, \$mediaalign, \$mediawidth, \$mediaheight, \$art, \$mediauser, \$user, \$charset="")

**Input parameters:**

\$site ... publication name

\$contentdata ... container (XML)

\$contentfile ... container name

\$mediafile ... media arrays (some are optional)

\$mediaobject\_curr ... article array or string [yes, no]

\$mediaobject ... content user array or string

\$mediaalltext ... user name

\$mediaalign ... chracter set of text content

\$mediawidth

\$mediaheight

\$art

\$mediauser

\$user

\$charset

**global input parameters:**

\$mgmt\_config

**Output:**

updated content container (XML)  
false on error

### 7.3.6 setpagelink

**Syntax:**

setpagelink (\$site, \$contentdata, \$contentfile, \$linkhref\_curr, \$linkhref, \$linktarget, \$linktext, \$art, \$linkuser, \$user, \$charset="")

**Input parameters:**

\$site ... publication name  
\$contentdata ... container (XML)  
\$contentfile ... container name  
\$linkhref\_curr ... current link array  
\$linkhref ... new link array  
\$linktarget ... link target array  
\$linktext ... link text array  
\$art ... article array or string [yes, no]  
\$linkuser ... content user array or string  
\$user ... user name  
\$charset ... chracter set of text content

**global input parameters:**

\$mgmt\_config

**Output:**

updated content container (XML)  
false on error

### 7.3.7 setcomplink

**Syntax:**

setcomplink (\$site, \$contentdata, \$contentfile, \$component\_curr, \$component, \$condition, \$art, \$compuser, \$user)

**Input parameters:**

\$site ... publication name  
\$contentdata ... container (XML)  
\$contentfile ... container name  
\$component\_curr ... component arrays (some are optional)  
\$component ... article array or string [yes, no]  
\$condition ... content user array or string  
\$art ... user name  
\$compuser  
\$user

**global input parameters:**

\$mgmt\_config

**Output:**

updated content container (XML)  
false on error

### 7.3.8 sethead

**Syntax:**

sethead (\$site, \$contentdata, \$contentfile, \$headcontent, \$user, \$charset="")

**Input parameters:**

\$site ... publication name

\$contentdata ... container (XML)

\$contentfile ... container name

\$headcontent ... content array

\$user ... user name

\$charset ... chracter set of text content

**global input parameters:**

\$mgmt\_config

**Output:**

updated content container (XML)

false on error

**Description:**

Only used for content in general head information of container

### 7.3.9 setfilename

**Syntax:**

setfilename (\$filedata, \$tagname, \$value)

**Input parameters:**

\$filedata ... file content

\$tagname ... hyperCMS tag name in page or component [content

\$value ... template

**Output:**

filedata/false on error

**Description:**

Sets or creates the file name of the hyperCMS content file, template file, media file or file name pointer

## 7.4 Connect API Functions

### 7.4.1 ftp\_userlogon

**Syntax:**

ftp\_userlogon (\$server, \$user, \$passwd, \$ssl=false)

**Input parameters:**

\$server ... FTP servername or IP

\$user ... user name

\$passwd ... password

\$ssl ... SSL [true, false] (optional)

**global input parameters:**

\$mgmt\_config

**Output:**

true / false on error

**Description:**

This function connects and performs logon to an FTP server

### 7.4.2 ftp\_userlogout

**Syntax:**

ftp\_userlogout (\$conn\_id)

**Input parameters:**

\$conn\_id ... FTP connection

**global input parameters:**

\$mgmt\_config

**Output:**

true / false on error

**Description:**

This function disconnects from an FTP server

### 7.4.3 ftp\_getfile

**Syntax:**

ftp\_getfile (\$conn\_id, \$remote\_file, \$local\_file, \$passive=true)

**Input parameters:**

\$conn\_id ... FTP connection

\$remote\_file ... path to file on FTP server

\$local\_file ... passive mode [true, false] (optional)

\$passive

**global input parameters:**

\$mgmt\_config

**Output:**

true / false on error

**Description:**

This function gets a file from the FTP server

### 7.4.4 ftp\_putfile

**Syntax:**

ftp\_putfile (\$conn\_id, \$local\_file, \$remote\_file, \$passive=true)

**Input parameters:**

\$conn\_id ... FTP connection

\$local\_file ... path to local file

\$remote\_file ... path to file on FTP server

\$passive ... passive mode [true, false] (optional)

**global input parameters:**

\$mgmt\_config

**Output:**

true / false on error

**Description:**

This function puts a file to the FTP server

#### 7.4.5 ftp\_filelist

**Syntax:**

ftp\_filelist (\$conn\_id, \$path=".", \$passive=true)

**Input parameters:**

\$conn\_id ... FTP connection

\$path ... path to remote directory (optional)

\$passive ... passive mode [true, false] (optional)

**global input parameters:**

\$mgmt\_config

**Output:**

result array / false on error

**Description:**

This function gets a file/directory listing of the FTP server

#### 7.4.6 createsharelink\_facebook

**Syntax:**

createsharelink\_facebook (\$site, \$url)

**Input parameters:**

\$site ... URL to share

\$url

**global input parameters:**

\$mgmt\_config

**Output:**

Share URL / false on error

#### 7.4.7 createsharelink\_twitter

**Syntax:**

createsharelink\_twitter (\$site, \$url, \$text)

**Input parameters:**

\$site ... URL to share

\$url ... message to share

\$text

**global input parameters:**

\$mgmt\_config

**Output:**

Share URL / false on error

#### 7.4.8 createsharelink\_googleplus

**Syntax:**

createsharelink\_googleplus (\$site, \$url)

**Input parameters:**

\$site ... URL to share  
\$url

**global input parameters:**

\$mgmt\_config

**Output:**

Share URL / false on error

#### 7.4.9 createsharelink\_linkedin

**Syntax:**

createsharelink\_linkedin (\$site, \$url, \$title, \$summary, \$source)

**Input parameters:**

\$site ... URL to share  
\$url ... title  
\$title ... summary (optional)  
\$summary ... source (optional)  
\$source

**global input parameters:**

\$mgmt\_config

**Output:**

Share URL / false on error

#### 7.4.10 createsharelink\_pinterest

**Syntax:**

createsharelink\_pinterest (\$site, \$image\_url, \$title, \$description)

**Input parameters:**

\$site ... image URL to share  
\$image\_url ... title  
\$title ... description (optional)  
\$description

**global input parameters:**

\$mgmt\_config

**Output:**

Share URL / false on error

## 7.5 Security API Functions

### 7.5.1 rootpermission

**Syntax:**

rootpermission (\$site\_name, \$site\_admin, \$permission\_str)

**Input parameters:**

\$site\_name ... publication name  
\$site\_admin ... publication admin

\$permission\_str ... permission string from group

**global input parameters:**

\$rootpermission

\$mgmt\_config

**Output:**

global permission array/false

**Description:**

Deserializes the permission string and returns the root permission array

## 7.5.2 globalpermission

**Syntax:**

globalpermission (\$site\_name, \$permission\_str)

**Input parameters:**

\$site\_name ... publication

\$permission\_str ... permission string from group

**Output:**

global permission array/false

**Description:**

Deserializes the permission string and returns the global permission array

## 7.5.3 localpermission

**Syntax:**

localpermission (\$site\_name, \$permission\_str)

**Input parameters:**

\$site\_name ... publication

\$permission\_str ... permission string from group

**Output:**

local permission array/false

**Description:**

Deserializes the permission string and returns the local permission array

## 7.5.4 accessgeneral

**Syntax:**

accessgeneral (\$site, \$location, \$cat)

**Input parameters:**

\$site ... publication

\$location ... location (path to folder)

\$cat ... object category ['page

**global input parameters:**

\$mgmt\_config

\$hiddenfolder

\$siteaccess

**Output:**

true/false

**Description:**

Checks general access to certain system folders, publications and returns true if access is granted

### 7.5.5 accesspermission

**Syntax:**

accesspermission (\$site, \$location, \$cat)

**Input parameters:**

\$site ... location (path to folder)

\$location ... object category ['page

\$cat ... comp']

**global input parameters:**

\$pageaccess

\$compaccess

\$hiddenfolder

\$hcms\_linking

\$mgmt\_config

**Output:**

group with access permissions as array / false on error

**Description:**

Evaluates page and component access permissions and returns group(s)

### 7.5.6 setlocalpermission

**Syntax:**

setlocalpermission (\$site, \$group\_array, \$cat)

**Input parameters:**

\$site ... publication

\$group\_array ... group name array

\$cat ... object category [page, comp]

**global input parameters:**

\$localpermission

**Output:**

local permission array / false on error

**Description:**

Sets local permissions of a user group for a specific publication

### 7.5.7 checkpublicationpermission

**Syntax:**

checkpublicationpermission (\$site, \$strict=true)

**Input parameters:**

\$site ... publication name

\$strict ... strictly limited to siteaccess only without inheritance [true/false] (optional)

**global input parameters:**



\$mgmt\_config  
\$siteaccess

**Output:**

"direct" for direct access via group permission / "inherited" for access through inheritance / false

**Description:**

Checks access to a publication based on the site access and inheritance settings

## 7.5.8 checkadminpermission

**Syntax:**

checkadminpermission ()

**Input parameters:**

**global input parameters:**

\$adminpermission

**Output:**

true/false

**Description:**

Checks super admin permission

## 7.5.9 checkrootpermission

**Syntax:**

checkrootpermission (\$name)

**Input parameters:**

\$name ... permission name

**global input parameters:**

\$rootpermission

**Output:**

true/false

**Description:**

Checks root permission

## 7.5.10 checkglobalpermission

**Syntax:**

checkglobalpermission (\$site, \$name)

**Input parameters:**

\$site ... publication name

\$name ... permission name

**global input parameters:**

\$globalpermission

**Output:**

true/false

**Description:**

Checks global permission for a publication

### 7.5.11      checklocalpermission

**Syntax:**

checklocalpermission (\$site, \$group, \$name)

**Input parameters:**

\$site ... publication name

\$group ... user group name

\$name ... permission name

**global input parameters:**

\$localpermission

**Output:**

true/false

**Description:**

Checks local permissions of a user group for a specific publication

### 7.5.12      userlogin

**Syntax:**

userlogin (\$user, \$passwd, \$hash="", \$objref="", \$objcode="", \$ignore\_password=false, \$locking=true)

**Input parameters:**

\$user ... username

\$passwd ... password

\$hash ... hash code of user

\$objref ... object reference for hcms linking (object ID)

\$objcode ... object code for hcms linking (crypted object ID)

\$ignore\_password ... ignore passwordcheck needed for WebDAV or access link [true/false]

\$locking ... lock IP after 10 failed attempts to login [true/false]

**global input parameters:**

\$mgmt\_config

\$eventsystem

\$hcms\_lang\_codepage

\$hcms\_lang

\$lang

**Output:**

result array

**Description:**

Login of user by sending user and password using the variables: \$sentuser, \$sentpasswd

This procedure will register the user in the hypercms session and in the php session.

The procedure will return true or false using the variable \$result.

### 7.5.13      registerinstance

**Syntax:**

registerinstance (\$instance, \$load\_config=true)

**Input parameters:**

\$instance ... instance name  
\$load\_config ... load main config of instance [true/false] (optional)

**global input parameters:**

\$mgmt\_config

**Output:**

true/false

#### 7.5.14 createchecksum

**Syntax:**

createchecksum (\$permissions="")

**Input parameters:**

\$permissions ... array or empty

**Output:**

MD5 checksum

#### 7.5.15 writesession

**Syntax:**

writesession (\$user, \$passwd, \$checksum)

**Input parameters:**

\$user ... user name

\$passwd ... password

\$checksum ... checksum

**global input parameters:**

\$mgmt\_config

**Output:**

true / false on error

**Description:**

Writes hyperCMS specific session data of a user

#### 7.5.16 writesessiondata

**Syntax:**

writesessiondata ()

**Input parameters:**

**global input parameters:**

\$mgmt\_config

**Output:**

true / false on error

**Description:**

Serializes and writes all session data of a user

### 7.5.17 createsession

**Syntax:**

createsession ()

**Input parameters:****global input parameters:**

\$mgmt\_config

**Output:**

true

**Description:**

Checks if session data of a user is available. This function does access session variables directly!

### 7.5.18 killsession

**Syntax:**

killsession (\$user="", \$destroy\_php=true)

**Input parameters:**

\$user ... user name for hyperCMS session (optional)

\$destroy\_php ... destroy php session [true, false] (optional)

**global input parameters:**

\$mgmt\_config

**Output:**

true

**Description:**

Destroys session data of user

### 7.5.19 checkdiskkey

**Syntax:**

checkdiskkey (\$users="", \$site="")

**Input parameters:**

\$users ... user count (optional)

\$site ... publication names (use | as separator) (optional)

**global input parameters:**

\$mgmt\_config

**Output:**

true/false

**Description:**

Checks the disc key of the installation

### 7.5.20 checkpassword

**Syntax:**

checkpassword (\$password)

**Input parameters:**

\$password ... password a string

**global input parameters:**

\$mgmt\_config

\$lang

**Output:**

true if passed / error message as string

**Description:**

This function checks the strength of a password and return the error messages or true

## 7.5.21 loguserip

**Syntax:**

loguserip (\$client\_ip, \$user="sys")

**Input parameters:**

\$client\_ip ... client IP address

\$user ... user logon name (optional)

**global input parameters:**

\$mgmt\_config

**Output:**

true / false on error

## 7.5.22 checkuserip

**Syntax:**

checkuserip (\$client\_ip, \$user="", \$timeout="")

**Input parameters:**

\$client\_ip ... client IP address

\$user ... user logon name (optional)

\$timeout ... timeout in minutes (optional)

**global input parameters:**

\$mgmt\_config

**Output:**

true if IP is not locked / false if IP is locked or on error

## 7.5.23 checkuserrequests

**Syntax:**

checkuserrequests (\$user="sys")

**Input parameters:**

\$user ... user name (optional)

**global input parameters:**

\$mgmt\_config

**Output:**

true / false if a certain amount of requests per minute is exceeded

**Description:**

Provides security for Cross-Site Request Forgery

### 7.5.24      checkusersession

**Syntax:**

checkusersession (\$user="sys", \$CSRF\_detection=true)

**Input parameters:**

\$user ... user name (optional)

\$CSRF\_detection ... include CSRF detection [true, false]

**global input parameters:**

\$mgmt\_config

**Output:**

true / html-output followed by termination requires config.inc.php

**Description:**

Checks if session data of user is correct. This function does access session variables directly!

### 7.5.25      allowuserip

**Syntax:**

allowuserip (\$site)

**Input parameters:**

\$site ... publication name

**global input parameters:**

\$mgmt\_config

**Output:**

true / false requires config.inc.php

**Description:**

Checks if the client IP is in the range of valid IPs

### 7.5.26      valid\_objectname

**Syntax:**

valid\_objectname (\$variable)

**Input parameters:**

\$variable ... variable (string or array)

**Output:**

variable / false on error

**Description:**

Checks if an expression includes forbidden characters (true) or doesnt (false) to prevent directory browsing

### 7.5.27      valid\_locationname

**Syntax:**

valid\_locationname (\$variable)

**Input parameters:**

\$variable ... variable (string or array)

**Output:**

variable / false on error

**Description:**

Checks if an expression includes forbidden characters (true) or doesnt (false) to prevent directory browsing

## 7.5.28 valid\_publicationname

**Syntax:**

valid\_publicationname (\$variable)

**Input parameters:**

\$variable ... variable (string or array)

**Output:**

variable / false on error

**Description:**

Checks if an expression includes forbidden characters (true) or doesnt (false) to prevent directory browsing

## 7.5.29 html\_encode

**Syntax:**

html\_encode (\$expression, \$encoding="", \$js\_protection=false)

**Input parameters:**

\$expression ... variable as string or array

\$encoding ... conversion of all special characters based on given character set or to ASCII (optional)

\$js\_protection ... remove characters to avoid JS injection [true, false] (optional)

**Output:**

html encoded value as array or string / false on error

**Description:**

This function encodes certain characters (&, <, >, ", ') into their HTML character entity equivalents to protect against XSS.

Converts a string into the html equivalents (also used for XSS protection).

Supports multibyte character sets like UTF-8 as well based on the ASCII value of the character.

## 7.5.30 html\_decode

**Syntax:**

html\_decode (\$expression, \$encoding="")

**Input parameters:**

\$expression ... variable as string or array

\$encoding ... conversion of all special characters based on given character set (optional)

**Output:**

html decoded value as array or string / false on error

**Description:**

This function decodes all characters which have been converted by `html_encode`

### 7.5.31 `scriptcode_encode`

**Syntax:**

`scriptcode_encode ($content)`

**Input parameters:**

`$content` ... content as string

**global input parameters:**

`$mgmt_config`

**Output:**

escaped content as string / false on error

**Description:**

This function escapes all script tags.

This function must be used to clean all user input in the CMS by removing all server side scripts tags.

### 7.5.32 `scriptcode_extract`

**Syntax:**

`scriptcode_extract ($content, $identifier_start("<?", $identifier_end(">"))`

**Input parameters:**

`$content` ... content as string

`$identifier_start` ... identifier of script begin

`$identifier_end` ... and end

**Output:**

script code as array / false on error or if nothing was found

**Description:**

This function extracts the script code of a given content

### 7.5.33 `scriptcode_clean_functions`

**Syntax:**

`scriptcode_clean_functions ($content, $type=3, $application="PHP")`

**Input parameters:**

`$content` ... content as string

`$type` ... cleaning level type from none = 0 to strong = 3 (no cleaning = 0

`$application` ... basic set of disabled functions = 1

**global input parameters:**

`$mgmt_config`

**Output:**

result array / false on error

**Description:**

This function removes all dangerous PHP functions



### 7.5.34      sql\_clean\_functions

**Syntax:**

sql\_clean\_functions (\$content)

**Input parameters:**

\$content ... SQL statement as string

**global input parameters:**

\$mgmt\_config

**Output:**

result array / false on error

**Description:**

This function checks SQL statements for write operations

### 7.5.35      url\_encode

**Syntax:**

url\_encode (\$variable)

**Input parameters:**

\$variable ... variable as string or array

**global input parameters:**

\$mgmt\_config

**Output:**

urlencoded value as array or string / false on error

**Description:**

This function encodes all characters

### 7.5.36      url\_decode

**Syntax:**

url\_decode (\$variable)

**Input parameters:**

\$variable ... variable as string or array

**global input parameters:**

\$mgmt\_config

**Output:**

urldecoded value as array or string / false on error

**Description:**

This function decodes all characters which have been converted by url\_encode or urlencode (PHP)

### 7.5.37      shellcmd\_encode

**Syntax:**

shellcmd\_encode (\$variable)

**Input parameters:**

\$variable ... variable as string or array

**Output:**

encoded value as array or string / false on error

**Description:**

This function encodes/escapes characters to secure the shell comand

### 7.5.38 hcms\_crypt

**Syntax:**

hcms\_crypt (\$string, \$start=0, \$length=0)

**Input parameters:**

\$string ... string to encode

\$start ... start position

\$length ... length for string extraction

**global input parameters:**

\$mgmt\_config

**Output:**

encoded string / false on error

**Description:**

Unidirectional encryption using crypt, MD5 and urlencode

### 7.5.39 hcms\_encrypt

**Syntax:**

hcms\_encrypt (\$string, \$key="", \$crypt\_level="", \$encoding="url")

**Input parameters:**

\$string ... string to encode

\$key ... key of length 16 or 24 or 32 (optional)

\$crypt\_level ... crypt strength level [weak, standard, strong] (optional)

\$encoding ... encoding [base64, url, none] (optional)

**global input parameters:**

\$mgmt\_config

**Output:**

encoded string / false on error

**Description:**

Encryption of a string. Only strong encryption is binary-safe!

### 7.5.40 hcms\_decrypt

**Syntax:**

hcms\_decrypt (\$string, \$key="", \$crypt\_level="", \$encoding="url")

**Input parameters:**

\$string ... hash-string to decode

\$key ... key of length 16 or 24 or 32 (optional)

\$crypt\_level ... crypt strength level [weak, standard, strong] (optional)

\$encoding ... encoding [base64, url, none] (optional)

**global input parameters:**

\$mgmt\_config

**Output:**

decoded string / false on error

**Description:**

Decryption of a string. Only strong encryption is binary-safe!

#### 7.5.41 createtimetoken

**Syntax:**

createtimetoken (\$lifetime=0, \$secret=4)

**Input parameters:**

\$lifetime ... token lifetime in seconds (optional)

\$secret ... secret value (optional)

**global input parameters:**

\$mgmt\_config

**Output:**

token / false on error

#### 7.5.42 checktimetoken

**Syntax:**

checktimetoken (\$token, \$secret=4)

**Input parameters:**

\$token ... token

\$secret ... secret value (optional)

**global input parameters:**

\$mgmt\_config

**Output:**

true / false

#### 7.5.43 createtoken

**Syntax:**

createtoken (\$user="sys", \$lifetime=0, \$secret=4)

**Input parameters:**

\$user ... user name (optional)

\$lifetime ... token lifetime in seconds (optional)

\$secret ... secret value (optional)

**global input parameters:**

\$mgmt\_config

**Output:**

token / false on error

#### 7.5.44 checktoken

**Syntax:**

checktoken (\$token, \$user="sys", \$secret=4)

**Input parameters:**

\$token ... token

\$user ... user name (optional)

\$secret ... secret value (optional)

**global input parameters:**

\$mgmt\_config

**Output:**

true / false

#### 7.5.45 createunique token

**Syntax:**

createunique token (\$length=16)

**Input parameters:**

\$length ... token length (optional)

**global input parameters:**

\$mgmt\_config

**Output:**

token as string / false

#### 7.5.46 rand\_secure

**Syntax:**

rand\_secure (\$min=1000, \$max=999999999999)

**Input parameters:**

\$min ... min and max value as integer (optional)

\$max

**Output:**

secure random number / false

### 7.6 Media API Functions

#### 7.6.1 indexcontent

**Syntax:**

indexcontent (\$site, \$location, \$file, \$container="", \$container\_content="", \$user)

**Input parameters:**

\$site ... publication name

\$location ... path to multimedia file

\$file ... multimedia file name (file to be indexed)

\$container ... container name or ID

\$container\_content ... container XML-content (optional)

\$user ... user name

**global input parameters:**

\$mgmt\_config  
\$mgmt\_parser  
\$mgmt\_uncompress  
\$hcms\_ext  
\$hcms\_lang  
\$lang

**Output:**

true / false

**Description:**

This function extracts the text content of multimedia objects and writes it the text to the container.

The given charset of the publication (not set by default), container or publication (not set by default) will be used.

The default character set of default.meta.tpl is UTF-8, so all content should be saved in UTF-8.

## 7.6.2 unindexcontent

**Syntax:**

unindexcontent (\$site, \$location, \$file, \$container, \$container\_content, \$user)

**Input parameters:**

\$site ... publication name  
\$location ... file location  
\$file ... file name  
\$container ... multimedia file to index  
\$container\_content ... container name or ID  
\$user ... container XML-content

**global input parameters:**

\$mgmt\_config  
\$mgmt\_parser  
\$hcms\_lang  
\$lang

**Output:**

true / false

**Description:**

This function removes media objects from the container

## 7.6.3 createthumbnail\_indesign

**Syntax:**

createthumbnail\_indesign (\$site, \$location\_source, \$location\_dest, \$file)

**Input parameters:**

\$site ... publication  
\$location\_source ... path to source dir  
\$location\_dest ... path to destination dir  
\$file ... file name

**global input parameters:**

\$mgmt\_config  
\$user

**Output:**

new file name / false on error (saves only thumbnail media file in destination location only jpeg format is supported as output)

**Description:**

Creates a thumbnail by extracting the thumbnail from an indesign file and transfers the generated image via remoteclient.

For good results, InDesign Preferences must be set to save preview image and at extra large size.

#### 7.6.4 createthumbnail\_video

**Syntax:**

createthumbnail\_video (\$site, \$location\_source, \$location\_dest, \$file, \$frame)

**Input parameters:**

\$site ... publication

\$location\_source ... path to source dir

\$location\_dest ... path to destination dir

\$file ... file name

\$frame ... frame of video in the seconds or hh:mm:ss[.xxx]

**global input parameters:**

\$mgmt\_config

\$mgmt\_mediapreview

\$user

**Output:**

new file name / false on error (saves only thumbnail media file in destination location only jpeg format is supported as output)

**Description:**

Creates a thumbnail picture of a video frame

#### 7.6.5 createmedia

**Syntax:**

createmedia (\$site, \$location\_source, \$location\_dest, \$file, \$format="", \$type="thumbnail", \$force\_no\_encrypt=false)

**Input parameters:**

\$site ... publication

\$location\_source ... path to source dir

\$location\_dest ... path to destination dir

\$file ... file name

\$format ... format (file extension w/o dot) (optional)

\$type ... type of image/video/audio file [thumbnail(for thumbnails of images), origthumb(thumbnail made from original video/audio), original(to overwrite original video/audio file), any other string present in \$mgmt\_imageoptions/\$mgmt\_mediaoptions] (optional)

\$force\_no\_encrypt ... force the file to be not encrypted even if the content of the publication must be encrypted [true, false] (optional)

**global input parameters:**

\$mgmt\_config

\$mgmt\_imagepreview

\$mgmt\_mediapreview

\$mgmt\_mediaoptions  
\$mgmt\_imageoptions  
\$mgmt\_maxsizepreview  
\$mgmt\_mediametadata  
\$hcms\_ext  
\$user

**Output:**

new file name / false on error

**Description:**

Creates an new image from original or creates a thumbnail and transfers the generated image via remoteclient.

Saves original or thumbnail media file in destination location, for thumbnail only jpeg format is supported as output.

### 7.6.6 convertmedia

**Syntax:**

convertmedia (\$site, \$location\_source, \$location\_dest, \$mediafile, \$format, \$media\_config="", \$force\_no\_encrypt=false)

**Input parameters:**

\$site ... publication name  
\$location\_source ... path to source dir  
\$location\_dest ... path to destination dir  
\$mediafile ... file name  
\$format ... target format (file extension w/o dot) of destination file  
\$media\_config ... media configuration to be used (optional)  
\$force\_no\_encrypt ... force the file to be not encrypted even if the content of the publication must be encrypted [true, false] (optional)

**global input parameters:**

\$mgmt\_config  
\$mgmt\_imagepreview  
\$mgmt\_mediapreview  
\$mgmt\_mediaoptions  
\$mgmt\_imageoptions  
\$mgmt\_maxsizepreview  
\$mgmt\_mediametadata  
\$hcms\_ext

**Output:**

new file name / false on error

**Description:**

Converts and creates a new image/video/audio or document from original. this is a wrapper function for createmedia and createdocument

### 7.6.7 convertimage

**Syntax:**

convertimage (\$site, \$file\_source, \$location\_dest, \$format="jpg", \$colorspace="RGB", \$iccprofile="", \$width="", \$height="", \$slug=0, \$units="px", \$dpi=72, \$quality="")

**Input parameters:**

\$site ... publication name  
\$file\_source ... path to source image file

\$location\_dest ... path to destination dir  
\$format ... format (file extension w/o dot) of destination file (optional)  
\$colorspace ... colorspace of new image [CMY, CMYK, Gray, HCL, HCLp, HSB, HSI, HSL, HSV, HWB, Lab, LCHab, LCHuv, LMS, Log, Luv, OHTA, Rec601YCbCr, Rec709YCbCr, RGB, scRGB, sRGB, Transparent, XYZ, YCbCr, YCC, YDbDr, YIQ, YPbPr, YUV] (optional)  
\$iccprofile ... width in pixel/mm/inch (optional)  
\$width ... height in pixel/mm/inch (optional)  
\$height ... slug in pixel/mm/inch (optional)  
\$slug ... units for width  
\$units ... height and slug [px, mm, inch] (optional)  
\$dpi ... dpi (optional)  
\$quality ... image quality (1 to 100)

**global input parameters:**

\$mgmt\_config  
\$mgmt\_imagepreview  
\$mgmt\_mediapreview  
\$mgmt\_mediaoptions  
\$mgmt\_imageoptions  
\$mgmt\_maxsizepreview  
\$mgmt\_mediametadate  
\$hcms\_ext

**Output:**

new file name / false on error

**Description:**

Converts and creates a new image from original. the new image keeps will be resized and cropped to fit width and height.  
This is a wrapper function for createmedia.

## 7.6.8 rotateimage

**Syntax:**

rotateimage (\$site, \$filepath, \$angle, \$imageformat)

**Input parameters:**

\$site ... publication  
\$filepath ... path to source media file  
\$angle ... rotation angle  
\$imageformat ... destination image format [jpg, png, gif]

**global input parameters:**

\$mgmt\_config

**Output:**

new image file name / false on error

**Description:**

Rotates an image (must be jpg, png or gif) using GD library. not used if ImageMagick is available

## 7.6.9 getimagecolors

**Syntax:**

getimagecolors (\$site, \$file)

**Input parameters:**



\$site ... publication  
\$file ... media file name

**global input parameters:**

\$mgmt\_config  
\$user

**Output:**

result array / false on error

**Description:**

Uses the thumbnail image to calculate the mean color (red, green, blue), defines the colorkey (5 most commonly used colors) and the image type (landscape, portrait, square)

### 7.6.10      getimagecolorkey

**Syntax:**

getimagecolorkey (\$image)

**Input parameters:**

\$image ... image resource

**global input parameters:**

\$mgmt\_config

**Output:**

color key of image / false on error

**Description:**

Extracts the color key for an image that represents the 5 mostly used colors:

K...black  
W...white  
E...grey  
R...red  
G...green  
B...blue  
C...cyan  
M...magenta  
Y...yellow  
O...orange  
P...pink  
N...brown

### 7.6.11      hex2rgb

**Syntax:**

hex2rgb (\$hex)

**Input parameters:**

\$hex ... image color as hex-code

**Output:**

RGB-color as array / false on error

### 7.6.12      rgb2hex

**Syntax:**

rgb2hex (\$red, \$green, \$blue)

**Input parameters:**

\$red ... image color in RGB  
\$green  
\$blue

**Output:**

hex-color as string / false on error

### 7.6.13 readmediaplayer\_config

**Syntax:**

readmediaplayer\_config (\$location, \$configfile)

**Input parameters:**

\$location ... path to media config file  
\$configfile ... config file name

**global input parameters:**

\$mgmt\_config  
\$user

**Output:**

config array / false on error

### 7.6.14 savemediaplayer\_config

**Syntax:**

savemediaplayer\_config (\$location, \$configfile, \$mediafiles, \$width=320, \$height=240, \$filesize="", \$duration="", \$videobitrate="", \$audiobitrate="", \$audiofrequency="", \$audiochannels="")

**Input parameters:**

\$location ... path to media config file  
\$configfile ... media config file name  
\$mediafiles ... media file name array or string  
\$width ... width in px (optional)  
\$height ... height in px (optional)  
\$filesize ... file size in kB (optional)  
\$duration ... duration in hh:mm:ss (optional)  
\$videobitrate ... video bitrate in kb/s (optional)  
\$audiobitrate ... audio bitrate in kb/s (optional)  
\$audiofrequency ... audio frequency in Hz (optional)  
\$audiochannels ... audio channels [mono]

**global input parameters:**

\$mgmt\_config  
\$user

**Output:**

true / false on error

### 7.6.15 createdocument

**Syntax:**

createdocument (\$site, \$location\_source, \$location\_dest, \$file, \$format="", \$force\_no\_encrypt=false)

**Input parameters:**

\$site ... publication  
 \$location\_source ... path to source location  
 \$location\_dest ... path to destination location  
 \$file ... file name  
 \$format ... destination file format (extension w/o dot)  
 \$force\_no\_encrypt ... force the file to be not encrypted even if the content of the publication must be encrypted [true, false] (optional)

**global input parameters:**

\$mgmt\_config  
 \$mgmt\_docpreview  
 \$mgmt\_dcoptions  
 \$mgmt\_docconvert  
 \$mgmt\_maxsizepreview  
 \$hcms\_ext  
 \$hcms\_lang  
 \$lang  
 \$user

**Output:**

new file name / false on error

**Description:**

Creates a new multimedia file of given format at source destination using UNOCONV and saves it as thumbnail file at the destination location

## 7.6.16 unzipfile

**Syntax:**

unzipfile (\$site, \$zipfilepath, \$location, \$filename, \$cat="comp", \$user)

**Input parameters:**

\$site ... publication  
 \$zipfilepath ... path to source zip file  
 \$location ... path to destination location  
 \$filename ... category [page, comp]  
 \$cat ... name of file for extraction  
 \$user ... user

**global input parameters:**

\$mgmt\_config  
 \$mgmt\_uncompress  
 \$mgmt\_imagepreview  
 \$mgmt\_mediapreview  
 \$mgmt\_mediaoptions

**Output:**

result array with all object paths / false

**Description:**

Unpacks ZIP file and creates media files in destination location for components or unzips files directly for pages.

## 7.6.17 zipfiles

**Syntax:**

zipfiles (\$site, \$multiobject\_array, \$destination="", \$zipfilename, \$user, \$activity="")

**Input parameters:**

\$site ... publication  
\$multioobject\_array ... array with path to source zip files  
\$destination ... destination location (if this is null then the \$location where the zip-file resists will be used)  
\$zipfilename ... name of ZIP-file  
\$user ... user name  
\$activity ... activity that need to be set for daily stats [download] (optional)

**global input parameters:**

\$mgmt\_config  
\$mgmt\_compress  
\$pageaccess  
\$compaccess  
\$hiddenfolder  
\$hcms\_linking  
\$globalpermission  
\$setlocalpermission

**Output:**

true/false

**Description:**

Help function that reads and copies all multimedia files from multimedia components to the structure based on the multimedia component names

7.6.18      px2mm

**Syntax:**

px2mm (\$pixel, \$dpi=72)

**Input parameters:**

\$pixel ... pixel  
\$dpi ... dpi (optional)

**Output:**

pixel / false

**Description:**

Convert mm to pixel

7.6.19      px2inch

**Syntax:**

px2inch (\$pixel, \$dpi=72)

**Input parameters:**

\$pixel ... pixel  
\$dpi ... dpi (optional)

**Output:**

inch / false

**Description:**

Convert pixel to inches

## 7.6.20      inch2px

### **Syntax:**

inch2px (\$inch, \$dpi=72)

### **Input parameters:**

\$inch ... pixel

\$dpi ... dpi (optional)

### **Output:**

pixel / false

### **Description:**

Convert inches to pixel

## 7.6.21      vtt2array

### **Syntax:**

vtt2array (\$vtt)

### **Input parameters:**

\$vtt ... VTT string

### **Output:**

array / false

### **Description:**

Converts VTT string to array

## 7.7 Metadata API Functions

### 7.7.1 createtaxonomy

#### **Syntax:**

createtaxonomy (\$recreate=false)

#### **Input parameters:**

\$recreate ... recreate taxonomy file [true, false] (optional)

#### **global input parameters:**

\$mgmt\_config

#### **Output:**

true / false

#### **Description:**

Generates a PHP array from a taxonomy definition file and saves the PHP file in data/include/publication-name.taxonomy.inc.php

### 7.7.2 splitkeywords

#### **Syntax:**

splitkeywords (\$keywords)

#### **Input parameters:**

\$keywords ... comma separated keywords string

**Output:**

keywords as array / false on error

**Description:**

Generates a keyword list from a text by splitting and transforming the comma separated string.

### 7.7.3 copymetadata

**Syntax:**

copymetadata (\$file\_source, \$file\_dest)

**Input parameters:**

\$file\_source ... path to source file

\$file\_dest ... path to destination file

**global input parameters:**

\$mgmt\_config

\$mgmt\_mediametadata

\$user

**Output:**

true / false

**Description:**

Copies all meta data from source to destination file using EXIFTOOL

### 7.7.4 extractmetadata

**Syntax:**

extractmetadata (\$file)

**Input parameters:**

\$file ... path to image file

**global input parameters:**

\$user

\$mgmt\_config

\$mgmt\_mediametadata

**Output:**

result array / false on error

**Description:**

Extracts all meta data from a file using EXIFTOOL

### 7.7.5 xmlobject2array

**Syntax:**

xmlobject2array (\$obj, \$namespace="")

**Input parameters:**

\$obj ... XML as object

\$namespace ... namespace as array (optional)

**Output:**

result array / false

**Description:**

Converts an xmlobject to an array, provided by xaviered at gmail dot com

### 7.7.6 id3\_getdata

**Syntax:**

id3\_getdata (\$file)

**Input parameters:**

\$file ... path to audio file

**global input parameters:**

\$mgmt\_config

\$hcms\_ext

**Output:**

result array / false on error

**Description:**

Requires getID3 library since EXIFTOOL cannot write ID3 tags so far

### 7.7.7 id3\_writefile

**Syntax:**

id3\_writefile (\$file, \$id3, \$keep\_data=true, \$movetempfile=true)

**Input parameters:**

\$file ... abs. path to audio file

\$id3 ... ID3 tag array

\$keep\_data ... keep existing ID3 data of file [true, false] (optional)

\$movetempfile ... move temporary file from unencrypted to encrypted [true, false] (optional)

**global input parameters:**

\$user

\$mgmt\_config

\$mgmt\_mediametadata

\$hcms\_ext

**Output:**

true / false on error

**Description:**

Writes ID3 tags into audio file for supported file types and keeps the existing ID3 tags

### 7.7.8 id3\_create

**Syntax:**

id3\_create (\$site, \$text)

**Input parameters:**

\$site ... publication name

\$text ... text array (from content container)

**global input parameters:**

\$mgmt\_config

**Output:**

ID3 tag array / false on error

**Description:**

Defines ID3 tag array based on the media mapping of a publication.

### 7.7.9 xmp\_getdata

**Syntax:**

xmp\_getdata (\$file)

**Input parameters:**

\$file ... path to image file

**global input parameters:**

\$user

\$mgmt\_config

\$hcms\_ext

**Output:**

result array / false on error

### 7.7.10 xmp\_writefile

**Syntax:**

xmp\_writefile (\$file, \$xmp, \$keep\_data=true, \$movetempfile=true)

**Input parameters:**

\$file ... abs. path to image file

\$xmp ... XMP tag array

\$keep\_data ... keep existing XMP data of file [true, false] (optional)

\$movetempfile ... move temporary file from unencrypted to encrypted [true, false] (optional)

**global input parameters:**

\$user

\$mgmt\_config

\$mgmt\_mediadata

\$hcms\_ext

**Output:**

true / false on error

**Description:**

Writes XMP tags into image file for supported file types and keeps the existing XMP tags

### 7.7.11 xmp\_create

**Syntax:**

xmp\_create (\$site, \$text)

**Input parameters:**

\$site ... publication name

\$text ... text array (from content container)

**global input parameters:**

\$mgmt\_config

**Output:**

XMP tag array / false on error



**Description:**

Defines XMP tag array based on the media mapping of a publication

### 7.7.12      geo2decimal

**Syntax:**

geo2decimal (\$deg, \$min, \$sec, \$hemi)

**Input parameters:**

\$deg ... geo location in degree

\$min ... minutes

\$sec ... seconds

\$hemi ... hemisphere [N, O, S, W]

**Output:**

decimal result / false

### 7.7.13      exif\_getdata

**Syntax:**

exif\_getdata (\$file)

**Input parameters:**

\$file ... path to image file

**global input parameters:**

\$user

\$mgmt\_config

\$hcms\_ext

**Output:**

result array / false

### 7.7.14      iptc\_getdata

**Syntax:**

iptc\_getdata (\$file)

**Input parameters:**

\$file ... path to image file

**global input parameters:**

\$user

\$mgmt\_config

\$hcms\_ext

**Output:**

result array / false

### 7.7.15      iptc\_getcharset

**Syntax:**

iptc\_getcharset (\$tag)

**Input parameters:**

\$tag ... iptc tag that holds character set information

**Output:**

charset as string / false on error

**Description:**

Copied from MediaWiki!

Warning, this function does not (and is not intended to) detect all iso 2022 escape codes. In practise, the code for utf-8 is the only code that seems to have wide use. It does detect that code.

According to iim standard, charset is defined by the tag 1:90.

in which there are iso 2022 escape sequences to specify the character set.

the iim standard seems to encourage that all necessary escape sequences are in the 1:90 tag, but says it doesn't have to be.

This is in need of more testing probably. This is definitely not complete.

however reading the docs of some other iptc software, it appears that most iptc software only recognizes utf-8. If 1:90 tag is not present content is

usually ascii or iso-8859-1 (and sometimes utf-8), but no guarantee.

This also won't work if there are more than one escape sequence in the 1:90 tag

or if something is put in the G2, or G3 charsets, etc. It will only reliably recognize utf-8.

This is just going through the charsets mentioned in appendix C of the iim standard.

### 7.7.16      iptc\_maketag

**Syntax:**

iptc\_maketag (\$record=2, \$tag, \$value)

**Input parameters:**

\$record ... type of tag (e.g. 2)

\$tag ... code of tag (e.g. 025)

\$value ... value of tag

**Output:**

binary IPTC tag / false on error

**Description:**

Convert the IPTC tag to binary code

### 7.7.17      iptc\_writefile

**Syntax:**

iptc\_writefile (\$file, \$iptc, \$keep\_data=true, \$movetempfile=true)

**Input parameters:**

\$file ... abs. path to image file

\$iptc ... IPTC tag array

\$keep\_data ... keep existing IPTC data of file [true, false] (optional)

\$movetempfile ... move temporary file from unencrypted to encrypted [true, false] (optional)

**global input parameters:**

\$user

\$mgmt\_config

\$mgmt\_mediadata

**Output:**

true / false on error

**Description:**

Writes IPTC tags into image file for supported file types and keeps the existing IPTC tags

### 7.7.18 iptc\_create

**Syntax:**

iptc\_create (\$site, \$text)

**Input parameters:**

\$site ... publication name

\$text ... text array (from content container)

**global input parameters:**

\$mgmt\_config

**Output:**

IPTC tag array / false on error

**Description:**

Defines IPTC tag array based on the media mapping of a publication

### 7.7.19 createmapping

**Syntax:**

createmapping (\$site, \$mapping)

**Input parameters:**

\$site ... publication name

\$mapping ... mapping definition

**global input parameters:**

\$mgmt\_config

**Output:**

true / false on error

**Description:**

Prepares the PHP mapping array from the provided mapping definition and saves media mapping file

### 7.7.20 getmapping

**Syntax:**

getmapping (\$site)

**Input parameters:**

\$site ... publication name

**global input parameters:**

\$mgmt\_config

**Output:**

mapping code for display / false

**Description:**

Loads the mapping file of the provided publication

### 7.7.21 setmetadata

**Syntax:**

```
setmetadata ($site, $location="", $object="", $mediafile="", $mapping="",  
$containerdata="", $user, $savecontainer=true)
```

**Input parameters:**

\$site ... publication name  
\$location ... location path (optional)  
\$object ... object name (optional)  
\$mediafile ... media file name (optional)  
\$mapping ... mapping array [meta data tag name -> text-id] (optional)  
\$containerdata ... container content as XML string (optional)  
\$user ... user name  
\$savecontainer ... save content container [true, false] (optional)

**global input parameters:**

\$eventsystem  
\$mgmt\_config  
\$hcms\_ext

**Output:**

container content as XML string / false

**Description:**

Saves meta data of a multimedia file using a provided mapping in the proper fields of the content container.

If no mapping is given a default mapping will be used.

## 7.8 Link API Functions

### 7.8.1 link\_db\_restore

**Syntax:**

```
link_db_restore ($site="")
```

**Input parameters:**

\$site ... publication name (optional)

**global input parameters:**

\$mgmt\_config

**Output:**

true / false on error

**Description:**

This function restores a given or all link management index files

### 7.8.2 link\_db\_load

**Syntax:**

```
link_db_load ($site, $user)
```

**Input parameters:**

\$site ... site  
\$user ... user

**global input parameters:**

\$mgmt\_config

**Output:**

link database [2 dim. array] or true / false on error

**Description:**

This function loads and locks the link management database  
each record of the link management database has the following design:  
xml-content container : | absolute path to 1-n objects : | 1-m links used by 1-n objects  
important: the link management database has to be saved or closed after loading it.

### 7.8.3 link\_db\_read

**Syntax:**

link\_db\_read (\$site)

**Input parameters:**

\$site ... site

**global input parameters:**

\$mgmt\_config

**Output:**

link database [2 dim. array] or true / false on error

**Description:**

This function loads the link management database for reading without locking

### 7.8.4 link\_db\_close

**Syntax:**

link\_db\_close (\$site, \$user)

**Input parameters:**

\$site ... site

\$user ... user

**global input parameters:**

\$mgmt\_config

**Output:**

true/false

**Description:**

closes and unlocks the link management database.

### 7.8.5 link\_db\_save

**Syntax:**

link\_db\_save (\$site, \$link\_db, \$user)

**Input parameters:**

\$site ... link database array

\$link\_db ... site

\$user ... user

**global input parameters:**

\$mgmt\_config

**Output:**

true/false on error

**Description:**

This function saves und unlocks the link management database

### 7.8.6 link\_db\_update

**Syntax:**

link\_db\_update (\$site, \$link\_db, \$attribute, \$contentfile, \$cat, \$link\_curr, \$link\_new, \$option)

**Input parameters:**

\$site ... publication name  
\$link\_db ... link database [2 dim. array]  
\$attribute ... attribute ['object']  
\$contentfile ... 'link'  
\$cat ... content container [optional] [string]  
\$link\_curr ... link category [optional] ['comp']  
\$link\_new ... 'page'  
\$option ... current link [optional]

**global input parameters:**

\$mgmt\_config

**Output:**

link database [2 dim. array] or true / false on error

**Description:**

This function inserts, updates and removes objects and their links from the link management database (add or update a link)

depending on which link is left empty:

link\_curr = "": add new link (just one link matching given category!)

link\_new = "": delete current link in use (just one linkm matching given category!)

link\_curr & link\_new are not empty: update current link with the new one

### 7.8.7 link\_db\_insert

**Syntax:**

link\_db\_insert (\$site, \$link\_db, \$contentfile, \$cat, \$object)

**Input parameters:**

\$site ... publication name  
\$link\_db ... link database [2 dim. array]  
\$contentfile ... content container  
\$cat ... link category ['comp']  
\$object ... page']

**global input parameters:**

\$mgmt\_config

**Output:**

link database [2 dim. array] or true / false

**Description:**

This function inserts a new record in the link management database optionally the created object can be also inserted

### 7.8.8 link\_db\_delete

**Syntax:**

link\_db\_delete (\$site, \$link\_db, \$contentfile)

**Input parameters:**

\$site ... link database [2 dim. array]

\$link\_db ... content container

\$contentfile

**global input parameters:**

\$mgmt\_config

**Output:**

link database [2 dim. array] or true / false on error

**Description:**

This function deletes a record in the link management database

### 7.8.9 link\_db\_getobject

**Syntax:**

link\_db\_getobject (\$multiobject)

**Input parameters:**

\$multiobject ... link database attribut (references to objects seperated by |)

**global input parameters:**

\$mgmt\_config

**Output:**

objects [array] / false on error

**Description:**

This function splits the object string into an array of objects.

### 7.8.10 link\_update

**Syntax:**

link\_update (\$site, \$container, \$link\_old, \$link\_new)

**Input parameters:**

\$site ... publication name

\$container ... container name

\$link\_old ... old link (converted)

\$link\_new ... new link (converted)

**global input parameters:**

\$user

\$mgmt\_config

**Output:**

true/false

**Description:**

This function updates the link of the published and working content container and link file

### 7.8.11 getlinkedobject

**Syntax:**

getlinkedobject (\$site, \$location, \$page, \$cat)

**Input parameters:**

\$site ... publication

\$location ... location

\$page ... object (name and extension)

\$cat ... category [page

**global input parameters:**

\$mgmt\_config

**Output:**

objects which link to the given object [array] or true / false

**Description:**

This function gets all objects which link to the given object.

works with pages (page links) and components (component links) if link management is enabled.

### 7.8.12 getconnectedobject

**Syntax:**

getconnectedobject (\$container, \$type="work")

**Input parameters:**

\$container ... container name

\$type ... container type [work, published, version] (optional)

**global input parameters:**

\$mgmt\_config

\$user

**Output:**

connected objects[array]

**Description:**

This function gets all objects which use the same content container and are therefore connected.

### 7.8.13 extractlinks

**Syntax:**

extractlinks (\$textcontent, \$identifier)

**Input parameters:**

\$textcontent ... text content as string

\$identifier ... link identified ("href" for hyperreferences

**global input parameters:**

\$mgmt\_config

**Output:**

object links [array] / false on error

**Description:**



This function extracts all links based on it's identifier from a text and returns an array of all links

## 7.9 Plugin API Functions

### 7.9.1 plugin\_getdefaultconf

**Syntax:**

plugin\_getdefaultconf ()

**Input parameters:**

**Output:**

default value as array

### 7.9.2 plugin\_readmenu

**Syntax:**

plugin\_readmenu (\$xml, \$pluginFolder)

**Input parameters:**

\$xml ... plugin xml as string

\$pluginFolder ... plugin directory

**global input parameters:**

\$mgmt\_config

**Output:**

menu point array used by navigator

**Description:**

Reads Menupoints and menugroups from the xml data.

Be carefull with nesting, function getcontent is used and you can't nest groups inside of groups as a subpoint!

Input parameter \$pluginFolder contains the directory of the plugin.

Returns an Array containing every group and menupoint with its configuration.

### 7.9.3 plugin\_parse

**Syntax:**

plugin\_parse (\$oldData=array())

**Input parameters:**

\$oldData ... mgmt\_plugin as array (optional)

**global input parameters:**

\$mgmt\_config

**Output:**

mgmt\_plugin as array

**Description:**

Reads the plugin configurations from the file system.

Checks the folder defined in mgmt\_config and searched for plugins and their configurations files.

It either takes needed values from the configuration, from the \$oldData or defaultConfiguration.

#### 7.9.4 plugin\_generateddefinition

**Syntax:**

plugin\_generateddefinition (\$arrayName, \$array)

**Input parameters:**

\$arrayName ... name of array holding the plugin definitions

\$array ... configuration array containing the values and keys

**global input parameters:**

\$mgmt\_config

**Output:**

plugin array / false on error

**Description:**

Generates the Array definition used in php for \$array with the name of \$arrayName.

Run recursively through the array and supports boolean, numeric and string types for the key and value.

#### 7.9.5 plugin\_saveconfig

**Syntax:**

plugin\_saveconfig (\$configuration)

**Input parameters:**

\$configuration ... configuration as array

**global input parameters:**

\$mgmt\_config

**Output:**

true / false on error

**Description:**

Saves the plugin configuration \$configuration into the configuration file.

The configuration file is located in the data/config directory and is named plugin.conf.php.

#### 7.9.6 plugin\_generatelink

**Syntax:**

plugin\_generatelink (\$plugin, \$page, \$control=false, \$additionalGetParameters=false)

**Input parameters:**

\$plugin ... plugin name

\$page ... plugin page (relative reference to the plugins main page)

\$control ... control (relative reference to the plugins control page)

\$additionalGetParameters ... additional GET parameters

**global input parameters:**

\$mgmt\_config

**Output:**

plugin link

**Description:**

Generates a link to be used when linking to other pages inside of a plugin

## 7.10 User Interface API Functions

### 7.10.1 toggleview

**Syntax:**

toggleview (\$view)

**Input parameters:**

\$view ... view [detail, small, medium, large]

**global input parameters:**

\$mgmt\_config

**Output:**

true / false

**Description:**

Sets explorer objectlist view parameter

### 7.10.2 togglesidebar

**Syntax:**

togglesidebar (\$view)

**Input parameters:**

\$view ... view [true, false]

**global input parameters:**

\$mgmt\_config

**Output:**

true / false

**Description:**

Enables or disables the sidebar

### 7.10.3 setfilter

**Syntax:**

setfilter (\$filter\_set)

**Input parameters:**

\$filter\_set ... set of filtera as array with keys [comp, image, document, video, audio] and value [0, 1]

**global input parameters:**

\$mgmt\_config

**Output:**

true / false

**Description:**

Set filter settings for object view in session

### 7.10.4 objectfilter

**Syntax:**

objectfilter (\$file)

**Input parameters:**

\$file ... file name

**global input parameters:**

\$mgmt\_config

\$hcms\_ext

**Output:**

true / false

**Description:**

If an object name is passing the filter-test. One or more filters need to be set in the session "hcms\_objectfilter".

### 7.10.5 showshorttext

**Syntax:**

showshorttext (\$text, \$length=0, \$linebreak=false)

**Input parameters:**

\$text ... text as string

\$length ... max. length of text (minus length starting from the end) (optional)

\$linebreak ... line break instead of cut [true, false] only if length is positive (optional)

**Output:**

shortened text if possible

or original text

**Description:**

Reduce the length of a string and add "..." at the end

### 7.10.6 showtopbar

**Syntax:**

showtopbar (\$show, \$lang="en", \$close\_link="", \$close\_target="", \$individual\_button="", \$id="bar")

**Input parameters:**

\$show ... message

\$lang ... language code (optional)

\$close\_link ... close button link (optional)

\$close\_target ... link target (optional)

\$individual\_button ... individual button (optional)

\$id ... ID of div-layer (optional)

**global input parameters:**

\$mgmt\_config

\$hcms\_charset

\$hcms\_lang

**Output:**

top bar box / false on error

**Description:**

Returns the standard top bar with or without close button

### 7.10.7 showtopmenubar

**Syntax:**

showtopmenubar (\$show, \$menu\_array, \$lang="en", \$close\_link="", \$close\_target="", \$id="bar")

**Input parameters:**

\$show ... message  
\$menu\_array ... menu as array [key=name  
\$lang ... value=properties/events]  
\$close\_link ... language code (optional)  
\$close\_target ... close button link (optional)  
\$id ... link target (optional)

**global input parameters:**

\$mgmt\_config  
\$hcms\_charset  
\$hcms\_lang

**Output:**

top bar box / false on error

**Description:**

Returns the menu top bar with or without close button

### 7.10.8 showmessage

**Syntax:**

showmessage (\$show, \$width="580px", \$height="70px", \$lang="en", \$style="", \$id="hcms\_messageLayer")

**Input parameters:**

\$show ... message  
\$width ... width in pixel (optional)  
\$height ... height in pixel (optional)  
\$lang ... language code (optional)  
\$style ... additional style definitions of div-layer (optional)  
\$id ... ID of div-layer (optional)

**global input parameters:**

\$mgmt\_config  
\$hcms\_charset  
\$hcms\_lang

**Output:**

message box / false on error

**Description:**

Returns the standard message box with close button

### 7.10.9 showinfopage

**Syntax:**

showinfopage (\$show, \$lang="en", \$onload="")

**Input parameters:**

\$show ... message  
\$lang ... language code (optional)

\$onload ... on load JS events (optional)

**global input parameters:**

\$mgmt\_config  
\$hcms\_charset  
\$hcms\_lang\_codepage  
\$hcms\_lang

**Output:**

message on html info page / false on error

**Description:**

Returns a full html info page

## 7.10.10 showinfobox

**Syntax:**

showinfobox (\$show, \$lang="en", \$style="", \$id="hcms\_infoboxLayer")

**Input parameters:**

\$show ... message  
\$lang ... language code (optional)  
\$style ... additional style definitions of div-layer (optional)  
\$id ... ID of div-layer (optional)

**global input parameters:**

\$mgmt\_config  
\$hcms\_charset  
\$hcms\_lang\_codepage  
\$hcms\_lang

**Output:**

message in div layer / false on error

**Description:**

Returns the infobox as long as it has not been closed. Saves the close event in localstorage of browser.

## 7.10.11 showsharelinks

**Syntax:**

showsharelinks (\$link, \$lang="en", \$style="", \$id="hcms\_shareLayer")

**Input parameters:**

\$link ... link to share  
\$lang ... language code (optional)  
\$style ... additional style definitions of div-layer (optional)  
\$id ... ID of div-layer (optional)

**global input parameters:**

\$mgmt\_config  
\$hcms\_charset  
\$hcms\_lang\_codepage  
\$hcms\_lang

**Output:**

message in div layer / false on error

**Description:**

Returns the presentation of share links of social media platforms

### 7.10.12 showmetadata

**Syntax:**

showmetadata (\$data, \$lang="en", \$class\_headline="hcmsRowData2")

**Input parameters:**

\$data ... meta data as array

\$lang ... hierarchy level

\$class\_headline ... CSS-class with background-color for headlines (optional)

**global input parameters:**

\$mgmt\_config

\$hcms\_charset

\$hcms\_lang\_codepage

\$hcms\_lang

**Output:**

result as HTML unordered list / false on error

### 7.10.13 showobject

**Syntax:**

showobject (\$site, \$location, \$page, \$cat="", \$name="")

**Input parameters:**

\$site ... publication name

\$location ... location

\$page ... object name

\$cat ... category [page, comp] (optional)

\$name ... object name (optional)

**global input parameters:**

\$mgmt\_config

\$hcms\_charset

\$hcms\_lang

\$lang

**Output:**

html presentation / false

### 7.10.14 showmedia

**Syntax:**

showmedia (\$mediafile, \$medianame, \$viewtype, \$id="", \$width="", \$height="",  
\$class="hcmsImageItem")

**Input parameters:**

\$mediafile ... mediafile (publication/filename)

\$medianame ... name of mediafile for display

\$viewtype ... view type [template

\$id ... preview

\$width ... preview\_download

\$height ... preview\_no\_rendering]

\$class ... ID of the media tag

**global input parameters:**

\$mgmt\_config  
 \$mgmt\_mediapreview  
 \$mgmt\_mediaoptions  
 \$mgmt\_imagepreview  
 \$mgmt\_docconvert  
 \$hcms\_charset  
 \$hcms\_lang\_codepage  
 \$hcms\_lang  
 \$lang  
 \$site  
 \$location  
 \$cat  
 \$page  
 \$user  
 \$pageaccess  
 \$compaccess  
 \$hiddenfolder  
 \$hcms\_linking  
 \$setlocalpermission  
 \$mgmt\_imageoptions  
 \$is\_mobile

**Output:**

html presentation of any media asset / false

**Description:**

This function requires site, location and cat to be set as global variable in order to validate the access permission of the user

## 7.10.15 showcompexplorer

**Syntax:**

showcompexplorer (\$site, \$dir, \$location\_esc="", \$page="", \$compcat="multi",  
 \$search\_expression="", \$search\_format="", \$mediatype="", \$lang="en", \$callback="",  
 \$scalingfactor="1")

**Input parameters:**

\$site ... publication name  
 \$dir ... current explorer location  
 \$location\_esc ... object location (optional)  
 \$page ... object name (optional)  
 \$compcat ... component category [single, multi, media] (optional)  
 \$search\_expression ... search expression (optional)  
 \$search\_format ... search format [object, document, image, video, audio] (optional)  
 \$mediatype ... media-type [audio, video, text, flash, image, compressed, binary] (optional)  
 \$lang ... callback of CKEditor (optional)  
 \$callback ... scalingfactor for images (optional)  
 \$scalingfactor

**global input parameters:**

\$user  
 \$mgmt\_config  
 \$siteaccess  
 \$pageaccess  
 \$compaccess  
 \$rootpermission  
 \$globalpermission



\$localpermission  
\$hiddenfolder  
\$html5file  
\$temp\_complacement  
\$hcms\_charset  
\$hcms\_lang

**Output:**

explorer with search / false on error

**Description:**

Creates component explorer including the search form

## 7.10.16 showeditor

**Syntax:**

showeditor (\$site, \$hypertagname, \$id, \$contentbot="", \$sizewidth=600, \$sizeheight=300, \$toolbar="Default", \$lang="en", \$dpi=72)

**Input parameters:**

\$site ... publication name  
\$hypertagname ... hypertag name  
\$id ... hypertag id  
\$contentbot ... content  
\$sizewidth ... width  
\$sizeheight ... height of the editor  
\$toolbar ... toolbar set  
\$lang ... language  
\$dpi ... dpi for scaling images

**global input parameters:**

\$mgmt\_config  
\$publ\_config

**Output:**

rich text editor code / false on error

**Description:**

Returns the rich text editor code

## 7.10.17 showinlineeditor\_head

**Syntax:**

showinlineeditor\_head (\$lang)

**Input parameters:**

\$lang ... language

**global input parameters:**

\$mgmt\_config  
\$hcms\_charset  
\$hcms\_lang

**Output:**

rich text editor code for html head section / false on error

**Description:**

Returns the rich text editor code (JS, CSS) for include into the html head section

### 7.10.18 showinlinedatepicker\_head

**Syntax:**

showinlinedatepicker\_head ()

**Input parameters:****global input parameters:**

\$mgmt\_config

**Output:**

date picker code for html head section / false on error

**Description:**

Returns the date picker code (JS, CSS) for include into the html head section

### 7.10.19 showinlineeditor

**Syntax:**

showinlineeditor (\$site, \$hypertag, \$id, \$contentbot="", \$sizewidth=600, \$sizeheight=300, \$toolbar="Default", \$lang="en", \$contenttype="", \$cat="", \$location\_esc="", \$page="", \$contentfile="", \$db\_connect=0, \$token="")

**Input parameters:**

\$site ... publication name

\$hypertag ... hypertag

\$id ... hypertag id

\$contentbot ... content

\$sizewidth ... width

\$sizeheight ... height of the editor

\$toolbar ... toolbar set

\$lang ... language

\$contenttype ... content-type

\$cat ... category[page, comp]

\$location\_esc ... converted location

\$page ... object name

\$contentfile ... container name

\$db\_connect ... DB-connect file name

\$token ... security token

**global input parameters:**

\$mgmt\_config

\$publ\_config

\$hcms\_charset

\$hcms\_lang

**Output:**

rich text editor code / false on error

**Description:**

shows the rich text inline editor

### 7.10.20 showvideoplayer

**Syntax:**

showvideoplayer (\$site, \$video\_array, \$width=320, \$height=240, \$logo\_url="", \$id="", \$title="", \$autoplay=true, \$fullscreen=true, \$loop=false, \$muted=false, \$controls=true, \$iframe=false, \$force\_reload=false)

**Input parameters:**

\$site ... videoArray (Array) containing the different html sources  
 \$video\_array ... width (Integer) Width of the video in pixel  
 \$width ... height (Integer) Height of the video in pixel  
 \$height ... logo\_url (String) Link to the logo which is displayed before you click on play (If the value is null the default logo will be used)  
 \$logo\_url ... id (String) The ID of the video (will be generated when empty)  
 \$id ... title (String) The title for this video  
 \$title ... autoplay (Boolean) Should the video be played on load (true)  
 \$autoplay ... default is false  
 \$fullscreen ... enableFullScreen (Boolean) Is it possible to view the video in fullScreen (true)  
 \$loop ... play loop (optional) [true, false]  
 \$muted ... muted/no sound (optional) [true, false]  
 \$controls ... player controls (optional) [true, false]  
 \$iframe ... use video in iframe (optional) [true, false]  
 \$force\_reload ... reload video sources to prevent the browser cache to show the same video even if it has been changed [true, false] (optional)

**global input parameters:**

\$mgmt\_config

**Output:**

HTML code of the video player / false on error

**Description:**

Generates a html segment for the video player code

### 7.10.21 showvideoplayer\_head

**Syntax:**

showvideoplayer\_head (\$secureHref=true, \$fullscreen=true)

**Input parameters:**

\$secureHref ... secure hyperreferences by adding 'hypercms\_'  
 \$fullscreen ... is it possible to view the video in fullScreen [true, false]

**global input parameters:**

\$mgmt\_config

**Output:**

head for video player / false on error

### 7.10.22 showaudioplayer

**Syntax:**

showaudioplayer (\$site, \$audioArray, \$width=320, \$height=320, \$logo\_url="", \$id="", \$autoplay=false, \$loop=false, \$controls=true, \$force\_reload=false)

**Input parameters:**

\$site ... publication name  
 \$audioArray ... audio files as array (Array)  
 \$width ... ID of the tag (optional)  
 \$height ... autoplay (optional) [true, false]  
 \$logo\_url ... play loop (optional) [true, false]  
 \$id ... player controls (optional) [true, false]  
 \$autoplay  
 \$loop

\$controls  
\$force\_reload

**global input parameters:**

\$mgmt\_config

**Output:**

code of the HTML5 player / false

**Description:**

Generates the html segment for the video player code

## 7.10.23 showaudioplayer\_head

**Syntax:**

showaudioplayer\_head (\$secureHref=true)

**Input parameters:**

\$secureHref ... secure hyperreferences by adding 'hypercms\_'

**global input parameters:**

\$mgmt\_config

**Output:**

head for audio player

## 7.10.24 debug\_getbacktracestring

**Syntax:**

debug\_getbacktracestring (\$valueSeparator, \$rowSeparator, \$ignoreFunctions=array())

**Input parameters:**

\$valueSeparator ... separator for arguments  
\$rowSeparator ... separator for a Row on screen/file  
\$ignoreFunctions ... functionnames to be ignored

**Output:**

debug message

**Description:**

Returns the current backtrace as a good readable string.  
Ignores debug and debug\_getbacktracestring.

## 7.10.25 showAPIdocs

**Syntax:**

showAPIdocs (\$file, \$return="html")

**Input parameters:**

\$file ... path to API file  
\$return ... return result as HTML or array [html, array] (optional)

**global input parameters:**

= array()

**Output:**

HTML output of documentation / false on error

**Description:**

Generates the documentation of an API file

## 7.10.26 readnavigation

**Syntax:**

readnavigation (\$site, \$docroot, \$object, \$view="publish", \$user="sys")

**Input parameters:**

\$site ... publication name

\$docroot ... location

\$object ... object name

\$view ... view [see view parameters of function buildview]

\$user ... user name

**global input parameters:**

\$mgmt\_config

\$navi\_config

**Output:**

navigation item array / false

**Description:**

Reads the content from the container and collects information about a single navigation item

## 7.10.27 createnavigation

**Syntax:**

createnavigation (\$site, \$docroot, \$urlroot, \$view="publish", \$currentobject="", \$recursive=true)

**Input parameters:**

\$site ... publication name

\$docroot ... document root for navigation

\$urlroot ... URL root for navigation

\$view ... view [see view parameters of function buildview]

\$currentobject ... path to current object (optional)

\$recursive ... recursive [true, false] (optional)

**global input parameters:**

\$mgmt\_config

\$navi\_config

**Output:**

navigation array / false

**Description:**

Generates an associative array (item => nav-item, sub => array with sub-items)

## 7.10.28 shownavigation

**Syntax:**

shownavigation (\$navigation, \$level=1)

**Input parameters:**

\$navigation ... navigation array (created by function readnavigation)

\$level ... level as integer (optional)

**global input parameters:**`$mgmt_config``$navi_config`**Output:**

navigation HTML presentation / false

**Description:**

display navigation as HTML code.

The following example configures the navigation:

`$navi_config = array();`

document root definitions:

`$navi_config['root_path'] = "%abs_page%/";``$navi_config['root_url'] = "%url_page%/";`

HTML / CSS class definitions (names between percentage signs are placeholders):

`$navi_config['attr_ul_top'] = "class=\"nav navbar-nav\"";``$navi_config['attr_ul_dropdown'] = "class=\"dropdown-menu\"";``$navi_config['attr_li_active'] = "class=\"active\"";``$navi_config['attr_li_dropdown'] = "class=\"dropdown\"";``$navi_config['attr_href_dropdown'] = "class=\"dropdown-toggle\" data-toggle=\"dropdown\"";``$navi_config['tag_li'] = "<li %attr_li%><a href=\"%link%\" %attr_href%>%title%</a>%sub%</li>\n";``$navi_config['tag_ul'] = "<ul %attr_ul%>%list%</ul>\n";`

language definitions

Session variable name that holds the language setting

`$navi_config['lang_session'] = "langcode";`

note: key = langcode &amp; value = text\_id of textnode

`$navi_config['lang_text_id']['DE'] = "Titel_DE";``$navi_config['lang_text_id']['EN'] = "Titel_EN";`

PermaLink definitions

note: key = langcode &amp; value = text\_id of textnode

`$navi_config['permalink_text_id']['DE'] = "PermaLink_DE";``$navi_config['permalink_text_id']['EN'] = "PermaLink_EN";`

Navigation hide and sort order definitions

`$navi_config['hide_text_id'] = "NavigationHide";``$navi_config['sort_text_id'] = "NavigationSortOrder";`

Use the first item in a folder for the main navigation item and display all following as sub navigation items [true, false]

`$navi_config['use_1st_folderitem'] = false;`

### 7.10.29 showselect

**Syntax:**`showselect ($value_array, $only_text=false, $selected_value="", $id="", $attributes="")`**Input parameters:**`$value_array` ... values array (array-key = value)`$only_text` ... array-value = text)`$selected_value` ... use values of array as option value and text [true, false] (optional)`$id` ... selected value (optional)`$attributes` ... attributes of select tags like name or id or events (optional)**Output:**

HTML select box presentation / false

### 7.10.30 showtranslator

**Syntax:**

showtranslator (\$site, \$id, \$type, \$charset="UTF-8", \$lang="en", \$style="")

**Input parameters:**

\$site ... publication name

\$id ... editor/text-tag ID

\$type ... unformatted or formatted texttag-type [u, f]

\$charset ... character set (optional)

\$lang ... language code (optional)

\$style ... style of div tag (optional)

**global input parameters:**

\$mgmt\_config

\$hcms\_lang

**Output:**

HTML translator box presentation / false

## 7.11 Template Engine API Functions

### 7.11.1 checklanguage

**Syntax:**

checklanguage (\$language\_array, \$language\_value)

**Input parameters:**

\$language\_array ... language array with all valid values

\$language\_value ... language value of attribute in hyperCMS tag

**Output:**

true if language array holds the given language value / false if not found

### 7.11.2 checkgroupaccess

**Syntax:**

checkgroupaccess (\$groupaccess, \$ownergroup)

**Input parameters:**

\$groupaccess ... group access string from hyperCMS group-tag attribute

\$ownergroup ... owner groups as array

**Output:**

true if current ownergroup has access or invalid input / false if not

### 7.11.3 transformlink

**Syntax:**

transformlink (\$viewstore)

**Input parameters:**

\$viewstore ... view of object

**global input parameters:**

\$site

\$location\_esc

\$page  
\$ctrlreload  
\$mgmt\_config

**Output:**

view with transformed links for easyedit mode

#### 7.11.4 followlink

**Syntax:**

followlink (\$site, \$follow)

**Input parameters:**

\$site ... publication name  
\$follow ... link to follow

**global input parameters:**

\$mgmt\_config

**Output:**

prepared input (location plus page) for easyedit mode (buildview) / false on error

#### 7.11.5 errorhandler

**Syntax:**

errorhandler (\$source\_code, \$return\_code, \$error\_identifier)

**Input parameters:**

\$source\_code ... source code  
\$return\_code ... return code  
\$error\_identifier ... error identifier

**Output:**

error message and view of the code with line identifiers

#### 7.11.6 viewinclusions

**Syntax:**

viewinclusions (\$site, \$viewstore, \$hypertag, \$view, \$application, \$charset="UTF-8")

**Input parameters:**

\$site ... view of object  
\$viewstore ... hypertag to create view of included objects  
\$hypertag ... view parameter  
\$view ... application  
\$application ... character set used (optional) view-parameter explanation: \$view = "template or any other word" -> the standard text (in table) will be included for the view \$view = "preview" -> preview of the content of the included file \$view = "publish" -> view the content of the included file as it is (for publishing)  
\$charset

**global input parameters:**

\$user  
\$mgmt\_config  
\$location  
\$hcms\_lang  
\$lang



**Output:**

view on the content including the content of included objects

### 7.11.7 buildview

**Syntax:**

```
buildview ($site, $location, $page, $user, $buildview="template", $ctrlreload="no",
$template="", $container="", $force_cat="", $execute_code=true)
```

**Input parameters:**

\$site ... publication name

\$location ... location

\$page ... object

\$user ... user

\$buildview ... view parameter (optional)

\$ctrlreload ... reload workplace control frame and add html & body tags if missing [yes, no] (optional)

\$template ... template name (optional)

\$container ... container name (optional)

\$force\_cat ... force category to use different location path [page, comp] (optional)

\$execute\_code ... execute\_code [true/false] (optional)

**global input parameters:**

\$container\_collection

\$eventsystem

\$db\_connect

\$mgmt\_config

\$siteaccess

\$adminpermission

\$setlocalpermission

\$token

\$mgmt\_lang\_shortcut\_default

\$hcms\_charset

\$hcms\_lang\_name

\$hcms\_lang\_shortcut

\$hcms\_lang\_codepage

\$hcms\_lang\_date

\$hcms\_lang

\$lang

**Output:**

result array with view of the content / false on error

**Description:**

buildview parameter may have the following values:

\$buildview = "formedit": use form for content editing

\$buildview = "formmeta": use form for content viewing only for meta informations (tag-type must be meta)

\$buildview = "formlock": use form for content viewing

\$buildview = "cmsview": view of page based on template, includes hyperCMS specific code (buttons)

\$buildview = "inlineview": view of page based on template, includes hyperCMS specific code (buttons) and inline text editing

\$buildview = "publish": view of page for publishing based on template without CMS specific code (buttons)

\$buildview = "preview": view of page based on template for preview (inactive hyperlinks) without CMS specific code (buttons)

\$buildview = "template": view of template based on template for preview (inactive hyperlinks) without CMS specific code (buttons)

#### 7.11.8 buildsearchform

**Syntax:**

buildsearchform (\$site="", \$template="", \$report="", \$ownergroup="", \$css\_display="inline-block")

**Input parameters:**

\$site ... publication name (optional for report)  
\$template ... template name (optional)  
\$report ... or report name (optional)  
\$ownergroup ... group access as array (optional)  
\$css\_display ... CSS display value for label tag (optional)

**global input parameters:**

\$user  
\$mgmt\_config  
\$mgmt\_lang\_shortcut\_default  
\$hcms\_charset  
\$hcms\_lang\_name  
\$hcms\_lang\_shortcut  
\$hcms\_lang\_codepage  
\$hcms\_lang\_date  
\$hcms\_lang  
\$lang

**Output:**

form view / false on error

#### 7.11.9 buildbarchart

**Syntax:**

buildbarchart (\$paper\_name, \$paper\_width=600, \$paper\_height=300, \$paper\_top=10, \$paper\_left=40, \$x\_axis, \$y1\_axis, \$y2\_axis="", \$y3\_axis="", \$paper\_style="", \$bar1\_style="", \$bar2\_style="", \$bar3\_style="", \$show\_value=false)

**Input parameters:**

\$paper\_name ... name/id of paper  
\$paper\_width ... width of paper in pixel  
\$paper\_height ... height of paper in pixel  
\$paper\_top ... top space in pixel  
\$paper\_left ... left space in pixel  
\$x\_axis ... x-axis values as array  
\$y1\_axis ... y1-axis values as array  
\$y2\_axis ... y2-axis values as array (optional)  
\$y3\_axis ... y3-axis values as array (optional)  
\$paper\_style ... paper CSS style  
\$bar1\_style ... 1st bar chart CSS style  
\$bar2\_style ... 2nd bar chart CSS style  
\$bar3\_style ... 3rd bar chart CSS style  
\$show\_value ... show y-value in bar [true, false]

**global input parameters:**

\$lang  
\$mgmt\_config

**Output:**

bar chart view / false on error

## 7.12 XML API Functions

### 7.12.1 setxmlparameter

**Syntax:**

setxmlparameter (\$xmldata, \$parameter, \$value)

**Input parameters:**

\$xmldata ... XML content container

\$parameter ... parameter name

\$value ... parameter value

**Output:**

XML content container / false on error

**Description:**

Set parameter values in XML declaration (e.g. encoding): encoding="UTF-8"

### 7.12.2 getcontent

**Syntax:**

getcontent (\$xmldata, \$starttagname)

**Input parameters:**

\$xmldata ... XML content container

\$starttagname ... tag name

**Output:**

result array with the content of the requested XML node (tag) / false on error

**Description:**

<tagname>content</tagname>

Extracts the content between the given \$starttagname xml-tags.

Only this function will decode special characters (&, <, >) in the content and removes CDATA.

Function getcontent will only decode values if they are non-xml and non\_html. so content inside child nodes including tags won't be decoded.

Wild card character "\*" can be used at the end of \$starttagname.

### 7.12.3 getcontent

**Syntax:**

getcontent (\$xmldata, \$starttagname)

**Input parameters:**

\$xmldata ... XML content container

\$starttagname ... tag name

**Output:**

result array with the content of the requested XML node (tag) / false on error

**Description:**

CASE-Insensitive version (XML parser are however always case-sensitive!)

<tagname>content</tagname>

Extracts the content between the given \$starttagname xml-tags.

Only this function will decode special characters (&, <, >) in the content and removes CDATA. getcontent will only decode values if they are non-xml and non\_html. so content inside child nodes including tags won't be decoded.  
Wild card character "\*" can be used at the end of \$starttagname

#### 7.12.4 getxmlcontent

**Syntax:**

getxmlcontent (\$xmldata, \$starttagname)

**Input parameters:**

\$xmldata ... XML content container

\$starttagname ... tag name

**Output:**

result array with the content of the requested XML node (tag) / false on error

**Description:**

<tagname>content</tagname>

Extracts the content together with the \$starttagname xml tags.

This function will NOT decode special characters like function getcontent!

Wild card character "\*" can be used at the end of \$starttagname.

#### 7.12.5 getxmlcontent

**Syntax:**

getxmlcontent (\$xmldata, \$starttagname)

**Input parameters:**

\$xmldata ... XML content container

\$starttagname ... tag name

**Output:**

result array with the content of the requested XML node (tag) / false on error

**Description:**

CASE-Insensitive version (XML parser are always case-sensitive!)

<tagname>content</tagname>

Extracts the content together with the \$starttagname xml tags.

This function will NOT decode special characters like function getcontent!

Wild card character "\*" can be used at the end of \$starttagname.

#### 7.12.6 selectcontent

**Syntax:**

selectcontent (\$xmldata, \$starttagname, \$startcondtag, \$condvalue)

**Input parameters:**

\$xmldata ... XML content container

\$starttagname ... tag name of requested XML node

\$startcondtag ... tag holding the conditional value inside the given starttagname

\$condvalue ... conditional value

**Output:**

result array with the content of the requested XML node (tag) / false on error

**Description:**

<tagname>

```
.....  
<condtag>condvalue</condtag>
```

```
.....  
</tagname>
```

Extracts the content between the given \$starttagname xml tags where the child xml tag \$startcondtag value is equal with the target value \$condvalue.

Wild card character "\*" can be used at the end of \$starttagname.

Wild card character "\*" can be used at begin and end of \$condvalue.

Be Aware: \$startcondtag must be a child of \$starttagname!

### 7.12.7 selectcontent

#### Syntax:

selectcontent (\$xmldata, \$starttagname, \$startcondtag, \$condvalue)

#### Input parameters:

\$xmldata ... XML content container

\$starttagname ... tag name of requested XML node

\$startcondtag ... tag holding the conditional value inside the given starttagname

\$condvalue ... conditional value

#### Output:

result array with the content of the requested XML node (tag) / false on error

#### Description:

CASE-Insensitive version (XML parser are always case-sensitive!)

```
<tagname>
```

```
.....  
<condtag>condvalue</condtag>
```

```
.....  
</tagname>
```

Extracts the content between the given \$starttagname xml tags where the child xml tag \$startcondtag value is equal with the target value \$condvalue.

Wild card character "\*" can be used at the end of \$starttagname.

Wild card character "\*" can be used at begin and end of \$condvalue.

Be Aware: \$startcondtag must be a child of \$starttagname!

### 7.12.8 selectxmlcontent

#### Syntax:

selectxmlcontent (\$xmldata, \$starttagname, \$startcondtag, \$condvalue)

#### Input parameters:

\$xmldata ... XML content container

\$starttagname ... tag name of requested XML node

\$startcondtag ... tag holding the conditional value inside the given starttagname

\$condvalue ... conditional value

#### Output:

result array with the content of the requested XML node (tag) / false on error

#### Description:

```
<tagname>
```

```
.....  
<condtag>condvalue</condtag>
```

```
.....  
</tagname>
```

Extracts the content between the given \$starttagname xml tags where the child xml tag

\$startcondtag value is equal with the target value \$condvalue  
Wild card character "\*" can be used at begin and end of \$condvalue.  
Be Aware: \$startcondtag must be a child of \$starttagname!

### 7.12.9 selectxmlcontent

**Syntax:**

selectxmlcontent (\$xmldata, \$starttagname, \$startcondtag, \$condvalue)

**Input parameters:**

\$xmldata ... XML content container

\$starttagname ... tag name of requested XML node

\$startcondtag ... tag holding the conditional value inside the given starttagname

\$condvalue ... conditional value

**Output:**

result array with the content of the requested XML node (tag) / false on error

**Description:**

CASE-Insensitive version (XML parser are always case-sensitive!)

<tagname>

.....

<condtag>condvalue</condtag>

.....

</tagname>

Extracts the content between the given \$starttagname xml tags where the child xml tag \$startcondtag value is equal with the target value \$condvalue.

Wild card character "\*" can be used at begin and end of \$condvalue.

Be Aware: \$startcondtag must be a child of \$starttagname!

### 7.12.10 deletecontent

**Syntax:**

deletecontent (\$xmldata, \$starttagname, \$startcondtag, \$condvalue)

**Input parameters:**

\$xmldata ... XML content container

\$starttagname ... tag name of requested XML node

\$startcondtag ... tag holding the conditional value inside the given starttagname

\$condvalue ... conditional value

**Output:**

XML content container / false on error

**Description:**

<tagname>

<condtag>condvalue</condtag>

</tagname>

Deletes the whole xml content including <tagname>.

Wild card character "\*" can be used at begin and end of \$condvalue.

### 7.12.11 deleteiccontent

**Syntax:**

deleteiccontent (\$xmldata, \$starttagname, \$startcondtag, \$condvalue)

**Input parameters:**

\$xmldata ... XML content container

\$starttagname ... tag name of requested XML node  
\$startcondtag ... tag holding the conditional value inside the given starttagname  
\$condvalue ... conditional value

**Output:**

XML content container / false on error

**Description:**

CASE-Insensitive version (XML parser are always case-sensitive!)

```
<tagname>  
<condtag>condvalue</condtag>  
</tagname>
```

Deletes the whole xml content including <tagname>.

Wild card character "\*" can be used at begin and end of \$condvalue.

### 7.12.12 setcontent

**Syntax:**

setcontent (\$xmldata, \$startparenttagname, \$starttagname, \$contentnew, \$startcondtag="", \$condvalue="")

**Input parameters:**

\$xmldata ... XML content container  
\$startparenttagname ... parent tag name  
\$starttagname ... tag name of XML node for the new content  
\$contentnew ... new XML node to be inserted  
\$startcondtag ... tag holding the conditional value inside the given starttagname  
\$condvalue ... conditional value

**Output:**

XML content container / false on error

**Description:**

```
<parenttagname>  
<condtag>condvalue</condtag>  
<tagname>contentnew</tagname>  
</parenttagname>
```

\$xmldata = data string to be parsed  
\$startparenttagname = name of the tag that is a parent node of starttagname (necessary if condition has been set!)  
\$starttagname = name of the tag (child node)  
\$contentnew = the content that will be inserted between the child tags \$starttagname  
\$startcondtag = child xml tag where condition will be set  
\$condvalue = value of the condition  
Wild card character "\*" can be used at begin and end of \$condvalue.

### 7.12.13 seticontent

**Syntax:**

seticontent (\$xmldata, \$startparenttagname, \$starttagname, \$contentnew, \$startcondtag, \$condvalue)

**Input parameters:**

\$xmldata ... XML content container  
\$startparenttagname ... parent tag name  
\$starttagname ... tag name of XML node for the new content  
\$contentnew ... new XML node to be inserted  
\$startcondtag ... tag holding the conditional value inside the given starttagname

\$condvalue ... conditional value

**Output:**

XML content container / false on error

**Description:**

CASE-Insensitive version (XML parser are always case-sensitive!)

<parenttagname>

<condtag>condvalue</condtag>

<tagname>contentnew</tagname>

</parenttagname>

\$xmldata = data string to be parsed

\$startparenttagname = name of the tag that is the parent node of starttagname (necessary if condition has been set!)

\$starttagname = name of the tag (child node)

\$contentnew = the content that will be inserted between the child tags \$starttagname

\$startcondtag = child xml tag where condition will be set

\$condvalue = value of the condition

Wild card character "\*" can be used at begin and end of \$condvalue.

### 7.12.14 setcontent\_fast

**Syntax:**

setcontent\_fast (\$xmldata, \$startparenttagname, \$starttagname, \$contentnew, \$startcondtag="", \$condvalue="")

**Input parameters:**

\$xmldata ... XML content container

\$startparenttagname ... parent tag name

\$starttagname ... tag name of XML node for the new content

\$contentnew ... new XML node to be inserted

\$startcondtag ... tag holding the conditional value inside the given starttagname

\$condvalue ... conditional value

**Output:**

XML content container / false on error

**Description:**

function designed for link management, extremely fast but with limitations (only CASE-Sensitive!)

<parenttagname>

<condtag>condvalue</condtag>

<tagname>contentnew</tagname>

</parenttagname>

\$xmldata = data string to be parsed

\$startparenttagname = name of the tag that is the parent node of starttagname (necessary if condition has been set!)

\$starttagname = name of the tag (child node)

\$contentnew = the content that will be inserted between the child tags \$starttagname

\$startcondtag = child xml tag where condition will be set

\$condvalue = value of the condition

Wild card character "\*" can be used at begin and end of \$condvalue.

### 7.12.15 updatecontent

**Syntax:**

updatecontent (\$xmldata, \$xmlnode, \$xmlnodenew)



**Input parameters:**

\$xmldata ... XML content container  
 \$xmlnode ... XML node to be replaced  
 \$xmlnodenew ... new XML node

**Output:**

XML content container / false on error

**Description:**

Updates a given xml string \$xmlnode in \$xmldata with the content \$xmlnodenew.  
 This method provides a faster way to update xml nodes when the node was selected before.

**7.12.16 insertcontent****Syntax:**

insertcontent (\$xmldata, \$insertxmldata, \$starttagname)

**Input parameters:**

\$xmldata ... XML content container  
 \$insertxmldata ... XML node to be inserted in starttagname  
 \$starttagname ... tag name of the parent XML node

**Output:**

XML content container / false on error

**Description:**

```
.....
.....
<tagname> <- list start
.....
.....
insertxmldata <- insertxmldata
</tagname> <- list end
.....
Inserts $insertxmldata string at the end of all child between the parent $tagname .
```

**7.12.17 inserticontent****Syntax:**

inserticontent (\$xmldata, \$insertxmldata, \$starttagname)

**Input parameters:**

\$xmldata ... XML content container  
 \$insertxmldata ... XML node to be inserted in starttagname  
 \$starttagname ... tag name of the parent XML node

**Output:**

XML content container / false on error

**Description:**

CASE-Insensitive version (XML parser are always case-sensitive!)

```
.....
.....
<tagname> <- list start
.....
.....
insertxmldata <- insertxmldata
</tagname> <- list end
```

.....  
Inserts \$insertxmldata string at the end of all child between the parent \$tagname.

### 7.12.18      addcontent

#### **Syntax:**

addcontent (\$xmldata, \$sub\_xmldata, \$startgrandtagname, \$startcondtag, \$condvalue, \$startparenttagname, \$starttagname, \$contentnew)

#### **Input parameters:**

\$xmldata ... XML content container  
\$sub\_xmldata ... xml node to be inserted  
\$startgrandtagname ... grandparent tag name  
\$startcondtag ... tag holding the conditional value inside the given starttagname  
\$condvalue ... conditional value  
\$startparenttagname ... parent tag name  
\$starttagname ... tag name of XML node for the new content  
\$contentnew ... new XML node to be inserted

#### **Output:**

XML content container / false on error

#### **Description:**

```
<grandtagname>
<condtag>condvalue</condtag>
<parenttagname> <- list start
.....
.....
..... }
<tagname>contentnew</tagname> } <- sub_xmldata
..... }
</parenttagname> <- list end
</grandtagname>
$xmldata = data string to be parsed
$sub_xmldata = xml node to be inserted
$startgrandtagname (optional) = name of the grand xml tag of parent xml tag where (article)
$startcondtag (optional) = xml tag inside the parent xml tags where condition will be set
$condvalue (optional) = value of the condition
$startparenttagname (optional) = name of the parent xml tag where the xml subschema
should be added (list)
$starttagname (optional) = name of the tag (child)
$contentnew (optional) = the content that will be inserted between the child tags
```

### 7.12.19      addcontent

#### **Syntax:**

addcontent (\$xmldata, \$sub\_xmldata, \$startgrandtagname, \$startcondtag, \$condvalue, \$startparenttagname, \$starttagname, \$contentnew)

#### **Input parameters:**

\$xmldata ... XML content container  
\$sub\_xmldata ... xml node to be inserted  
\$startgrandtagname ... grandparent tag name  
\$startcondtag ... tag holding the conditional value inside the given starttagname  
\$condvalue ... conditional value  
\$startparenttagname ... parent tag name  
\$starttagname ... tag name of XML node for the new content  
\$contentnew ... new XML node to be inserted

**Output:**

XML content container / false on error

**Description:**

CASE-Insensitive version (XML parser are always case-sensitive!)

<grandtagname>

<condtag>condvalue</condtag>

<parenttagname> <- list start

.....

.....

..... }

<tagname>contentnew</tagname> } <- sub\_xmldata

..... }

</parenttagname> <- list end

</grandtagname>

\$xmldata = data string to be parsed

\$sub\_xmldata = xml subschema to be inserted

\$startgrandtagname (optional) = name of the grand xml tag of parent xml tag where (article)

\$startcondtag (optional) = xml tag inside the parent xml tags where condition will be set

\$condvalue (optional) = value of the condition

\$startparenttagname (optional) = name of the parent xml tag where the xml subschema should be added (list)

\$starttagname (optional) = name of the tag (child)

\$contentnew (optional) = the content that will be inserted between the child tags

## 7.13 Report API Functions

### 7.13.1 createreport

**Syntax:**

createreport (\$report\_name)

**Input parameters:**

\$report\_name ... report name

**global input parameters:**

\$user

\$eventsystem

\$mgmt\_config

\$hcms\_lang

\$lang

**Output:**

result array

**Description:**

This function creates a new report

### 7.13.2 editreport

**Syntax:**

editreport (\$report\_name, \$config)

**Input parameters:**

\$report\_name ... report name

\$config ... report configuration as array

**global input parameters:**

\$user  
\$eventsystem  
\$mgmt\_config  
\$hcms\_lang  
\$lang

**Output:**

result array

**Description:**

This function saves the configuration of a report.

### 7.13.3 loadreport

**Syntax:**

loadreport (\$report\_name)

**Input parameters:**

\$report\_name ... report name

**global input parameters:**

\$user  
\$eventsystem  
\$mgmt\_config  
\$hcms\_lang  
\$lang

**Output:**

result array / false

**Description:**

This function loads the report configuration file and provides the data as array.

### 7.13.4 deletereport

**Syntax:**

deletereport (\$report\_name)

**Input parameters:**

\$report\_name ... report name

**global input parameters:**

\$user  
\$eventsystem  
\$mgmt\_config  
\$hcms\_lang  
\$lang

**Output:**

result array

**Description:**

This function deletes a report

### 7.13.5 analyzeSQLselect

**Syntax:**

analyzeSQLselect (\$sql)

**Input parameters:**

\$sql ... SQL statement as string

**Output:**

result array

**Description:**

This function analyzes an SQL Select statement and return its parts in an array

### 7.13.6 showpiechart

**Syntax:**

showpiechart (\$title="", \$x\_title, \$x\_value, \$y\_title, \$y\_value, \$data\_array, \$width="100%", \$height="100%", \$id="hcmsPieChart", \$suffix="", \$loadAPI=true)

**Input parameters:**

\$title ... chart title

\$x\_title ... chart x-axis title

\$x\_value ... chart x-axis array key name holding the values

\$y\_title ... chart y-axis title

\$y\_value ... chart y-axis array key name holding the values

\$data\_array ... assoz. data array

\$width ... chart width in pixels (optional)

\$height ... chart height in pixels (optional)

\$id ... ID of chart (optional)

\$suffix ... function name suffix (optional)

\$loadAPI ... load Google Chart API [true, false] (optional)

**global input parameters:**

\$mgmt\_config

\$hcms\_lang

\$lang

**Output:**

Google Chart code / false

**Description:**

This function creates the Google Chart Code for a pie chart. The Google Chart API need to be loaded first!

### 7.13.7 showcolumnchart

**Syntax:**

showcolumnchart (\$title="", \$x\_title, \$x\_value, \$y1\_title, \$y1\_value, \$y2\_title="", \$y2\_value="", \$y3\_title="", \$y3\_value="", \$data\_array, \$width="100%", \$height="100%", \$id="hcmsColumnChart", \$suffix="", \$loadAPI=true)

**Input parameters:**

\$title ... chart title

\$x\_title ... chart x-axis title

\$x\_value ... chart x-axis array key name holding the values

\$y1\_title ... chart y1-axis title

\$y1\_value ... chart y1-axis array key name holding the values

\$y2\_title ... chart y2-axis title

\$y2\_value ... chart y2-axis array key name holding the values

\$y3\_title ... chart y3-axis title

\$y3\_value ... chart y3-axis array key name holding the values  
\$data\_array ... assoz. data array  
\$width ... chart width in pixels (optional)  
\$height ... chart height in pixels (optional)  
\$id ... ID of chart (optional)  
\$suffix ... function name suffix (optional)  
\$loadAPI

**global input parameters:**

\$mgmt\_config  
\$hcms\_lang  
\$lang

**Output:**

Google Chart code / false

**Description:**

This function creates the Google Chart Code for a column chart. The Google Chart API need to be loaded first!

### 7.13.8 showtimelinechart

**Syntax:**

showtimelinechart (\$title="", \$y\_title, \$y\_value, \$x1\_title, \$x1\_value, \$x2\_title="",  
\$x2\_value="", \$data\_array, \$width="100%", \$height="100%", \$id="hcmsTimelineChart",  
\$suffix="", \$loadAPI=true)

**Input parameters:**

\$title ... chart title  
\$y\_title ... chart y-axis title  
\$y\_value ... chart y-axis array key name holding the values  
\$x1\_title ... chart x1-axis title  
\$x1\_value ... chart x1-axis array key name holding the values  
\$x2\_title ... chart x2-axis title  
\$x2\_value ... chart x2-axis array key name holding the values  
\$data\_array ... assoz. data array  
\$width ... chart width in pixels (optional)  
\$height ... chart height in pixels (optional)  
\$id ... ID of chart (optional)  
\$suffix ... function name suffix (optional)  
\$loadAPI ... load Google Chart API [true, false] (optional)

**global input parameters:**

\$mgmt\_config  
\$hcms\_lang  
\$lang

**Output:**

Google Chart code / false

**Description:**

This function creates the Google Chart Code for a timeline chart. The Google Chart API need to be loaded first!

### 7.13.9 showgeolocationchart

**Syntax:**

showgeolocationchart (\$title="", \$marker\_value, \$lat\_value, \$lng\_value, \$link\_value,

`$data_array, $width="100%", $height="100%", $id="hcmsGeolocationChart", $suffix="", $loadAPI=true)`

**Input parameters:**

`$title` ... chart title  
`$marker_value` ... array key name holding the value for the marker title  
`$lat_value` ... array key name holding the value for the latitude of the marker  
`$lng_value` ... array key name holding the value for the longitude of the marker  
`$link_value` ... array key name holding the value for the marker link  
`$data_array` ... assoz. data array  
`$width` ... chart width in pixels (optional)  
`$height` ... chart height in pixels (optional)  
`$id` ... ID of chart (optional)  
`$suffix` ... function name suffix (optional)  
`$loadAPI` ... load Google Maps API [true, false] (optional)

**global input parameters:**

`$mgmt_config`  
`$hcms_lang`  
`$lang`

**Output:**

Google Chart code / false

**Description:**

This function creates the Google Maps Code for a geolocation chart. The Google Maps API need to be loaded first!

## 7.14 Project API Functions

### 7.14.1 createproject

**Syntax:**

`createproject ($subproject_id=0, $object_id="", $user, $projectname, $description="")`

**Input parameters:**

`$subproject_id` ... ID of main project (only if project is a subproject)  
`$object_id` ... user name of owner  
`$user` ... object ID or path to object (optional)  
`$projectname` ... project name  
`$description` ... project description (optional)

**global input parameters:**

`$mgmt_config`  
`$hcms_lang_codepage`  
`$hcms_lang`  
`$lang`

**Output:**

true/false

**Description:**

This function creates a new project.

### 7.14.2 setproject

**Syntax:**

`setproject ($project_id, $subproject_id=0, $object_id="", $user="", $projectname="",`

\$description="")

**Input parameters:**

\$project\_id ... project ID  
\$subproject\_id ... ID of main project (only if project is a subproject)  
\$object\_id ... object ID or path to object (optional)  
\$user ... user name of owner (optional)  
\$projectname ... project name (optional)  
\$description ... project description (optional)

**global input parameters:**

\$mgmt\_config  
\$hcms\_lang\_codepage  
\$hcms\_lang  
\$lang

**Output:**

result array

**Description:**

This function saves data of an existing project

### 7.14.3 deleteproject

**Syntax:**

deleteproject (\$project\_id)

**Input parameters:**

\$project\_id ... project ID or array of project IDs to be deleted

**global input parameters:**

\$mgmt\_config  
\$hcms\_lang  
\$lang

**Output:**

result array

**Description:**

This function removes projects.

### 7.14.4 getprojectstructure

**Syntax:**

getprojectstructure (\$project\_id)

**Input parameters:**

\$project\_id ... project ID

**global input parameters:**

\$mgmt\_config

**Output:**

associative result array / false

**Description:**

This function creates an associative array presenting the project structure (project -> subprojects -> tasks)



#### 7.14.5 showprojectrecord

**Syntax:**

showprojectrecord (\$project\_record, \$masterproject\_id)

**Input parameters:**

\$project\_record ... project record as array (project/subproject/task)

\$masterproject\_id ... ID of master project

**global input parameters:**

\$mgmt\_config

\$hcms\_lang

\$lang

\$siteaccess

\$user\_array

**Output:**

html presentation of the record / false

**Description:**

This function creates a table row for the projects presentation

#### 7.14.6 showganttchart

**Syntax:**

showganttchart (\$title="", \$task\_title, \$task\_value, \$resource\_title="", \$resource\_value="", \$start\_title, \$start\_value, \$end\_title="", \$end\_value="", \$status\_title="", \$status\_value="", \$data\_array, \$width="100%", \$height="100%", \$id="hcmsTimelineChart", \$suffix="", \$loadAPI=true)

**Input parameters:**

\$title ... chart title

\$task\_title ... task title

\$task\_value ... task array key name holding the values

\$resource\_title ... resource title (optional)

\$resource\_value ... resource array key name holding the values (optional)

\$start\_title ... task start date title

\$start\_value ... task start date array key name holding the values

\$end\_title ... task finish date title

\$end\_value ... task finish date array key name holding the values

\$status\_title ... work status title (optional)

\$status\_value ... work status array key name holding the values (optional)

\$data\_array ... assoz. data array

\$width ... chart width in pixels (optional)

\$height ... chart height in pixels (optional)

\$id ... ID of chart (optional)

\$suffix ... function name suffix (optional)

\$loadAPI ... load Google Chart API [true, false] (optional)

**global input parameters:**

\$mgmt\_config

\$hcms\_lang

\$lang

**Output:**

Google Chart code / false

**Description:**

This function creates the Google Chart Code for a Gantt chart. The Google Chart API need to be loaded first!

## 7.15 Task API Functions

### 7.15.1 createtask

#### Syntax:

```
createtask ($site="", $from_user, $from_email="", $to_user, $to_email="", $startdate="",  
$finishdate="", $category="", $object_id="", $taskname, $message="", $sendmail=true,  
$priority="low", $project_id=0, $planned="")
```

#### Input parameters:

\$site ... publication name (optional)  
\$from\_user ... from\_user name  
\$from\_email ... from\_email [email-address] (optional)  
\$to\_user ... to\_user name  
\$to\_email ... to\_email [email-address] (optional)  
\$startdate ... start date (optional)  
\$finishdate ... finish date (optional)  
\$category ... category [link, user, workflow] (optional)  
\$object\_id ... object ID or object path  
\$taskname ... task name  
\$message ... message (optional)  
\$sendmail ... sendmail [true/false]  
\$priority ... priority [high, medium, low] (optional)  
\$project\_id ... project ID (optional)  
\$planned ... planned effort in taskunit (optional)

#### global input parameters:

\$mgmt\_config  
\$hcms\_lang\_codepage  
\$hcms\_lang  
\$lang

#### Output:

true/false

#### Description:

Creates a new user task and send optional e-mail to user.  
Since version 5.8.4 the data will be stored in RDBMS instead of XML files.

### 7.15.2 settask

#### Syntax:

```
settask ($task_id, $object_id="", $to_user="", $startdate="", $finishdate="", $taskname="",  
$message="", $sendmail=true, $priority="", $status="", $planned="", $actual="",  
$project_id=0)
```

#### Input parameters:

\$task\_id ... task ID  
\$object\_id ... object ID (optional)  
\$to\_user ... to\_user name (optional)  
\$startdate ... start date (optional)  
\$finishdate ... finish date (optional)  
\$taskname ... category (optional)  
\$message ... task name (optional)  
\$sendmail ... message (optional)

\$priority ... sendmail [true/false]  
\$status ... priority [high, medium, low] (optional)  
\$planned ... status [0-100] (optional)  
\$actual ... planned effort in taskunit (optional)  
\$project\_id ... actual effort in taskunit (optional)

**global input parameters:**

\$mgmt\_config  
\$hcms\_lang\_codepage  
\$hcms\_lang  
\$lang

**Output:**

true/false

**Description:**

Saves data of a user task and send optional e-mail to user.  
Since version 5.8.4 the data will be stored in RDBMS instead of XML files.

### 7.15.3 deletetask

**Syntax:**

deletetask (\$task\_id)

**Input parameters:**

\$task\_id ... task ID or array of task IDs to be deleted

**global input parameters:**

\$mgmt\_config  
\$hcms\_lang  
\$lang

**Output:**

true/false

**Description:**

Deletes user tasks.

### 7.15.4 tasknotification

**Syntax:**

tasknotification (\$date)

**Input parameters:**

\$date ... date

**global input parameters:**

\$mgmt\_config  
\$hcms\_lang\_codepage  
\$hcms\_lang  
\$lang

**Output:**

true/false

**Description:**

Sends e-mail notifications to users if a task starts or ends on the given date.

## 7.16 Workflow API Functions

### 7.16.1 deleteworkflow

**Syntax:**

deleteworkflow (\$site, \$wf\_name, \$cat)

**Input parameters:**

\$site ... publication name

\$wf\_name ... workflow name

\$cat ... category [man, script]

**global input parameters:**

\$user

\$eventsystem

\$mgmt\_config

\$hcms\_lang

\$lang

**Output:**

result array

**Description:**

This function deletes a workflow.

### 7.16.2 buildworkflow

**Syntax:**

buildworkflow (\$workflow\_data)

**Input parameters:**

\$workflow\_data ... workflow [2 dim. Array]

**Output:**

workflow item

### 7.16.3 workflowaccept

**Syntax:**

workflowaccept (\$site, \$location, \$object, \$workflow, \$item\_id, \$user, \$message,  
\$sendmail=true, \$priority="medium")

**Input parameters:**

\$site ... publication name [string]

\$location ... location name [string]

\$object ... object name [string]

\$workflow ... workflow [XML-string]

\$item\_id ... item id [string]

\$user ... user name [string]

\$message ... task message [string]

\$sendmail ... sendmail [true, false]

\$priority ... priority [high, medium, low]

**global input parameters:**

\$mgmt\_config

\$hcms\_lang\_codepage

\$hcms\_lang

\$lang

**Output:**

workflow [XML-string]/false

## 7.16.4 acceptobject

**Syntax:**

acceptobject (\$site, \$location, \$object, \$item\_id, \$user, \$message, \$sendmail, \$priority="medium")

**Input parameters:**

\$site ... publication name [string]  
\$location ... location name [string]  
\$object ... object name [string]  
\$item\_id ... current item id [string]  
\$user ... current user [string]  
\$message ... task message [string]  
\$sendmail ... sendmail  
\$priority ... priority [high, medium, low]

**global input parameters:**

\$mgmt\_config  
\$contentfile  
\$hcms\_lang\_codepage  
\$hcms\_lang  
\$lang

**Output:**

array/false

## 7.16.5 workflowreject

**Syntax:**

workflowreject (\$site, \$location, \$object, \$workflow, \$item\_id, \$user, \$message, \$sendmail, \$priority="medium")

**Input parameters:**

\$site ... publication name [string]  
\$location ... location name [string]  
\$object ... object name [string]  
\$workflow ... workflow [XML-string]  
\$item\_id ... item id [string]  
\$user ... user  
\$message ... task message [string]  
\$sendmail ... send mail [true, false]  
\$priority ... priority [high, medium, low]

**global input parameters:**

\$mgmt\_config  
\$hcms\_lang  
\$lang

**Output:**

workflow [XML-string]/false

### 7.16.6 rejectobject

**Syntax:**

rejectobject (\$site, \$location, \$object, \$item\_id, \$user, \$message, \$sendmail, \$priority="medium")

**Input parameters:**

\$site ... publication name [string]  
\$location ... location name [string]  
\$object ... object name [string]  
\$item\_id ... workflow [XML-string]  
\$user ... item id [string]  
\$message ... user  
\$sendmail ... task message [string]  
\$priority ... send mail [true, false]

**global input parameters:**

\$mgmt\_config  
\$contentfile  
\$hcms\_lang  
\$lang

**Output:**

array/false

## 7.17 Cloud Storage API Functions

### 7.17.1 loadcloudobject

**Syntax:**

loadcloudobject (\$site, \$location, \$file, \$user="", \$save=true, \$directoutput=false)

**Input parameters:**

\$site ... publication name  
\$location ... media file location  
\$file ... media file name  
\$user ... user name (optional)  
\$save ... save file in local media repository or return content as result [true, false] (optional)  
\$directoutput ... direct output of the streamed content [true, false] (optional)

**global input parameters:**

\$mgmt\_config

**Output:**

data / false on error

**Description:**

This functions load/streams an object/file from the cloud storage and saves it the local media repository on request

### 7.17.2 savecloudobject

**Syntax:**

savecloudobject (\$site, \$location, \$file, \$user="")

**Input parameters:**

\$site ... publication name

\$location ... media file location  
\$file ... media file name  
\$user ... user name (optional)

**global input parameters:**

\$mgmt\_config

**Output:**

true / false on error

**Description:**

This function saves an object/file in the cloud storage if no daily cloud snyc has been defined, see main configuration setting \$mgmt\_config['storage\_dailycloudsnyc']

### 7.17.3 copycloudobject

**Syntax:**

copycloudobject (\$site, \$location, \$file\_old, \$file\_new, \$user="")

**Input parameters:**

\$site ... publication name  
\$location ... media file location  
\$file\_old ... source media file name  
\$file\_new ... destination media file name  
\$user ... user name (optional)

**global input parameters:**

\$mgmt\_config

**Output:**

true / false on error

**Description:**

This functions renames an object/file in the cloud storage

### 7.17.4 renamecloudobject

**Syntax:**

renamecloudobject (\$site, \$location, \$file\_old, \$file\_new, \$user="")

**Input parameters:**

\$site ... publication name  
\$location ... media file location  
\$file\_old ... old media file name  
\$file\_new ... new media file name  
\$user ... user name (optional)

**global input parameters:**

\$mgmt\_config

**Output:**

true / false on error

**Description:**

This functions renames an object/file in the cloud storage

### 7.17.5 deletecloudobject

**Syntax:**

deletecloudobject (\$site, \$location, \$file, \$user="")

**Input parameters:**

\$site ... publication name

\$location ... media file location

\$file ... media file name

\$user ... user name (optional)

**global input parameters:**

\$mgmt\_config

**Output:**

true / false on error

**Description:**

This function deletes an object/file and its data from the cloud storage

### 7.17.6 synccloudobjects

**Syntax:**

synccloudobjects (\$user="")

**Input parameters:**

\$user ... user name (optional)

**global input parameters:**

\$mgmt\_config

**Output:**

true / false on error

**Description:**

This function synchronizes all local objects/files with the cloud storage.

See also main configuration settings \$mgmt\_config['storage\_dailycloudsnyc'] and \$mgmt\_config['storage\_type'].

## 7.18 Import/Export API Functions

### 7.18.1 exportobjects

**Syntax:**

exportobjects ()

**Input parameters:****global input parameters:**

\$mgmt\_config

**Output:**

true / false

**Description:**

Executes the export job.



## 7.18.2 importobjects

**Syntax:**

importobjects ()

**Input parameters:****global input parameters:**

\$mgmt\_config

**Output:**

true / false

**Description:**

Executes the import job.

## 8 Legal reference / flag

### 8.1 Questions and suggestions

For advanced questions and suggestions, please contact the support. We are available for every question regarding our reseller- and partner-program. You can apply for an access to our enhanced Online-Demo of the hyper Content Management Servers via our support.

**hyperCMS Support:**

[support@hypercms.com](mailto:support@hypercms.com)

<http://www.hypercms.com>

### 8.2 Imprint

Responsible for the content:

hyperCMS  
Content Management Solutions GmbH  
Rembrandtstr. 35/6  
A-1020 Vienna – Austria

[office@hypercms.com](mailto:office@hypercms.com)  
<http://www.hypercms.com>

### 8.3 Legal information

The present product information is based on the version of the program, which was available at the time the document was composed.

The maker reserves the rights of modifications and corrections of the program.  
Errors and misapprehension accepted.

© 2016 by hyperCMS Content Management Solutions