

PHP OTServ Toolkit



Contents

POT	1
PHP 5.0	3
POT class preview	5
Quick start	6
DAO objects	9
Guilds	13
Guild action drivers	15
Account number hack	18
Server online status	19
About OTServ AAC scripts	21
Package POT Procedural Elements	25
E OTS ErrorCode.php	25
E OTS FileLoaderError.php	26
E OTS Generic.php	27
E OTS NoDriver.php	28
E OTS NotAContainer.php	29
E OTS NotLoaded.php	30
E OTS OTBMEError.php	31
E OTS OutOfBuffer.php	32
E OTS ReadOnly.php	33
IOTS Cipher.php	34
IOTS DAO.php	35
IOTS DB.php	36
IOTS Display.php	37
IOTS FileCache.php	38
IOTS GuildAction.php	39
IOTS ItemsCache.php	40
OTS.php	41
OTS Account.php	42
OTS Accounts List.php	43
OTS Admin.php	44
OTS Base DAO.php	45
OTS Base List.php	46
OTS BinaryTools.php	47
OTS Buffer.php	48
OTS Container.php	49
OTS DB MySQL.php	50
OTS DB ODBC.php	51
OTS DB PostgreSQL.php	52
OTS DB SQLite.php	53
OTS FileLoader.php	54

OTS FileNode.php	55
OTS Group.php	56
OTS Groups List.php	57
OTS Guild.php	58
OTS GuildRank.php	59
OTS GuildRanks List.php	60
OTS Guilds List.php	61
OTS House.php	62
OTS HousesList.php	63
OTS InfoRespond.php	64
OTS Item.php	65
OTS ItemsList.php	66
OTS ItemType.php	67
OTS MapCoords.php	68
OTS Monster.php	69
OTS MonstersList.php	70
OTS OTBMFile.php	71
OTS Player.php	72
OTS Players List.php	73
OTS Row DAO.php	74
OTS RSA.php	75
OTS Spell.php	76
OTS SpellsList.php	77
OTS SQLField.php	78
OTS SQLFilter.php	79
OTS SQLite Results.php	80
OTS Toolbox.php	81
OTS VocationsList.php	82
OTS XTEA.php	83
Package POT Classes	84
Class E OTS ErrorCode	84
Constructor construct	84
Class E OTS FileLoaderError	85
Class Constant ERROR_CAN_NOT_OPEN	85
Class Constant ERROR_EOF	85
Class Constant ERROR_INVALID_FILE_VERSION	86
Class Constant ERROR_INVALID_FORMAT	86
Class Constant ERROR_NOT_OPEN	86
Class Constant ERROR_SEEK_ERROR	87
Class Constant ERROR_TELL_ERROR	87
Class E OTS Generic	88
Class Constant CONNECT_INVALID_DRIVER	88
Class Constant CONNECT_NO_DRIVER	88
Class Constant CREATE_ACCOUNT_IMPOSSIBLE	89
Class E OTS NoDriver	89
Class E OTS NotAContainer	90
Class E OTS NotLoaded	90
Class E OTS OTBMErrors	91
Class Constant LOADMAPERROR_OUTDATEDHEADER	91

Class Constant LOADMAPERROR_UNKNOWNNODETYPE	91
Class E OTS OutOfBuffer	92
Class E OTS ReadOnly	92
Class IOTS Cipher	93
Method decrypt	93
Method encrypt	93
Class IOTS DAO	94
Class IOTS DB	94
Constructor construct	95
Method fieldName	95
Method lastInsertId	96
Method limit	96
Method SQLquery	97
Method SQLquote	97
Method tableName	97
Class IOTS Display	98
Method displayAccount	98
Method displayAccountsList	99
Method displayGroup	99
Method displayGroupsList	100
Method displayGuild	100
Method displayGuildRank	101
Method displayGuildRanksList	101
Method displayGuildsList	102
Method displayPlayer	102
Method displayPlayersList	103
Class IOTS FileCache	103
Method readCache	103
Method writeCache	104
Class IOTS GuildAction	105
Constructor construct	105
Method addRequest	106
Method deleteRequest	106
Method listRequests	106
Method submitRequest	107
Class IOTS ItemsCache	107
Method readItems	108
Method writeItems	108
Class OTS Account	109
Method ban	109
Method block	110
Method count	110
Method create	111
example: account.php	111
Method createEx	112
Method delete	113
Method find	113
Method getAccess	114
Method getCustomField	114

Method getEmail	114
Method getGroup	115
Method getGuildAccess	115
Method getId	116
Method getIterator	116
Method getPACCDays	117
Method getPassword	117
Method getPlayers	117
Method getPlayersList	118
Method isBanned	118
Method isBlocked	119
Method isLoaded	119
Method load	119
Method save	120
Method setCustomField	120
Method setEmail	121
Method setGroup	122
Method setPACCDays	122
Method setPassword	123
Method unban	123
Method unblock	123
Method get	124
Method set	124
Method toString	125
Class OTS_Accounts_List	125
Method deleteAccount	126
Method init	126
Method toString	127
Class OTS_Admin	127
Class Constant COMMAND_BAN_MANAGER	128
Class Constant COMMAND_BROADCAST	128
Class Constant COMMAND_CLOSE_SERVER	128
Class Constant COMMAND_GETHOUSE	129
Class Constant COMMAND_GETONLINE	129
Class Constant COMMAND_KICK	129
Class Constant COMMAND_OPEN_SERVER	130
Class Constant COMMAND_PAY_HOUSES	130
Class Constant COMMAND_PLAYER_INFO	130
Class Constant COMMAND_RELOAD_SCRIPTS	131
Class Constant COMMAND_SERVER_INFO	131
Class Constant COMMAND_SHUTDOWN_SERVER	131
Class Constant ENCRYPTION_RSA1024XTEA	132
Class Constant REQUEST_COMMAND	132
Class Constant REQUEST_ENCRYPTION	132
Class Constant REQUEST_KEY_EXCHANGE	133
Class Constant REQUEST_LOGIN	133
Class Constant REQUEST_PING	133
Class Constant REQUIRE_ENCRYPTION	134
Class Constant REQUIRE_LOGIN	134

Class Constant RESPOND_COMMAND_FAILED	134
Class Constant RESPOND_COMMAND_OK	135
Class Constant RESPOND_ENCRYPTION_FAILED	135
Class Constant RESPOND_ENCRYPTION_OK	135
Class Constant RESPOND_ERROR	136
Class Constant RESPOND_HELLO	136
Class Constant RESPOND_KEY_EXCHANGE_FAILED	136
Class Constant RESPOND_KEY_EXCHANGE_OK	137
Class Constant RESPOND_LOGIN_FAILED	137
Class Constant RESPOND_LOGIN_OK	137
Class Constant RESPOND_MESSAGE	138
Class Constant RESPOND_PING_OK	138
Constructor construct	138
Method broadcast	139
Method close	139
Method login	140
Method payHouses	140
Method ping	141
Method requiresEncryption	141
Method requiresLogin	141
Method send	142
Method shutdown	142
Method usesRSA1024XTEA	142
Method clone	143
Method destruct	143
Method set_state	143
Method sleep	144
Method wakeup	144
Class OTS_Base_DAO	145
Var \$db	145
Constructor construct	146
Method clone	146
Method set_state	146
Method sleep	147
Method wakeup	147
Class OTS_Base_List	148
Var \$class	148
Var \$table	149
Constructor construct	149
Method count	149
Method current	150
Method init	150
Method key	150
Method next	151
Method orderBy	151
Method resetOrder	151
Method rewind	152
Method setFilter	152
Method setLimit	153

Method setOffset	153
Method valid	153
Method set	154
Method set state	154
Method sleep	155
Method wakeup	155
Class OTS BinaryTools	156
Method bin2Int	156
Method int2Bin	157
Method unsignedAdd	157
Method unsignedRightShift	158
Class OTS Buffer	158
Var \$buffer	159
Var \$pos	159
Constructor construct	160
Method check	160
Method getBuffer	161
Method getChar	161
Method getLong	161
Method getPos	162
Method getShort	162
Method getSize	162
Method getString	162
Method isValid	163
Method putChar	163
Method putLong	164
Method putShort	164
Method putString	165
Method reset	165
Method setBuffer	165
Method setPos	166
Method skip	166
Method clone	167
Method get	167
Method set	168
Method set state	168
Method toString	169
Class OTS Container	169
Method addItem	169
Method count	170
Method current	170
Method getIterator	171
Method key	171
Method next	171
Method removeItem	172
Method rewind	172
Method valid	173
Class OTS DB MySQL	173
Constructor construct	174

Method fieldName	174
Method limit	175
Method SQLquery	175
Method SQLquote	176
Method tableName	176
Class OTS_DB_ODBC	177
Constructor construct	177
Method fieldName	178
Method limit	178
Method SQLquery	179
Method SQLquote	179
Method tableName	180
Class OTS_DB_PostgreSQL	180
Constructor construct	181
Method fieldName	181
Method limit	182
Method SQLquery	182
Method SQLquote	183
Method tableName	183
Class OTS_DB_SQLite	184
Constructor construct	184
Method fieldName	185
Method limit	186
Method SQLquery	186
Method SQLquote	187
Method tableName	187
Class OTS_FileLoader	188
Class Constant ESCAPE_CHAR	188
Class Constant NODE_END	188
Class Constant NODE_START	189
Var \$cache	189
Var \$root	189
Method loadFile	190
Method setCacheDriver	190
Method clone	191
Method set	191
Method set_state	192
Method sleep	192
Class OTS_FileNode	193
Method getChild	193
Method getNext	194
Method getType	194
Method setChild	194
Method setNext	195
Method setType	195
Method clone	196
Method get	196
Method set	197
Class OTS_Group	197

Method count	198
Method delete	198
Method find	199
Method getAccess	199
Method getCustomField	200
Method getFlags	200
Method getId	201
Method getIterator	201
Method getMaxDepotItems	202
Method getMaxVIPList	202
Method getName	202
Method getPlayers	203
Method getPlayersList	203
Method isLoaded	204
Method load	204
Method save	204
Method setAccess	205
Method setCustomField	205
Method setFlags	206
Method setMaxDepotItems	207
Method setMaxVIPList	207
Method setName	207
Method get	208
Method set	208
Method toString	209
Class OTS_Groups_List	210
Method deleteGroup	210
Method init	210
Method toString	211
Class OTS_Guild	211
Method acceptInvite	212
Method acceptRequest	212
Method count	213
Method delete	213
Method deleteInvite	214
Method deleteRequest	214
Method find	215
Method getCreationData	215
Method getCustomField	216
Method getGuildRanks	216
Method getGuildRanksList	217
Method getId	217
Method getIterator	218
Method getName	218
Method getOwner	218
Method invite	219
Method isLoaded	219
Method listInvites	220
Method listRequests	220

Method load	220
Method request	221
Method save	221
Method setCreationData	222
Method setCustomField	222
Method setInvitesDriver	223
Method setName	223
Method setOwner	224
Method setRequestsDriver	224
Method clone	225
Method get	225
Method set	226
Method sleep	226
Method toString	227
Class OTS_GuildRank	227
Method count	228
Method delete	228
Method find	229
Method getCustomField	229
Method getGuild	230
Method getId	230
Method getIterator	230
Method getLevel	231
Method getName	231
Method getPlayers	232
Method getPlayersList	232
Method isLoaded	233
Method load	233
Method save	233
Method setCustomField	234
Method setGuild	234
Method setLevel	235
Method setName	235
Method get	236
Method set	236
Method toString	237
Class OTS_GuildRanks_List	238
Method deleteGuildRank	238
Method init	238
Method toString	239
Class OTS_Guilds_List	239
Method deleteGuild	240
Method init	240
Method toString	240
Class OTS_House	241
Constructor construct	242
Method addTile	242
Method delete	242
Method getEntry	243

Method getId	243
Method getName	243
Method getOwner	244
Method getPaid	244
Method getRent	244
Method getSize	245
Method getTiles	245
Method getTownId	245
Method getTownName	246
Method getWarnings	246
Method save	246
Method setOwner	247
Method setPaid	247
Method setWarnings	248
Method get	248
Method set	249
Method sleep	249
Class OTS HousesList	250
Constructor construct	250
Method count	250
Method getHouse	251
Method getHouseId	251
Method getIterator	252
Method offsetExists	252
Method offsetGet	252
Method offsetSet	253
Method offsetUnset	253
Method set state	254
Class OTS InfoRespond	254
Method getClientVersion	256
Method getEmail	256
Method getIP	256
Method getLocation	257
Method getMapAuthor	257
Method getMapHeight	257
Method getMapName	258
Method getMapWidth	258
Method getMaxPlayers	258
Method getMonstersCount	259
Method getMOTD	259
Method getName	259
Method getOnlinePlayers	260
Method getOwner	260
Method getPlayersPeak	260
Method getPort	260
Method getServer	261
Method getServerVersion	261
Method getTSPQVersion	261
Method getUptime	262

Method <u>getURL</u>	262
Method <u>get</u>	262
Method <u>toString</u>	263
Class <u>OTS Item</u>	264
Constructor <u>construct</u>	264
Method <u>count</u>	264
Method <u>getAttributes</u>	265
Method <u>getCount</u>	265
Method <u>getId</u>	265
Method <u>getItemType</u>	266
Method <u>setAttributes</u>	266
Method <u>setCount</u>	267
Method <u>get</u>	267
Method <u>set</u>	268
Class <u>OTS ItemsList</u>	268
Class Constant <u>CLIENT_VERSION 750</u>	269
Class Constant <u>CLIENT_VERSION 755</u>	269
Class Constant <u>CLIENT_VERSION 760</u>	270
Class Constant <u>CLIENT_VERSION 770</u>	270
Class Constant <u>CLIENT_VERSION 780</u>	270
Class Constant <u>CLIENT_VERSION 790</u>	271
Class Constant <u>CLIENT_VERSION 792</u>	271
Class Constant <u>CLIENT_VERSION 800</u>	271
Class Constant <u>ITEM_ATTR_CLIENTID</u>	272
Class Constant <u>ITEM_ATTR_LIGHT2</u>	272
Class Constant <u>ITEM_ATTR_SERVERID</u>	272
Class Constant <u>ITEM_ATTR_SPEED</u>	273
Class Constant <u>ITEM_ATTR_TOPORDER</u>	273
Class Constant <u>ROOT_ATTR_VERSION</u>	273
Method <u>count</u>	274
Method <u>current</u>	274
Method <u>getBuildVersion</u>	274
Method <u>getClientVersion</u>	275
Method <u>getItemType</u>	275
Method <u>getItemTypeId</u>	275
Method <u>getItemTypesList</u>	276
Method <u>getIterator</u>	276
Method <u>getOTBVersion</u>	277
Method <u>key</u>	277
Method <u>loadItems</u>	277
Method <u>next</u>	278
Method <u>offsetExists</u>	278
Method <u>offsetGet</u>	279
Method <u>offsetSet</u>	279
Method <u>offsetUnset</u>	280
Method <u>rewind</u>	280
Method <u>valid</u>	281
Method <u>get</u>	281
Method <u>set state</u>	282

Method wakeup	282
Class OTS ItemType	283
Class Constant FLAG ALLOWDISTREAD	284
Class Constant FLAG ALWAYSONTOP	284
Class Constant FLAG BLOCK_PATHFIND	284
Class Constant FLAG BLOCK_PROJECTILE	285
Class Constant FLAG BLOCK_SOLID	285
Class Constant FLAG CANNOTDECAY	285
Class Constant FLAG FLOORCHANGEDOWN	286
Class Constant FLAG FLOORCHANGEEST	286
Class Constant FLAG FLOORCHANGENORTH	286
Class Constant FLAG FLOORCHANGESOUTH	287
Class Constant FLAG FLOORCHANGEWEST	287
Class Constant FLAG HANGABLE	287
Class Constant FLAG HAS_HEIGHT	288
Class Constant FLAG HORIZONTAL	288
Class Constant FLAG MOVEABLE	288
Class Constant FLAG PICKUPABLE	289
Class Constant FLAG READABLE	289
Class Constant FLAG ROTABLE	289
Class Constant FLAG STACKABLE	290
Class Constant FLAG USEABLE	290
Class Constant FLAG VERTICAL	290
Class Constant ITEM GROUP AMMUNITION	291
Class Constant ITEM GROUP ARMOR	291
Class Constant ITEM GROUP CONTAINER	291
Class Constant ITEM GROUP DEPRECATED	292
Class Constant ITEM GROUP DOOR	292
Class Constant ITEM GROUP FLUID	293
Class Constant ITEM GROUP GROUND	293
Class Constant ITEM GROUP KEY	293
Class Constant ITEM GROUP MAGICFIELD	294
Class Constant ITEM GROUP NONE	294
Class Constant ITEM GROUP RUNE	294
Class Constant ITEM GROUP SPLASH	295
Class Constant ITEM GROUP TELEPORT	295
Class Constant ITEM GROUP WEAPON	295
Class Constant ITEM GROUP WRITEABLE	296
Class Constant ITEM TYPE CONTAINER	296
Class Constant ITEM TYPE DEPOT	296
Class Constant ITEM TYPE DOOR	297
Class Constant ITEM TYPE MAGICFIELD	297
Class Constant ITEM TYPE MAILBOX	297
Class Constant ITEM TYPE NONE	298
Class Constant ITEM TYPE TELEPORT	298
Class Constant ITEM TYPE TRASHHOLDER	298
Constructor construct	299
Method createItem	299
Method getAttribute	299

Method <u>getAttributesList</u>	300
Method <u>getClientId</u>	300
Method <u>getGroup</u>	301
Method <u>getId</u>	301
Method <u>getName</u>	301
Method <u>getType</u>	302
Method <u>hasHeight</u>	302
Method <u>isAlwaysOnTop</u>	302
Method <u>isBlocking</u>	303
Method <u>isHangable</u>	303
Method <u>isHorizontal</u>	303
Method <u>isMovable</u>	303
Method <u>isPickupable</u>	304
Method <u>isReadable</u>	304
Method <u>isRotable</u>	304
Method <u>isStackable</u>	305
Method <u>isUsable</u>	305
Method <u>isVertical</u>	305
Method <u>setAttribute</u>	306
Method <u>setClientId</u>	306
Method <u>setFlags</u>	307
Method <u>setGroup</u>	307
Method <u>setName</u>	308
Method <u>setType</u>	308
Method <u>get</u>	309
Method <u>set</u>	309
Method <u>set state</u>	310
Class <u>OTS_MapCoords</u>	310
 Constructor <u>construct</u>	311
 Method <u>getX</u>	311
 Method <u>getY</u>	312
 Method <u>getZ</u>	312
 Method <u>get</u>	312
 Method <u>set state</u>	313
Class <u>OTS_Monster</u>	313
 Method <u>getArmor</u>	314
 Method <u>getAttacks</u>	315
 Method <u>getDefense</u>	315
 Method <u>getDefenses</u>	315
 Method <u>getExperience</u>	316
 Method <u>getFlag</u>	316
 Method <u>getFlags</u>	316
 Method <u>getHealth</u>	317
 Method <u>getImmunities</u>	317
 Method <u>getItems</u>	317
 Method <u>getLoot</u>	318
 Method <u>getManaCost</u>	318
 Method <u>getName</u>	318
 Method <u>getRace</u>	319

Method <code>getSpeed</code>	319
Method <code>getVoices</code>	319
Method <code>hasImmunity</code>	320
Method <code>get</code>	320
Method <code>toString</code>	321
Class <code>OTS MonstersList</code>	321
Constructor <code>construct</code>	322
Method <code>count</code>	322
Method <code>current</code>	322
Method <code>getMonster</code>	323
Method <code>key</code>	323
Method <code>next</code>	324
Method <code>offsetExists</code>	324
Method <code>offsetGet</code>	324
Method <code>offsetSet</code>	325
Method <code>offsetUnset</code>	325
Method <code>rewind</code>	326
Method <code>valid</code>	326
Method <code>set state</code>	326
Class <code>OTS OTBMFile</code>	327
Class Constant <code>OTBM_ATTR_ACTION_ID</code>	328
Class Constant <code>OTBM_ATTR_DEPOT_ID</code>	328
Class Constant <code>OTBM_ATTR_DESC</code>	328
Class Constant <code>OTBM_ATTR_DESCRIPTION</code>	329
Class Constant <code>OTBM_ATTR_EXT_FILE</code>	329
Class Constant <code>OTBM_ATTR_EXT_HOUSE_FILE</code>	329
Class Constant <code>OTBM_ATTR_EXT_SPAWN_FILE</code>	330
Class Constant <code>OTBM_ATTR_HOUSEDOORID</code>	330
Class Constant <code>OTBM_ATTR_ITEM</code>	330
Class Constant <code>OTBM_ATTR_RUNE_CHARGES</code>	331
Class Constant <code>OTBM_ATTR_TELE_DEST</code>	331
Class Constant <code>OTBM_ATTR_TEXT</code>	331
Class Constant <code>OTBM_ATTR_TILE_FLAGS</code>	332
Class Constant <code>OTBM_ATTR_UNIQUE_ID</code>	332
Class Constant <code>OTBM_NODE_HOusetile</code>	332
Class Constant <code>OTBM_NODE_ITEM</code>	333
Class Constant <code>OTBM_NODE_ITEM_DEF</code>	333
Class Constant <code>OTBM_NODE_MAP_DATA</code>	333
Class Constant <code>OTBM_NODE_MONSTER</code>	334
Class Constant <code>OTBM_NODE_ROOTV1</code>	334
Class Constant <code>OTBM_NODE_SPAWNS</code>	334
Class Constant <code>OTBM_NODE_SPAWN_AREA</code>	335
Class Constant <code>OTBM_NODE_TILE</code>	335
Class Constant <code>OTBM_NODE_TILE_AREA</code>	335
Class Constant <code>OTBM_NODE_TILE_REF</code>	336
Class Constant <code>OTBM_NODE_TILE_SQUARE</code>	336
Class Constant <code>OTBM_NODE_TOWN</code>	336
Class Constant <code>OTBM_NODE_TOWNS</code>	337
Method <code>count</code>	337

Method current	337
Method getDescription	338
Method getHeight	338
Method getHousesList	339
Method getIterator	339
Method getTownID	339
Method getTownName	340
Method getTownsList	340
Method getTownTemple	341
Method getWidth	341
Method key	341
Method loadFile	342
Method next	342
Method offsetExists	343
Method offsetGet	343
Method offsetSet	344
Method offsetUnset	344
Method rewind	345
Method valid	345
Method get	346
Method set state	346
Method wakeup	347
Class OTS Player	347
Method ban	349
Method delete	350
Method find	350
Method getAccount	351
Method getBalance	351
Method getCap	351
Method getConditions	352
Method getCustomField	352
Method getDepot	353
Method getDirection	354
Method getExperience	354
Method getGroup	354
Method getGuildNick	355
Method getHealth	355
Method getHealthMax	356
Method getHouse	356
Method getId	356
Method getLastIP	357
Method getLastLogin	357
Method getLevel	357
Method getLookAddons	358
Method getLookBody	358
Method getLookFeet	359
Method getLookHead	359
Method getLookLegs	359
Method getLookType	360

Method getLossExperience	360
Method getLossMana	361
Method getLossSkills	361
Method getMagLevel	361
Method getMana	362
Method getManaMax	362
Method getManaSpent	363
Method getName	363
Method getPosX	363
Method getPosY	364
Method getPosZ	364
Method getPremiumEnd	365
Method getRank	365
Method getRankId	365
Method getRedSkullTime	366
Method getSave	366
Method getSex	367
Method getSkill	367
Method getSkillTries	368
Method getSlot	368
Method getSoul	369
Method getStorage	369
Method getTownId	370
Method getTownName	370
Method getVocation	371
Method getVocationName	371
Method hasRedSkull	371
Method isBanned	372
Method isLoading	372
Method isSaveSet	372
Method load	373
Method save	373
Method setAccount	374
Method setBalance	374
Method setCap	375
Method setConditions	375
Method setCustomField	376
Method setDepot	376
Method setDirection	377
Method setExperience	378
Method setGroup	378
Method setGuildNick	378
Method setHealth	379
Method setHealthMax	379
Method setLastIP	380
Method setLastLogin	380
Method setLevel	381
Method setLookAddons	381
Method setLookBody	382

Method setLookFeet	382
Method setLookHead	382
Method setLookLegs	383
Method setLookType	383
Method setLossExperience	384
Method setLossMana	384
Method setLossSkills	385
Method setMagLevel	385
Method setMana	386
Method setManaMax	386
Method setManaSpent	387
Method setName	387
Method setPosX	387
Method setPosY	388
Method setPosZ	388
Method setPremiumEnd	389
Method setRank	389
Method setRankId	390
Method setRedSkull	390
Method setRedSkullTime	391
Method setSave	391
Method setSex	392
Method setSkill	392
Method setSkillTries	393
Method setSlot	393
Method setSoul	394
Method setStorage	394
Method setTownId	395
Method setVocation	395
Method unban	396
Method unsetRedSkull	396
Method unsetSave	396
Method get	397
Method set	397
Method sleep	398
Method toString	398
Class OTS Players List	399
Method deletePlayer	399
Method init	400
Method toString	400
Class OTS Row DAO	401
Constructor construct	401
Method find	401
Method load	402
Class OTS RSA	402
Class Constant D	403
Class Constant P	403
Class Constant Q	404
Constructor construct	404

Method decrypt	405
Method encrypt	405
Class OTS Spell	406
Constructor construct	407
Method createConjure	407
Method getCharges	408
Method getConjure	408
Method getConjureCount	408
Method getConjureId	409
Method getID	409
Method getLevel	409
Method getMagicLevel	410
Method getMana	410
Method getName	410
Method getReagent	411
Method getReagentId	411
Method getSoul	411
Method getType	412
Method getVocations	412
Method getWords	412
Method hasParams	413
Method isAggressive	413
Method isAggressive	413
Method isEnabled	414
Method isFarUseAllowed	414
Method isLearnNeeded	414
Method isPremium	415
Method get	415
Method toString	416
Class OTS SpellsList	416
Class Constant SPELL_CONJURE	417
Class Constant SPELL_INSTANT	417
Class Constant SPELL_RUNE	417
Constructor construct	418
Method getConjure	418
Method getConjuresList	419
Method getInstant	419
Method getInstantsList	419
Method getRune	420
Method getRunesList	420
Method get	420
Method set state	421
Class OTS SQLField	421
Constructor construct	422
Method getName	422
Method getTable	423
Method get	423
Method toString	424
Class OTS SQLFilter	424

Class Constant CRITERIUM AND	425
Class Constant CRITERIUM OR	425
Class Constant OPERATOR EQUAL	425
Class Constant OPERATOR GREATER	426
Class Constant OPERATOR LIKE	426
Class Constant OPERATOR LOWER	426
Class Constant OPERATOR NEQUAL	427
Class Constant OPERATOR NGREATER	427
Class Constant OPERATOR NLIKE	427
Class Constant OPERATOR NLOWER	428
Method addFilter	428
Method compareField	429
Method getTables	429
Method get	430
Method sleep	430
Method toString	430
Class OTS Toolbox	431
Method experienceForLevel	431
Method levelForExperience	432
Class OTS VocationsList	432
Constructor construct	433
Method count	433
Method getIterator	434
Method getVocationId	434
Method getVocationName	434
Method offsetExists	435
Method offsetGet	435
Method offsetSet	436
Method offsetUnset	436
Method set state	437
Class OTS XTEA	437
Constructor construct	438
Method decrypt	438
Method encrypt	439
Class POT	439
Class Constant BAN ACCOUNT	440
Class Constant BAN IP	440
Class Constant BAN PLAYER	440
Class Constant DB MYSQL	441
Class Constant DB ODBC	441
Class Constant DB PGSQL	442
Class Constant DB SQLITE	442
Class Constant DEPOT_SID_FIRST	442
Class Constant DIRECTION EAST	443
Class Constant DIRECTION NORTH	443
Class Constant DIRECTION SOUTH	443
Class Constant DIRECTION WEST	444
Class Constant ORDER ASC	444
Class Constant ORDER DESC	445

Class Constant SEX_FEMALE	445
Class Constant SEX_MALE	445
Class Constant SKILL_AXE	446
Class Constant SKILL_CLUB	446
Class Constant SKILL_DISTANCE	446
Class Constant SKILL_FISHING	447
Class Constant SKILL_FIST	447
Class Constant SKILL_SHIELDING	448
Class Constant SKILL_SWORD	448
Class Constant SLOT_AMMO	448
Class Constant SLOT_ARMOR	449
Class Constant SLOT_BACKPACK	449
Class Constant SLOT_FEET	450
Class Constant SLOT_HEAD	450
Class Constant SLOT_LEFT	451
Class Constant SLOT_LEGS	451
Class Constant SLOT_NECKLACE	451
Class Constant SLOT_RIGHT	452
Class Constant SLOT_RING	452
Class Constant SPELL_CONJURE	453
Class Constant SPELL_INSTANT	453
Class Constant SPELL_RUNE	453
Class Constant VOCATION_DRUID	454
Class Constant VOCATION_KNIGHT	454
Class Constant VOCATION_NONE	455
Class Constant VOCATION_PALADIN	455
Class Constant VOCATION_SORCERER	455
Method areHousesLoaded	456
Method areItemsLoaded	456
Method areMonstersLoaded	456
Method areSpellsLoaded	457
Method areVocationsLoaded	457
Method banIP	458
Method connect	458
example: connect.php	458
Method createFilter	460
Method createObject	460
Method getConjure	461
Method getConjuresList	461
Method getDBHandle	462
Method getDisplayDriver	462
Method getHouse	463
Method getHouseld	463
Method getHousesList	464
Method getInstance	464
Method getInstant	464
Method getInstantsList	465
Method getItemsList	465
Method getItemType	466

Method getItemTypeId	466
Method getMap	467
Method getMapDescription	467
Method getMapHeight	468
Method getMapWidth	468
Method getMonster	469
Method getMonstersList	469
Method getRune	470
Method getRunesList	470
Method getSpellsList	471
Method getTownId	471
Method getTownName	472
Method getVocationId	472
Method getVocationName	473
Method getVocationsList	473
Method isDisplayDriverLoaded	474
Method isIPBanned	474
Method isMapLoaded	475
Method loadClass	475
Method loadHouses	476
Method loadItems	476
Method loadMap	477
Method loadMonsters	477
Method loadSpells	478
Method loadVocations	478
Method serverStatus	479
example: example	479
Method setDisplayDriver	480
Method setItemsCache	480
Method setMapCache	481
Method setPOTPath	481
example: fakeroot.php	481
Method unbanIP	482
Method unloadDisplayDriver	483
Method unloadHouses	483
Method unloadItems	483
Method unloadMap	484
Method unloadMonsters	484
Method unloadSpells	485
Method unloadVocations	485
compat.php	486
Appendices	487
Appendix A - Class Trees	488
POT	488
Appendix B - README/CHANGELOG/INSTALL	495
LICENSE	496
CHANGELOG	498
RULES	500
README	502

INSTALL	503
NEWS	503
Appendix D - Todo List	505

POT

This is documentation of POT - official toolkit for OTServ AAC scripts.

PHP OTServ Toolkit

There are several reasons why POT was created:

- Just because it was needed - OTServ should have had that long time ago.
- To unify AAC scripts - there are tons of them, and you never know how to write even a single line of code to them as each of them are created different way.
- To provide reliable way of database accessing - most of people who create AAC scripts don't know what PHP really is, how to use it, they just "want to make own AAC script".
- To provide easy interface - people who write in PHP want to write in PHP, not using SQL, XML and many other languages. POT provides abstract PHP interface for data stored in database.

POT has been created for latest SVN release, it will work best with pure SVN servers. However it provides routines to access custom database structure elements. However it won't work with broken database - it relies on database foreign key constraints, triggers etc.

System requirements

To use POT you need [PHP](#) version at least 5.0 with [PDO extension installed](#) (so it means you will mostly need PHP 5.1, but it is possible to download PDO as external libraries for PHP 5.0.x).

DOM XML

POT uses [DOM PHP extension](#) which is available in core PHP5. However some people still have enabled outdated [DOMXML extension](#) which was created for PHP4, but is still possible to compile it for PHP5. If you use any code that uses PHP DOM extension (which POT does) make sure you have DOMXML extension disabled in your php.ini file.

What POT is

POT is a toolkit/library for accessing OTServ database from PHP. It provides PHP classes that represents OTServ database information as an objects.

What POT is not

- It is not AAC script - this is a toolkit for making them, but you can't directly run it as website. It has only programming interface.
- It is not application/system framework - you won't create website with only POT. POT has only functionality connected with OTServ database, it doesn't contain for example templates engine. You also won't be able to use it as an ordinary database connection engine - it makes use of [PDO](#) so you can use PDO by itself, POT doesn't provide any additional universal functionality. All it's classes are strictly connected with OTServ database.

How to use

This is toolkit - set of classes/methods for OTServ database. It abstracts database mechanisms for you so you can work on "physical" PHP objects. But you must know how to use them. This documentation describes some basic steps and toolkit API, but you must know PHP in order to make use of them - the best place to get some knowledge is [PHP manual](#).

Don't copy any of included examples, neither codes provided as examples - they probably won't work. Mainly it's because you have to put your database configuration into them and your script paths. But it's not enough. If you have your own `__autoload()` mechanism you won't be able to just include example codes - you would need to redefine `__autoload()` function, which PHP doesn't allow to (but you should know that very well). Example codes are examples - write your own (if you want them to work the best way for you).

Link

If you use POT in your script and want to show that you can put this image on your website:

You can use following code for that:

```
1 <a href="http://otserv-aac.info/" >
2 
3 </a>
```

PHP 5.0

Some things that you should know if you use POT under PHP 5.0.x.

PHP 5.0

PHP5 was a huge step in PHP history. It is completely other language than PHP4 (and older versions). POT is written for PHP5 but currently most PHP5 installations are done with PHP 5.1 and higher versions. PHP 5.0 differs from next versions in few details (or even not details, but huge changes, but those mostly doesn't affect POT). There are some important things you should know if you use POT with PHP 5.0.

PDO

POT requires [PDO extension](#). It is bundled with PHP since 5.1 version. If you use PHP 5.0 you still can install PDO, but you need to do that using [PECL extensions](#). Detailed information about how to do that are in [PHP manual PDO page](#).

Sub package "compat"

If you use PHP 5.0 you should include special [compatibility assurance library](#). POT uses some mechanisms that exists since PHP 5.1 like [Countable interface](#). It doesn't disallow you using POT with PHP 5.0. Compatibility library will create unexisting interfaces, classes, functions, constants etc. However keep in mind that you won't be able to use PHP 5.1 and newer language mechanisms as it is not possible to redefine PHP behaviour. Here is an example:

```
1  <?php
2
3  /**
4   * @ignore
5   * @package examples
6   * @author Wrzasq <wrzasq@gmail.com>
7   * @copyright 2007 (C) by Wrzasq
8   * @license http://www.gnu.org/licenses/lgpl-3.0.txt GNU Lesser General Public License, Version 3
9   */
10
11  // do that before any POT operations!
12  include('compat.php');
13
14  // to not repeat all that stuff
15  include('quickstart.php');
16
17  // STEP 1: no error here - even though we loaded class that implements Countable interface which does not
18  // exists in PHP 5.0 SPL library, because 'compat' library defines it.
19  $list = new OTS_Players_List();
20
21  // STEP 2: we can do that in every version - count() is in fact just a public method
22  echo $list->count();
23
24  // STEP 3: it won't work correctly in PHP 5.0 - PHP won't call internal count() method of object, will print trivial
25  // count() evaluation result on object
26  echo count($list);
```

25
26 ?>

Nothin new

Compatibility library makes you sure, that POT scripts won't cause FATAL errors if you run them on older versions of PHP. However it doesn't introduce any new mechanisms so you won't find anything new in this package. It is safe to include compat.php file even if you work with PHP version 5.1 or newer, but there is no point in doing that.

`__autoload()`

POT registers own `__autoload()` handler with [spl_autoload_register\(\)](#). This function exists since PHP 5.1.2. Compatibility library defines this function as definer of another function - ordinary `__autoload()`. If you have own `__autoload()` function, compat's `spl_autoload_register()` won't redefine `__autoload()` to avoid `E_ERROR`. You then need to bind [POT::loadClass\(\) method](#) to your `__autoload()` function manually.

What about older PHP versions?

No way. POT was written using new PHP5 object engine - you cant use it with PHP4 and older versions of PHP, PHP/FI.

POT class preview

Here main POT class will be described in more guided way.

What it is

[POT](#) class is main class of this toolkit. You will access any other classes using this one. It creates for you instances of other classes when you call it's methods and handles class files loading.

Creating instance of POT class

To get POT object you have to use [POT::getInstance\(\)](#) static method. You should never ever create POT class instances directly! [POT::getInstance\(\)](#) will save static instance and return it globally so you won't need to re-create instances of this class. It is important, as object of this class contains another resources like database connection, or classes directory path so after creating new instance it would not contain them from previous one.

[__autoload\(\)](#) and POT classes

PHP5 provides nice [autoloading mechanism](#). POT makes use of [spl_autoload_register\(\) function](#) to bind own mechanism with it automatically. If you have your own [__autoload](#) function defined, after including POT class you have to register your function with [spl_autoload_register\(\)](#) as well.

DAO classes

Key part of this toolbox are Data Access Objects which provides abstraction layer in PHP for plain database data. You create them via main POT class using [createObject\(\) method](#).

Quick start

Quick start guide.

Putting this all together

To set POT up for using you have to create it's instance and connect to database (it will automatically bind [POT classes loading mechanism](#) to `__autoload()` function. Here is a startup code example:

```
1  <?php
2
3  /**
4   * @ignore
5   * @package examples
6   * @author Wrzasq <wrzasq@gmail.com>
7   * @copyright 2007 (C) by Wrzasq
8   * @license http://www.gnu.org/licenses/lgpl-3.0.txt GNU Lesser General Public License, Version 3
9   */
10
11 // binds your __autoload code
12 if( function_exists('__autoload') )
13 {
14     spl_autoload_register('__autoload');
15 }
16
17 // includes POT main file
18 include( './classes/OTS.php' );
19
20 // database configuration - can be simply moved to external file, eg. config.php
21 $config= array(
22     'driver' =>  POT::DB_MYSQL,
23     'host' =>    'localhost',
24     'user' =>    'wrzasq',
25     'database' => 'otserv'
26 );
27
28 // creates POT instance (or get existing one)
29 $ots= POT::getInstance();
30 $ots-> connect(null, $config;
31
32 ?>
```

Account creation

It is very simple to create account with POT. Here is example code that is self-explainable:

```
1  <?php
2
3  /**
4   * @ignore
5   * @package examples
6   * @author Wrzasq <wrzasq@gmail.com>
7   * @copyright 2007 (C) by Wrzasq
8   * @license http://www.gnu.org/licenses/lgpl-3.0.txt GNU Lesser General Public License, Version 3
```

```

9  */
10
11 // to not repeat all that stuff
12 include('quickstart.php');
13
14 // creates new OTS_Account object
15 $account= new OTS_Account();
16
17 // generates new account number
18 $number= $account->create();
19
20 /*
21 to generate number from 111111 to 999999 use:
22 $number = $account->create(111111, 999999);
23 */
24
25 // sets account info
26 $account-> setPassword('secret');// $account->setPassword( md5('secret') );
27 $account-> setEmail('foo@example.com');
28 $account-> unblock();// remember to unblock!
29 $account-> setPACCDays(0);
30 $account-> save();
31
32 // give user his number
33 echo 'Your account number is: ', $number
34
35 ?>

```

It is important to remember that [create\(\) method](#) sets `blocked` field of record to true by default, so for smaller projects where you, for example, wouldn't need e-mail activation unblock it after creation.

Character reading

Here comes also simple example for character search:

```

1  <?php
2
3  /**
4   * @ignore
5   * @package examples
6   * @author Wrzasq <wrzasq@gmail.com>
7   * @copyright 2007 (C) by Wrzasq
8   * @license http://www.gnu.org/licenses/lgpl-3.0.txt GNU Lesser General Public License, Version 3
9   */
10
11 // to not repeat all that stuff
12 include('quickstart.php');
13
14 // creates new OTS_Player object
15 $player= new OTS_Player();
16
17 // loads player
18 $player-> find('Wrzasq');
19
20 // checks if player exists
21 if( $player-> isLoading() )
22 {
23     // prints character info

```

```

24     echo 'Player \'' . $player> getName() . '\' has ' . $player> getLevel() . ' level.', "\n"
25
26     // example of associated objects retrieving
27     echo 'Player \'' . $player> getName() . '\' is member of ' . $player> getGroup()-> getName() . '
group.', "\n"
28 }
29 else
30 {
31     echo 'Player does not exists.', "\n"
32 }
33
34 ?>

```

Objects listings

There are also classes for entire sets of records. For each of row classes there is list class. Throught list object you can read single objects and/or delete them from database. Also you can set limitation (for example for pagination). All list classes implements Countable and Iterator interfaces:

```

1  <?php
2
3  /**
4   * @ignore
5   * @package examples
6   * @author Wrzasq <wrzasq@gmail.com>
7   * @copyright 2007 (C) by Wrzasq
8   * @license http://www.gnu.org/licenses/lgpl-3.0.txt GNU Lesser General Public License, Version 3
9   */
10
11 // to not repeat all that stuff
12 include('quickstart.php');
13
14 // creates new OTS_Player object
15 $players= new OTS_Players_List();
16
17 // count of all players - Countable interface implemented
18 echo 'There are ' . count( $players) . ' players in our database.', "\n"
19
20 // sets limitation
21 $players> setLimit(10);
22 $players> setOffset(2);
23
24 // iterates through selected players
25 foreach($playersas $index=> $player)
26 {
27     // each returned item is instance of OTS_Player class
28     echo (2 + $index) . ': ' . $player> getName(), "\n"
29 }
30
31 ?>

```

DAO objects

Main part of POT are Data Access Objects objects

What are DAO objects?

DAO stands for Data Access Objects. Those are objects which you use mostly - players, accounts, groups, objects lists. They use database resource to fetch/store data and provides you programming interface to access that data without using additional languages like SQL, or XML.

Why this way?

PHP is a PHP. When you write a code in PHP each element has a meaning. While using SQL you have to use database queries. In code they are simply a strings which doesn't represent any particular data for programming environment. DAO objects wraps database operations in objective aspect, so "dead" string queries becomes a fully functional objects which you can control more strictly, allows you to assign relations and automate some parts.

Basic operations

Most basic operations are loading, editing and saving data. To see examples of this, see [Quick start guide](#).

Lists objects

For each table there exist single object class and objects list class. List classes implements [Iterator interface](#) so to list their's content you must use [foreach\(\) loop](#). Each element returned for this loop will be instance of single DAO object. You also use lists to delete items.

Custom fields

POT was created for basic SVN database structure. However you can access custom fields with POT. You do that with `getCustomField()` and `setCustomField()` methods of DAO objects (single, not lists).

While accessing custom fields you have to remember about using proper PHP types of passed values. POT doesn't know anything about those fields so it uses value type to check the way it should serve it for a query. Don't worry about safety - it doesn't create any hole for SQL injections. But you must remember, that 1 (integer) is not same as '1' (string), or 1.0 (float). POT will quote strings to fit SQL query and to prevent from SQL injections so make sure you [cast](#) your values to type that represents field type to prevent (mainly) from quoting numeric fields.

You should use those methods only to access custom fields that are not accessible through standard POT API. Those methods executes SQL query each time you call them so it would be a huge effectivity loss to access standard fields with `getCustomField()/setCustomField()`.

Also it is important that in difference to fields accessible with standard setters you can set custom field value

on not loaded/saved object. You must either load object from database, or save standard record before using custom fields as they need record primary key assigned to object for queries. Here is an example:

```
1  <?php
2
3  /**
4   * @ignore
5   * @package examples
6   * @author Wrzasq <wrzasq@gmail.com>
7   * @copyright 2007 (C) by Wrzasq
8   * @license http://www.gnu.org/licenses/lgpl-3.0.txt GNU Lesser General Public License, Version 3
9   */
10
11 // to not repeat all that stuff
12 include('quickstart.php');
13
14 // creates new OTS_Player object
15 $player= new OTS_Player();
16
17 // sets basic fields
18 $player->setName('Wrzasq');
19 $player->setSex(POT::SEX_MALE);
20 $player->setVocation(POT::VOCATION_KNIGHT);
21 /* etc... */
22
23 /*
24  this is bad! we can't call this now as we dont have object ID assinged yet
25
26  $player->setCustomField('my_field', 2);
27
28  must save before that to get automatic ID:
29  */
30 $player->save();
31
32 // now we can call that:
33 // 2 won't be quoted - it's integer
34 $player->setCustomField('my_field', 2);
35 // 3 will be quoted - '3' is a string!
36 $player->setCustomField('another_field', '3');
37
38 ?>
```

Player items

POT provides also objective way of browsing/editing player items (body slots and depot items with all containers). You have [OTS_Item](#) and [OTS_Container](#) classes for that. OTS_Item represents single item, OTS_Container can contain sub-items (either OTS_Item objects, or next level OTS_Container objects).

There is important thing to mention - POT doesn't know anything about item types! Items tree only contains item IDs from database, it doesn't load any information from items.otb, nor items.xml files.

Detailed API you will find in documentation of those classes. Here are examples of how you use slot and depot items fetching and saving:

```
1  <?php
2
3  /**
4   * @ignore
```

```

5  * @package examples
6  * @author Wrzasq <wrzasq@gmail.com>
7  * @copyright 2007 (C) by Wrzasq
8  * @license http://www.gnu.org/licenses/lgpl-3.0.txt GNU Lesser General Public License, Version 3
9  */
10
11 // to not repeat all that stuff
12 include('quickstart.php');
13
14 // loads item typing information
15 $ots-> getInstance()-> loadItems('/path/to/your/ots/data/items');
16
17 // creates new OTS_Player object
18 $player= new OTS_Player();
19 $player-> find('Wrzasq');
20
21 /*
22  Items loading example.
23 */
24
25 // loading item from ammunition slot
26 $item= $player-> getSlot(POT::SLOT_AMMO);
27
28 echo $player-> getName(), ' has item with id ', $item-> getId(), ' in his/her ammo slot.', "\n" ;
29
30 // checks if item is a container
31 if($item instanceof OTS_Container)
32 {
33     // list backpack content
34     foreach($item as $inside)
35     {
36         echo 'Container contains item with id ', $inside-> getId(), ' ', "\n" ;
37     }
38 }
39
40 /*
41  Items tree composing example.
42 */
43
44 // creates container - here it would be a depot locker (we pass ID of item to create)
45 $container= new OTS_Container(2590);
46
47 // now let's create depot chest
48 $chest= new OTS_Container(2594);
49
50 // let's put chest inside locker
51 $container-> addItem($chest);
52
53 // now let's put something deeper - into the chest
54 $item1= new OTS_Item(3015);
55 $chest-> addItem($item1);
56
57 // and more...
58 $item2= new OTS_Item(3013);
59 $chest-> addItem($item2);
60
61 // let's set count for an item
62 $item2-> setCount(2);
63

```

```

64  /*
65  Here is a tree of items which we created:
66
67  $container [depot locker]
68  |-- $chest [depot chest]
69      |-- $item1 [first item inserted into chest]
70      |-- $item2 [second item inserted into chest] count=2
71  */
72
73  /*
74  Items saving example.
75  */
76
77  // now we simply put those items into players depot (2 is depot ID)
78  $player-> setDepot(2, $container);
79
80  ?>

```

Important thing - OTS_Container class is subclass of OTS_Item. Each container is also an item.

Guilds

Guilds system basics.

Baiscs

Like for most other data types, for guilds and ranks there are two kinds of classes - single object class and list class. For guilds those are [OTS_Guild](#) and [OTS_Guilds_List](#), for ranks - [OTS_GuildRank](#) and [OTS_GuildRanks_List](#).

Guild management

Listing guilds is simple so there is no need to explain it more. More complex is listing guild members. Guild membership is not assigned directly - it is done throught guild ranks. To list guild members you first need to list it's ranks. Here is an example solution to list members in oryiginal Tibia-like way:

```
1  <?php
2
3  /**
4   * @ignore
5   * @package examples
6   * @author Wrzasq <wrzasq@gmail.com>
7   * @copyright 2007 (C) by Wrzasq
8   * @license http://www.gnu.org/licenses/lgpl-3.0.txt GNU Lesser General Public License, Version 3
9   */
10
11 // to not repeat all that stuff
12 include('quickstart.php');
13
14 // loads guild
15 $guild= new OTS_Guild();
16 $guild->load(1);
17
18 $color= '#FFFFCC';
19
20 echo '<h1>Members of ' . htmlspecialchars( $guild->getName() ), '</h1>' ;
21
22 ?>
23 <table>
24     <thead>
25         <tr>
26             <th>Rank</th>
27             <th>Members</th>
28         </tr>
29     </thead>
30     <tbody>
31 <?php
32
33 // lists members of all ranks
34 foreach($guildas $guildRank)
35 {
36     // display rank in first row
37     $first= true;
```

```

38 // switches rank rows color
39 $color= $color== '#FFFFCC' ? '#FFCCFF' : '#FFFFCC';
40
41 // list members of this rank
42 foreach($guildRankas $player)
43 {
44     echo '<tr style="background-color: ' . $color
45     <td>' . $first?htmlspecialchars( $guildRank> getName() ) : ", '</td>
46     <td>' . $player> getName(), '</td>
47 </tr>' ;
48     $first= false;
49 }
50 }
51
52 ?>
53 </tbody>
54 </table>

```

Guild action drivers

Handling invites/requests system for guilds.

How does it work?

OTServ database contains all guilds contents. But it is very common in AAC world to create invites system (or also requests system, but invitations are more common). It is not provided by standard OTServ database, though nearly all AAC scripts contains such mechanisms. POT classes allows you to set own drivers for invitations and requests to extend basic OTS functionality.

You have to write a driver class and assign it's object to guild object - then guild object will call requested actions on driver which will execute action code dependent on your script.

Driver structure

Both invites and requests drivers are similar - they must implement [IOTS_GuildAction interface](#). When the driver is assigned to guild object, each time a method of [OTS_Guild](#) object is called, it will forward this to action driver.

Sample driver

Driver implements your logic for invites (or membership requests). Here is sample code that you can base on:

```
1  <?php
2
3  /**
4   * @ignore
5   * @package examples
6   * @author Wrzasq <wrzasq@gmail.com>
7   * @copyright 2007 (C) by Wrzasq
8   * @license http://www.gnu.org/licenses/lgpl-3.0.txt GNU Lesser General Public License, Version 3
9   */
10
11 // to not repeat all that stuff
12 include('quickstart.php');
13
14 /**
15  POT guilds invites driver.
16  */
17
18 /**
19  * @ignore
20  */
21 class InvitesDriver implements IOTS_GuildAction
22 {
23     // assigned guild
24     private $guild;
25
26     // initializes driver
27     public function __construct(OTS_Guild $guild)
28     {
```

```

29     $this>    guild= $guild
30     // this line automates the process - you can call it manually from outside, but why?
31     $this>    guild>    setInvitesDriver$this;
32 }
33
34 // returns all invited players to current guild
35 public functionlistRequests()
36 {
37     $invites= array();
38
39     /* here you must create OTS_Player object for each invited player */
40
41     return$invites
42 }
43
44 // invites player to current guild
45 public functionaddRequest(OTS_Player $player)
46 {
47     /* here you must save invitation for given player */
48 }
49
50 // un-invites player
51 public functiondeleteRequest(OTS_Player $player)
52 {
53     /* here you must delete invitation for given player */
54 }
55
56 // commits invitation
57 public functionsubmitRequest(OTS_Player $player)
58 {
59     $rank= null;
60
61     // finds normal member rank
62     foreach( $this>    guild>    getGuildRanks($s $guildRank)
63     {
64         if( $guildRank>    getLevel() == 1)
65         {
66             $rank= $guildRank
67             break;
68         }
69     }
70
71     $player>    setRank($rank);
72     $player>    save();
73
74     // clears invitation
75     $this>    deleteRequest($player);
76 }
77 }
78
79 /*
80 Parts of this class driver has been taken from OTSCMS (http://otscms.sourceforge.net/) project source
81 code.
82 */
83 // loads player wiht ID 1
84 $player= new OTS_Player();
85 $player>    load(1);
86

```

```
87 // loads guild with ID 1
88 $guild= new OTS_Guild();
89 $guild-> load(1);
90
91 // creates invitation logic driver for your implementation for current guild
92 new InvitesDriver( $guild);
93
94 // note that you call guild method!
95 $guild-> invite($player);
96
97 ?>
```


Account number hack

Example code of how to use prepared account number instead of random.

Walkaround

POT always generates random account number - [it is the way your script should work](#). It is done that way with premeditation. However you can walk around it with simple code:

```
1  <?php
2
3  /**
4   * @ignore
5   * @package examples
6   * @author Wrzasq <wrzasq@gmail.com>
7   * @copyright 2007 (C) by Wrzasq
8   * @license http://www.gnu.org/licenses/lgpl-3.0.txt GNU Lesser General Public License, Version 3
9   */
10
11 // to not repeat all that stuff
12 include('quickstart.php');
13
14 // your non-random number
15 $number= 123456;
16
17 // creates new OTS_Account object
18 $account= new OTS_Account();
19 $account->load($number);
20
21 // number is busy
22 if( $account->isLoaded() )
23 {
24     echo 'Account number ', $number, 'is used.', "\n" ;
25 }
26 // it is not
27 else
28 {
29     // generate number from exactly $number - $number range
30     $number= $account->create($number, $number);
31     echo 'Your account number is: ', $number, "\n" ;
32 }
33
34 ?>
```

Server online status

This tutorial will describe how to test server status with POT.

Such a simple way

[POT class](#) contains [serverStatus\(\) method](#) which sends 'info' packet to OTS and handles results. It returns object of class [OTS_InfoRespond](#) which provides access methods for all OTServ respond info. It will return false if server is offline. Here is a simple example of this method usage:

```
1  <?php
2
3  /**
4   * @ignore
5   * @package examples
6   * @author Wrzasq <wrzasq@gmail.com>
7   * @copyright 2007 (C) by Wrzasq
8   * @license http://www.gnu.org/licenses/lgpl-3.0.txt GNU Lesser General Public License, Version 3
9   */
10
11 // to not repeat all that stuff
12 include('quickstart.php');
13
14 // server and port
15 $server= '127.0.0.1';
16 $port= 7171;
17
18 // queries server of status info
19 $status= $ots-> serverStatus($server, $port);
20
21 // offline
22 if(!$status)
23 {
24     echo 'Server ', $server, ' is offline.', "\n" ;
25 }
26 // displays various info
27 else
28 {
29     echo 'Server name: ', $status-> getName(), "\n" ;
30     echo 'Server owner: ', $status-> getOwner(), "\n" ;
31     echo 'Players online: ', $status-> getOnlinePlayers(), "\n" ;
32     echo 'Maximum allowed number of players: ', $status-> getMaxPlayers(), "\n" ;
33     echo 'Required client version: ', $status-> getClientVersion(), "\n" ;
34     echo 'All monsters: ', $status-> getMonstersCount(), "\n" ;
35     echo 'Server message: ', $status-> getMOTD(), "\n" ;
36 }
37
38 ?>
```

DOM way

In case you would want to use this method for some non-SVN server which contains custom fields in respond packet you can still use it. OTS_InfoRespond class is child of DOMDocument class and doesn't overwrite it's

interface neither behaviour in any way. Returned object is standard DOM document so you can work with it in standard DOM-way.

About OTServ AAC scripts

This small article describes general info about OTServ AAC scripts.

Basics

Welcome! On this website you will find info about OTServ accmakers. This website is dedicated both for people who don't know anything about that and authors of such scripts. Beginners will find here basics and clues about how to use such scripts, as well as ready solutions for their's AAC. AAC creators should follow instructions on this site in order to make those scripts well.

Many people want to start using AAC scripts without knowledge. They spam forums, IRC channels and people IMs. If you don't know anything about AAC scripts, then this site is perfect for you. Before you will ask any question, read this website. If you will still don't know the answer, think before ask.

Mainly last times there went out plenty new AAC scripts. Too bad they are very poor and people who make them don't know anything about their's job. People started to think that if they made "own AAC" (which usually means to copy other script and sign with own nickname) they are cool - sorry guys - you only show how stupid your codes are. This website provides information about how to make good scripts and promotes ready solutions for safe and stable websites. We hope this website will change the situation and people who are creating accmakers will correct their's works or leave publishing bad scripts.

What is AAC?

AAC stands for Automatic Account Creator also called accmaker. Most generally it is a program (application, or script) that automates account creation process. However for a long time already simple accmakers aren't enough - nowadays average AAC should have additional options like account management, statistics and character lookup.

Types of AAC

Basically there two types of accmakers: websites and in-game. In-game AACs are NPCs that ask user for account and character information. To use such AAC person must log into special account (usually 1/1). Website AACs provides much more features - you can browse web from every place and from many devices. You can access it globally. Also those accmakers aren't restricted by Tibia client and can be extended in many ways. Usually website accmakers are PHP scripts and works on various HTTP servers.

Why not ingame

First when people of OTS world weren't familiar with PHP, HTTP servers they were just adding some code to server and account used to be created after logging in on special password where was NPC to complete the process. As OTS community were extending and new ideas came out, there appeared first website scripts - they provided at least so much functionality as NPC accmaker. With time website AACs were extending and now they usually contains many features that NPC would never have. Website AAC is accessible from every device where you have the Internet and browser so it means nearly every computer all over the world in this days. But those are all advantages of website AAC - there is one more reason which simply disqualifies in-game AACs: they are in fact impossible. Why? It is possible to create such AAC only if you add your server to many lists and links. To use in-game AAC people need to know your IP to connect and have Tibia client to use AAC. Normally they wouldn't know

that and Tibia client is not a standard application that is installed in every computer. The only way to provide accessibility for users is a website.

Website AAC HOWTO

Website AAC is most commonly used type of accmaker. People who want to use it first time find it hard to install and maintain. In fact it can be - you really need to know what you are doing with it and how does it work. Usually this type of AAC is a PHP script so we won't discuss other cases. To run PHP script you need a HTTP server - program which will provide website for people from outside, with installed PHP - interpreter of PHP scripts that executes them. It is quite easy to install Apache and PHP manually, but it is described all over the net, so we won't describe it here.

Main features

Of course basic AAC script feature, as the name says, must be account creation. But from the time when the first website accmaker was made (about 2004) scripts of that kind were extended and now "just AAC" is never enough. Empty site with only form for account creation shows that server is poor, that administrator doesn't care about it (and users) and he is probably a noob that just wanted to have "my own masta OTS". Currently even simple accmaker must provide some basic features.

Account creation is of course the most basic AAC feature. But this is also the point which is made wrong in nearly every script. Account number has to be random and generated during account creation, not during entering website by user. That's most important criterium which we used from our recommended scripts. This is for safety reasons, but not only. It is simply only possible way of correct implementation - people who make it other way simply don't know what they are doing as this is very unstable realisation.

On the beginning first accmaker was just a website form for creating account and character. But someone who made script was just lazy and finished work in that point. First more extended script was OTSCMS which was first that introduced login mechanism and allowed users to manage account from website. It means that they could create many characters on one account, or for example change password. Currently accmakers with only account creation form are not even worth to downloading (except PVP servers) and account managers are now standard.

Other important features of website AAC scripts are ideas based on [original Tibia](#) website. Many script contains statistics page where are listed players with highest scores, character view page where it is possible to check information about given player. Also lastly guilds system is very popular. Very important is, to merge website with OTSserv world and create some kind of community, it means that on website player should be affected, or at least connected with character in game.

Also very important is, to provide easy way of changing website behaviors and/or layout without editing script engine. Some scripts contain template engines, multilanguage support and modular structure. All those things give user ability to make website to look just like he wants. Everyone wants to have original website which will impress visitor.

Don't touch!

Before using, every AAC needs to be installed. It requires to put information about OTSserv as it needs to work on its database. Many dumb people create scripts and just put configuration file there so people will edit it. Too bad they don't even know about how to distribute PHP scripts, so how average user should know it? Script is a code - user mustn't touch the code. Code is a hermetic environment - when user will edit it and type something wrong it will crash. Every PHP script that needs to be edited in any way, includes installer, or just configuration editor. With

such script user fills settings on website and installer checks and validates them and then creates configuration file with saved settings.

Important notes

This website presents some ways of AAC script development that should be followed. It is not just our wish, but we presents clever and considered ideas. Those are just real points of view.

We want to promote "good scripts". If you have a script that fits our requirements you can contact us - we will add it to list of our recommended scripts.

Ready scripts

Here are links to some major AAC scripts:

- [OTSCMS](#) - uses POT.
- [SmartAss](#).
- [Nicaw CMS](#).
- [TauAccmaker](#).

Link



If you want to help us you can put following image on your website:

You can use following code for that:

```
1 <a href="http://otserv-aac.info/" >
2 
3 </a>
```


Package POT Procedural Elements

E_OTS_ErrorCode.php

- **Package** POT
- **Author** Wrzasq < wrzasq@gmail.com>
- **Version** 0.1.2
- **Copyright** 2007 (C) by Wrzasq
- **Since** 0.0.6
- **License** [GNU Lesser General Public License, Version 3](#)

E_OTS_FileLoaderError.php

Code in this file bases on oryiginal OTServ binary format loading C++ code (fileloader.

Code in this file bases on oryiginal OTServ binary format loading C++ code (fileloader.h, fileloader.cpp).

- **Package** POT
- **Author** Wrzasq < wrzasq@gmail.com>
- **Version** 0.0.6
- **Copyright** 2007 (C) by Wrzasq
- **Since** 0.0.6
- **License** [GNU Lesser General Public License, Version 3](#)

E_OTS_Generic.php

- **Package** POT
- **Author** Wrzasq < wrzasq@gmail.com>
- **Version** 0.1.1
- **Copyright** 2007 (C) by Wrzasq
- **Since** 0.1.1
- **License** [GNU Lesser General Public License, Version 3](#)

E_OTS_NoDriver.php

- **Package** POT
- **Author** Wrzasq < wrzasq@gmail.com>
- **Version** 0.0.4
- **Copyright** 2007 (C) by Wrzasq
- **Since** 0.0.4
- **License** [GNU Lesser General Public License, Version 3](#)

E_OTS_NotAContainer.php

- **Package** POT
- **Author** Wrzasq < wrzasq@gmail.com>
- **Version** 0.1.0
- **Copyright** 2007 (C) by Wrzasq
- **Since** 0.1.0
- **License** [GNU Lesser General Public License, Version 3](#)

E_OTS_NotLoaded.php

- **Package** POT
- **Author** Wrzasq < wrzasq@gmail.com>
- **Version** 0.0.3
- **Copyright** 2007 (C) by Wrzasq
- **Since** 0.0.3
- **License** [GNU Lesser General Public License, Version 3](#)

E_OTS_OTBMErrror.php

Code in this file bases on oryiginal OTServ OTBM format loading C++ code (iomapotbm.

Code in this file bases on oryiginal OTServ OTBM format loading C++ code (iomapotbm.h, iomapotbm.cpp).

- **Package** POT
- **Author** Wrzasq < wrzasq@gmail.com>
- **Version** 0.0.6
- **Copyright** 2007 (C) by Wrzasq
- **Since** 0.0.6
- **License** [GNU Lesser General Public License, Version 3](#)

E_OTS_OutOfBuffer.php

Code in this file bases on oryiginal OTServ binary format loading C++ code (fileloader.

Code in this file bases on oryiginal OTServ binary format loading C++ code (fileloader.h, fileloader.cpp).

- **Package** POT
- **Author** Wrzasq < wrzasq@gmail.com>
- **Version** 0.0.6
- **Copyright** 2007 (C) by Wrzasq
- **Since** 0.0.6
- **License** [GNU Lesser General Public License, Version 3](#)

E_OTS_ReadOnly.php

- **Package** POT
- **Author** Wrzasq < wrzasq@gmail.com>
- **Version** 0.1.0
- **Copyright** 2007 (C) by Wrzasq
- **Since** 0.1.0
- **License** [GNU Lesser General Public License, Version 3](#)

IOTS_Cipher.php

- **Package** POT
- **Author** Wrzasq < wrzasq@gmail.com>
- **Version** 0.1.2
- **Copyright** 2007 - 2008 (C) by Wrzasq
- **Since** 0.1.2
- **License** [GNU Lesser General Public License, Version 3](#)

IOTS_DAO.php

- **Package** POT
- **Author** Wrzasq < wrzasq@gmail.com>
- **Version** 0.1.0
- **Copyright** 2007 (C) by Wrzasq
- **Since** 0.0.1
- **License** [GNU Lesser General Public License, Version 3](#)

IOTS_DB.php

- **Package** POT
- **Author** Wrzasq < wrzasq@gmail.com>
- **Version** 0.0.6
- **Version** 0.0.1
- **Copyright** 2007 (C) by Wrzasq
- **Since** 0.0.1
- **License** [GNU Lesser General Public License, Version 3](#)

IOTS_Display.php

- **Package** POT
- **Author** Wrzasq < wrzasq@gmail.com>
- **Version** 0.1.0
- **Copyright** 2007 (C) by Wrzasq
- **Since** 0.1.0
- **License** [GNU Lesser General Public License, Version 3](#)

IOTS_FileCache.php

- **Package** POT
- **Author** Wrzasq < wrzasq@gmail.com>
- **Version** 0.0.6
- **Copyright** 2007 (C) by Wrzasq
- **Since** 0.0.6
- **License** [GNU Lesser General Public License, Version 3](#)

IOTS_GuildAction.php

- **Package** POT
- **Author** Wrzasq < wrzasq@gmail.com>
- **Version** 0.0.4
- **Copyright** 2007 (C) by Wrzasq
- **Since** 0.0.4
- **License** [GNU Lesser General Public License, Version 3](#)

IOTS_ItemsCache.php

- **Package** POT
- **Author** Wrzasq < wrzasq@gmail.com>
- **Version** 0.0.8
- **Copyright** 2007 (C) by Wrzasq
- **Since** 0.0.8
- **License** [GNU Lesser General Public License, Version 3](#)

OTS.php

This file contains main toolkit class.

This file contains main toolkit class. Please read README file for quick startup guide and/or tutorials for more info.

- **Package** POT
- **Author** Wrzasq < wrzasq@gmail.com>
- **Version** 0.1.2
- **Version** 0.0.1
- **Copyright** 2007 - 2008 (C) by Wrzasq
- **TODO** 1.0.0: Complete phpUnit test.
- **TODO** 1.0.0: More detailed documentation and tutorials, also update examples and tutorials.
- **TODO** 1.0.0: Implement OutOfBoundsException instead of mixed results types.
- **TODO** 1.0.0: Deprecations cleanup.
- **TODO** 1.0.0: Unify *List and *_List naming (probably into *List).
- **TODO** 1.0.0: Main POT class as database instance.
- **TODO** 2.0.0: Code as C++ extension (as an alternative to pure PHP library which of course would still be available).
- **TODO** 1.0.0: E_* classes into *Exception, IOTS* into *Interface, change POT classes prefix from OTS_* into OT_*.
- **TODO** ?: Implement POT namespace when it will be supported by PHP.
- **TODO** 1.0.0: PHAR and PHK packages.
- **Since** 0.0.1
- **License** [GNU Lesser General Public License, Version 3](#)

OTS_Account.php

- **Package** POT
- **Author** Wrzasq < wrzasq@gmail.com>
- **Version** 0.1.1
- **Version** 0.0.1
- **Copyright** 2007 - 2008 (C) by Wrzasq
- **Since** 0.0.1
- **License** [GNU Lesser General Public License, Version 3](#)

OTS_Accounts_List.php

- **Package** POT
- **Author** Wrzasq < wrzasq@gmail.com>
- **Version** 0.1.0
- **Copyright** 2007 (C) by Wrzasq
- **Since** 0.0.1
- **License** [GNU Lesser General Public License, Version 3](#)

OTS_Admin.php

- **Package** POT
- **Author** Wrzasq < wrzasq@gmail.com>
- **Version** 0.1.2
- **Copyright** 2007 (C) by Wrzasq
- **Since** 0.1.2
- **License** [GNU Lesser General Public License, Version 3](#)

OTS_Base_DAO.php

- **Package** POT
- **Author** Wrzasq < wrzasq@gmail.com >
- **Version** 0.1.0
- **Version** 0.0.5
- **Copyright** 2007 (C) by Wrzasq
- **Since** 0.0.5
- **License** [GNU Lesser General Public License, Version 3](#)

OTS_Base_List.php

- **Package** POT
- **Author** Wrzasq < wrzasq@gmail.com>
- **Version** 0.1.0
- **Version** 0.0.5
- **Copyright** 2007 (C) by Wrzasq
- **Since** 0.0.5
- **License** [GNU Lesser General Public License, Version 3](#)

OTS_BinaryTools.php

- **Package** POT
- **Author** Wrzasq < wrzasq@gmail.com>
- **Version** 0.1.2
- **Copyright** 2007 - 2008 (C) by Wrzasq
- **Since** 0.1.2
- **License** [GNU Lesser General Public License, Version 3](#)

OTS_Buffer.php

This is generic class for classes that uses buffer-baser read-write operations (it can also emulate C-like pointers).

This is generic class for classes that uses buffer-baser read-write operations (it can also emulate C-like pointers).

- **Package** POT
- **Author** Wrzasq < wrzasq@gmail.com>
- **Version** 0.1.2
- **Copyright** 2007 - 2008 (C) by Wrzasq
- **Since** 0.1.2
- **License** [GNU Lesser General Public License, Version 3](#)

OTS_Container.php

- **Package** POT
- **Author** Wrzasq < wrzasq@gmail.com>
- **Version** 0.1.0
- **Version** 0.0.3
- **Copyright** 2007 (C) by Wrzasq
- **Since** 0.0.3
- **License** [GNU Lesser General Public License, Version 3](#)

OTS_DB_MySQL.php

- **Package** POT
- **Author** Wrzasq < wrzasq@gmail.com >
- **Version** 0.0.6
- **Version** 0.0.1
- **Copyright** 2007 (C) by Wrzasq
- **Since** 0.0.1
- **License** [GNU Lesser General Public License, Version 3](#)

OTS_DB_ODBC.php

- **Package** POT
- **Author** Wrzasq < wrzasq@gmail.com >
- **Version** 0.0.6
- **Version** 0.0.4
- **Copyright** 2007 (C) by Wrzasq
- **Since** 0.0.4
- **License** [GNU Lesser General Public License, Version 3](#)

OTS_DB_PostgreSQL.php

- **Package** POT
- **Author** Wrzasq < wrzasq@gmail.com >
- **Version** 0.0.6
- **Version** 0.0.4
- **Copyright** 2007 (C) by Wrzasq
- **Since** 0.0.4
- **License** [GNU Lesser General Public License, Version 3](#)

OTS_DB_SQLite.php

- **Package** POT
- **Author** Wrzasq < wrzasq@gmail.com >
- **Version** 0.0.6
- **Version** 0.0.1
- **Copyright** 2007 (C) by Wrzasq
- **Since** 0.0.1
- **License** [GNU Lesser General Public License, Version 3](#)

OTS_FileLoader.php

Code in this file bases on oryiginal OTServ binary format loading C++ code (fileloader.

Code in this file bases on oryiginal OTServ binary format loading C++ code (fileloader.h, fileloader.cpp).

- **Package** POT
- **Author** Wrzasq < wrzasq@gmail.com>
- **Version** 0.1.0
- **Version** 0.0.6
- **Copyright** 2007 - 2008 (C) by Wrzasq
- **Since** 0.0.6
- **License** [GNU Lesser General Public License, Version 3](#)

OTS_FileNode.php

Code in this file bases on oryiginal OTServ binary format loading C++ code (fileloader.

Code in this file bases on oryiginal OTServ binary format loading C++ code (fileloader.h, fileloader.cpp).

- **Package** POT
- **Author** Wrzasq < wrzasq@gmail.com>
- **Version** 0.1.2
- **Version** 0.0.6
- **Copyright** 2007 - 2008 (C) by Wrzasq
- **Since** 0.0.6
- **License** [GNU Lesser General Public License, Version 3](#)

OTS_Group.php

- **Package** POT
- **Author** Wrzasq < wrzasq@gmail.com>
- **Version** 0.1.1
- **Version** 0.0.1
- **Copyright** 2007 - 2008 (C) by Wrzasq
- **Since** 0.0.1
- **License** [GNU Lesser General Public License, Version 3](#)

OTS_Groups_List.php

- **Package** POT
- **Author** Wrzasq < wrzasq@gmail.com >
- **Version** 0.1.0
- **Copyright** 2007 (C) by Wrzasq
- **Since** 0.0.1
- **License** [GNU Lesser General Public License, Version 3](#)

OTS_Guild.php

- **Package** POT
- **Author** Wrzasq < wrzasq@gmail.com>
- **Version** 0.1.1
- **Version** 0.0.4
- **Copyright** 2007 - 2008 (C) by Wrzasq
- **Since** 0.0.4
- **License** [GNU Lesser General Public License, Version 3](#)

OTS_GuildRank.php

- **Package** POT
- **Author** Wrzasq < wrzasq@gmail.com>
- **Version** 0.1.1
- **Version** 0.0.4
- **Copyright** 2007 - 2008 (C) by Wrzasq
- **Since** 0.0.4
- **License** [GNU Lesser General Public License, Version 3](#)

OTS_GuildRanks_List.php

- **Package** POT
- **Author** Wrzasq < wrzasq@gmail.com>
- **Version** 0.1.0
- **Copyright** 2007 (C) by Wrzasq
- **Since** 0.0.4
- **License** [GNU Lesser General Public License, Version 3](#)

OTS_Guilds_List.php

- **Package** POT
- **Author** Wrzasq < wrzasq@gmail.com>
- **Version** 0.1.0
- **Copyright** 2007 (C) by Wrzasq
- **Since** 0.0.4
- **License** [GNU Lesser General Public License, Version 3](#)

OTS_House.php

- **Package** POT
- **Author** Wrzasq < wrzasq@gmail.com>
- **Version** 0.1.0
- **Copyright** 2007 - 2008 (C) by Wrzasq
- **Since** 0.1.0
- **License** [GNU Lesser General Public License, Version 3](#)

OTS_HousesList.php

- **Package** POT
- **Author** Wrzasq < wrzasq@gmail.com>
- **Version** 0.1.0
- **Copyright** 2007 (C) by Wrzasq
- **Since** 0.1.0
- **License** [GNU Lesser General Public License, Version 3](#)

OTS_InfoRespond.php

- **Package** POT
- **Author** Wrzasq < wrzasq@gmail.com >
- **Version** 0.1.0
- **Version** 0.0.2
- **Copyright** 2007 (C) by Wrzasq
- **Since** 0.0.2
- **License** [GNU Lesser General Public License, Version 3](#)

OTS_Item.php

- **Package** POT
- **Author** Wrzasq < wrzasq@gmail.com>
- **Version** 0.1.0
- **Version** 0.0.3
- **Copyright** 2007 - 2008 (C) by Wrzasq
- **Since** 0.0.3
- **License** [GNU Lesser General Public License, Version 3](#)

OTS_ItemsList.php

Code in this file bases on oryiginal OTServ items loading C++ code (itemloader.

Code in this file bases on oryiginal OTServ items loading C++ code (itemloader.h, items.cpp, items.h).

- **Package** POT
- **Author** Wrzasq < wrzasq@gmail.com >
- **Version** 0.1.0
- **Version** 0.0.8
- **Copyright** 2007 (C) by Wrzasq
- **Since** 0.0.8
- **License** [GNU Lesser General Public License, Version 3](#)

OTS_ItemType.php

Code in this file bases on oryiginal OTServ items loading C++ code (items.

Code in this file bases on oryiginal OTServ items loading C++ code (items.cpp, items.h).

- **Package** POT
- **Author** Wrzasq < wrzasq@gmail.com>
- **Version** 0.1.0
- **Version** 0.0.8
- **Copyright** 2007 - 2008 (C) by Wrzasq
- **Since** 0.0.8
- **License** [GNU Lesser General Public License, Version 3](#)

OTS_MapCoords.php

Code in this file bases on oryiginal OTServ OTBM format loading C++ code (iomapotbm.

Code in this file bases on oryiginal OTServ OTBM format loading C++ code (iomapotbm.h, iomapotbm.cpp).

- **Package** POT
- **Author** Wrzasq < wrzasq@gmail.com >
- **Version** 0.1.0
- **Version** 0.0.6
- **Copyright** 2007 (C) by Wrzasq
- **Since** 0.0.6
- **License** [GNU Lesser General Public License, Version 3](#)

OTS_Monster.php

- **Package** POT
- **Author** Wrzasq < wrzasq@gmail.com>
- **Version** 0.1.0
- **Version** 0.0.6
- **Copyright** 2007 (C) by Wrzasq
- **Since** 0.0.6
- **License** [GNU Lesser General Public License, Version 3](#)

OTS_MonstersList.php

- **Package** POT
- **Author** Wrzasq < wrzasq@gmail.com>
- **Version** 0.1.0
- **Copyright** 2007 (C) by Wrzasq
- **Since** 0.1.0
- **License** [GNU Lesser General Public License, Version 3](#)

OTS_OTBMFile.php

Code in this file bases on oryiginal OTServ OTBM format loading C++ code (iomapotbm.

Code in this file bases on oryiginal OTServ OTBM format loading C++ code (iomapotbm.h, iomapotbm.cpp).

- **Package** POT
- **Author** Wrzasq < wrzasq@gmail.com>
- **Version** 0.0.6
- **Version** 0.1.0
- **Copyright** 2007 (C) by Wrzasq
- **TODO** 1.0.0: Spawns support.
- **TODO** 1.0.0: Complete OTBM support: link tiles with items and spawns.
- **Since** 0.0.6
- **License** [GNU Lesser General Public License, Version 3](#)

OTS_Player.php

- **Package** POT
- **Author** Wrzasq < wrzasq@gmail.com>
- **Version** 0.1.2
- **Version** 0.0.1
- **Copyright** 2007 - 2008 (C) by Wrzasq
- **Since** 0.0.1
- **License** [GNU Lesser General Public License, Version 3](#)

OTS_Players_List.php

- **Package** POT
- **Author** Wrzasq < wrzasq@gmail.com>
- **Version** 0.1.0
- **Copyright** 2007 (C) by Wrzasq
- **Since** 0.0.1
- **License** [GNU Lesser General Public License, Version 3](#)

OTS_Row_DAO.php

- **Package** POT
- **Author** Wrzasq < wrzasq@gmail.com>
- **Version** 0.1.1
- **Copyright** 2007 - 2008 (C) by Wrzasq
- **Since** 0.1.1
- **License** [GNU Lesser General Public License, Version 3](#)

OTS_RSA.php

- **Package** POT
- **Author** Alexander Valyalkin < valyala@gmail.com >
- **Author** Wrzasq < wrzasq@gmail.com >
- **Version** 0.1.2
- **Copyright** 2007 - 2008 (C) by Wrzasq
- **Since** 0.1.2
- **License** [GNU Lesser General Public License, Version 3](#)
- **License** [PHP License, Version 3.0](#)

OTS_Spell.php

- **Package** POT
- **Author** Wrzasq < wrzasq@gmail.com>
- **Version** 0.1.0
- **Version** 0.0.7
- **Copyright** 2007 (C) by Wrzasq
- **Since** 0.0.7
- **License** [GNU Lesser General Public License, Version 3](#)

OTS_SpellsList.php

- **Package** POT
- **Author** Wrzasq < wrzasq@gmail.com>
- **Version** 0.1.0
- **Copyright** 2007 (C) by Wrzasq
- **Since** 0.1.0
- **License** [GNU Lesser General Public License, Version 3](#)

OTS_SQLField.php

- **Package** POT
- **Author** Wrzasq < wrzasq@gmail.com>
- **Version** 0.1.0
- **Version** 0.0.5
- **Copyright** 2007 (C) by Wrzasq
- **Since** 0.0.5
- **License** [GNU Lesser General Public License, Version 3](#)

OTS_SQLFilter.php

- **Package** POT
- **Author** Wrzasq < wrzasq@gmail.com >
- **Version** 0.1.0
- **Version** 0.0.5
- **Copyright** 2007 (C) by Wrzasq
- **Since** 0.0.5
- **License** [GNU Lesser General Public License, Version 3](#)

OTS_SQLite_Results.php

- **Package** POT
- **Author** Wrzasq < wrzasq@gmail.com>
- **Version** 0.0.1
- **Copyright** 2007 (C) by Wrzasq
- **Since** 0.0.1
- **License** [GNU Lesser General Public License, Version 3](#)

OTS_Toolbox.php

- **Package** POT
- **Author** Wrzasq < wrzasq@gmail.com>
- **Version** 0.1.1
- **Copyright** 2007 - 2008 (C) by Wrzasq
- **Since** 0.1.1
- **License** [GNU Lesser General Public License, Version 3](#)

OTS_VocationsList.php

- **Package** POT
- **Author** Wrzasq < wrzasq@gmail.com >
- **Version** 0.1.0
- **Copyright** 2007 (C) by Wrzasq
- **Since** 0.1.0
- **License** [GNU Lesser General Public License, Version 3](#)

OTS_XTEA.php

- **Package** POT
- **Author** Jeroen Derks < jeroen@derks.it >
- **Author** Wrzasq < wrzasq@gmail.com >
- **Version** 0.1.2
- **Copyright** 2007 - 2008 (C) by Wrzasq
- **Since** 0.1.2
- **License** [GNU Lesser General Public License, Version 3](#)
- **License** [PHP License, Version 2.02](#)

Package POT Classes

Class E_OTS_ErrorCode

[line 20]

Generic exception class for error codes.

Generic exception class for error codes.

- **Package** POT
- **Version** 0.1.2
- **Since** 0.0.6

Constructor *void* function E_OTS_ErrorCode::__construct(\$code, [\$message = "]) [line 28]

Function Parameters:

- *int* **\$code** Error code.
- *string* **\$message** Optional error message.

Sets error code.

Sets error code.

- **Version** 0.1.2

- **Since** 0.0.6
- **Access** public

Class E_OTS_FileLoaderError

[line 22]

Error during reading OTServ binary file.
Error during reading OTServ binary file.

- **Package** POT
- **Version** 0.0.6
- **Since** 0.0.6

E_OTS_FileLoaderError::ERROR_CAN_NOT_OPEN

= 2 [line 31]

Could not open file.
Could not open file.

- **Version** 0.0.6
- **Since** 0.0.6

E_OTS_FileLoaderError::ERROR_EOF

= 4 [line 35]

Unexpected end of file.
Unexpected end of file.

- **Version** 0.0.6
- **Since** 0.0.6

E_OTS_FileLoaderError::ERROR_INVALID_FILE_VERSION

= 1 [*line 27*]

Unsupported file version.
Unsupported file version.

- **Version** 0.0.6
- **Since** 0.0.6

E_OTS_FileLoaderError::ERROR_INVALID_FORMAT

= 8 [*line 47*]

File corrupted.
File corrupted.

- **Version** 0.0.6
- **Since** 0.0.6

E_OTS_FileLoaderError::ERROR_NOT_OPEN

= 6 [*line 43*]

Attempted to execute operation on not opened file.

Attempted to execute operation on not opened file.

- **Version** 0.0.6
- **Since** 0.0.6

E_OTS_FileLoaderError::ERROR_SEEK_ERROR

= 5 [*line 39*]

Failed to seek in given position in file.

Failed to seek in given position in file.

- **Version** 0.0.6
- **Since** 0.0.6

E_OTS_FileLoaderError::ERROR_TELL_ERROR

= 9 [*line 51*]

Failed to read position in file.

Failed to read position in file.

- **Version** 0.0.6
- **Since** 0.0.6

Class E_OTSGeneric

[line 20]

Generic exception class for general exceptions.

Generic exception class for general exceptions.

- **Package** POT
- **Version** 0.1.1
- **Since** 0.1.1

E_OTSGeneric::CONNECT_INVALID_DRIVER

= 2 *[line 29]*

Invalid database driver.

Invalid database driver.

- **Version** 0.1.1
- **Since** 0.1.1

E_OTSGeneric::CONNECT_NO_DRIVER

= 1 *[line 25]*

No database driver specified.

No database driver specified.

- **Version** 0.1.1
- **Since** 0.1.1

E_OTS_Generic::CREATE_ACCOUNT_IMPOSSIBLE

= 3 *[line 33]*

No free account numbers to create account.
No free account numbers to create account.

- **Version** 0.1.1
- **Since** 0.1.1

Class E_OTS_NoDriver

[line 20]

Occurs when code attempts to execute driven action that has no assigned driver to handle it.

Occurs when code attempts to execute driven action that has no assigned driver to handle it.

- **Package** POT
- **Version** 0.0.4
- **Since** 0.0.4

Class E_OTS_NotAContainer

[line 20]

Occurs when in database item which is not a container contains sub-items.

Occurs when in database item which is not a container contains sub-items.

- **Package** POT
- **Version** 0.1.0
- **Since** 0.1.0

Class E_OTS_NotLoaded

[line 20]

Occurs when code attempts to access property of not loaded object.

Occurs when code attempts to access property of not loaded object.

- **Package** POT
- **Version** 0.0.3
- **Since** 0.0.3

Class E_OTS_OTBMEError

[line 22]

OTBM map loading error.
OTBM map loading error.

- **Package** POT
- **Version** 0.0.6
- **Since** 0.0.6

E_OTS_OTBMErrror::LOADMAPERROR_OUTDATEDHEADER

= 3 *[line 27]*

Unsupported file version.
Unsupported file version.

- **Version** 0.0.6
- **Since** 0.0.6

E_OTS_OTBMErrror::LOADMAPERROR_UNKNOWNNODETYPE

= 8 *[line 31]*

Unknown node type.
Unknown node type.

- **Version** 0.0.6
- **Since** 0.0.6

Class E_OTS_OutOfBuffer

[line 22]

Occurs when properties stream has ended and there is still read attempt.

Occurs when properties stream has ended and there is still read attempt.

- **Package** POT
- **Version** 0.0.6
- **Since** 0.0.6

Class E_OTS_ReadOnly

[line 20]

Occurs when code attempts to evaluate write operation on read-only object.

Occurs when code attempts to evaluate write operation on read-only object.

- **Package** POT
- **Version** 0.1.0
- **Since** 0.1.0

Class IOTS_Cipher

[line 20]

Cypher encryption/decryption class interface.

Cypher encryption/decryption class interface.

- **Package** POT
- **Version** 0.1.2
- **Since** 0.1.2

string function IOTS_Cipher::decrypt(\$message) [line 36]

Function Parameters:

- *string* **\$message** Encrypted message.

Decrypts encrypted message.

Decrypts encrypted message.

- **Version** 0.1.2
- **Since** 0.1.2
- **Access** public

string function IOTS_Cipher::encrypt(\$message) [line 28]

Function Parameters:

- *string* **\$message** Message to be encrypted.

Ecnrypts message.

Ecnrypts message.

- **Version** 0.1.2
- **Since** 0.1.2
- **Access** public

Class IOTS_DAO

[line 23]

OTServ database object.

OTServ database object.

This interface indicates that class is a OTServ DAO class.

- **Package** POT
- **Version** 0.1.0
- **Deprecated** 0.1.0 This interface is not used anymore.
- **Since** 0.0.1

Class IOTS_DB

[line 25]

OTServ database handler interface.

OTServ database handler interface.

This interface specifies routines requires by DAO classes.

- **Package** POT
- **Version** 0.0.6
- **Version** 0.0.1
- **Deprecated** 0.0.5 Don't rely on this interface - it is for backward compatibility only. Check PDO instance instead.
- **Since** 0.0.1

Constructor *void* function IOTS_DB::__construct(\$params) [*line 33*]

Function Parameters:

- *array* **\$params** Connection configuration.

Connection parameters.

Connection parameters.

- **Version** 0.0.6
- **Version** 0.0.1
- **Since** 0.0.1
- **Access** public

string function IOTS_DB::fieldName(\$name) [*line 41*]

Function Parameters:

- *string* **\$name** Field name.

Query-quoted field name.

Query-quoted field name.

- **Version** 0.0.1
- **Since** 0.0.1
- **Access** public

int function IOTS_DB::lastInsertId() [*line 68*]

ID of last created record.

ID of last created record.

- **Version** 0.0.1
- **Since** 0.0.1
- **Access** public

string function IOTS_DB::limit([\$limit = false], [\$offset = false]) [*line 76*]

Function Parameters:

- *int|bool* **\$limit** Limit of rows to be affected by query (false if no limit).
- *int|bool* **\$offset** Number of rows to be skipped before applying query effects (false if no offset).

LIMIT/OFFSET clause for queries.

LIMIT/OFFSET clause for queries.

- **Version** 0.0.1
- **Since** 0.0.1
- **Access** public

mixed function IOTS_DB::SQLquery(\$query) [*line 62*]

Function Parameters:

- *string* **\$query** Database query.

Evaluates query.

Evaluates query.

- **Version** 0.0.1
- **Since** 0.0.1
- **Access** public

string function IOTS_DB::SQLquote(\$value) [*line 55*]

Function Parameters:

- *string* **\$value** Value to be quoted to be suitable for database query.

Query-quoted string value.

Query-quoted string value.

- **Version** 0.0.1
- **Since** 0.0.1
- **Access** public

string function IOTS_DB::tableName(\$name) [*line 48*]

Function Parameters:

- *string* **\$name** Table name.

Query-quoted table name.

Query-quoted table name.

- **Version** 0.0.1
- **Since** 0.0.1
- **Access** public

Class IOTS_Display

[line 24]

Display interface.

Display interface.

This way you can define low-level part of display logic to bind templates directly with POT objects.

- **Package** POT
- **Version** 0.1.0
- **Since** 0.1.0

string function IOTS_Display::displayAccount(\$account) [line 46]

Function Parameters:

- [OTS Account](#) **\$account** Account to be displayed.

Displays account.

Displays account.

- **Version** 0.1.0
- **Since** 0.1.0
- **Access** public

string function IOTS_Display::displayAccountsList(\$accountList, \$accountsList) [*line 53*]

Function Parameters:

- [OTS Accounts List](#) **\$accountsList** List to be displayed.
- [OTS Accounts List](#) **\$accountList**

Displays accounts list.

Displays accounts list.

- **Version** 0.1.0
- **Since** 0.1.0
- **Access** public

string function IOTS_Display::displayGroup(\$group) [*line 74*]

Function Parameters:

- [OTS Group](#) **\$group** Group to be displayed.

Displays group.

Displays group.

- **Version** 0.1.0
- **Since** 0.1.0
- **Access** public

string function IOTS_Display::displayGroupsList(\$groupsList) [*line 81*]
Function Parameters:

- [OTS_Groups_List](#) **\$groupsList** List to be displayed.

Displays groups list.

Displays groups list.

- **Version** 0.1.0
- **Since** 0.1.0
- **Access** public

string function IOTS_Display::displayGuild(\$guild) [*line 60*]
Function Parameters:

- [OTS_Guild](#) **\$guild** Guild to be displayed.

Displays guild.

Displays guild.

- **Version** 0.1.0
- **Since** 0.1.0
- **Access** public

string function IOTS_Display::displayGuildRank(\$guildRank) [*line 88*]

Function Parameters:

- [*OTS_GuildRank*](#) **\$guildRank** Rank to be displayed.

Displays rank.

Displays rank.

- **Version** 0.1.0
- **Since** 0.1.0
- **Access** public

string function IOTS_Display::displayGuildRanksList(\$guildRanksList) [*line 95*]

Function Parameters:

- [*OTS_GuildRanks_List*](#) **\$guildRanksList** List to be displayed.

Displays guild ranks list.

Displays guild ranks list.

- **Version** 0.1.0

- **Since** 0.1.0
- **Access** public

string function IOTS_Display::displayGuildsList(\$guildList, \$guildsList) [*line 67*]

Function Parameters:

- [*OTS_Guilds_List*](#) **\$guildsList** List to be displayed.
- [*OTS_Guild_List*](#) **\$guildList**

Displays guilds list.

Displays guilds list.

- **Version** 0.1.0
- **Since** 0.1.0
- **Access** public

string function IOTS_Display::displayPlayer(\$player) [*line 32*]

Function Parameters:

- [*OTS_Player*](#) **\$player** Player to be displayed.

Displays player.

Displays player.

- **Version** 0.1.0
- **Since** 0.1.0
- **Access** public

string function IOTS_Display::displayPlayersList(\$playersList) [*line 39*]

Function Parameters:

- [OTS Players List](#) **\$playersList** List to be displayed.

Displays players list.

Displays players list.

- **Version** 0.1.0
- **Since** 0.1.0
- **Access** public

Class IOTS_FileCache

[*line 20*]

This interface describes binary files cache control drivers.

This interface describes binary files cache control drivers.

- **Package** POT
- **Version** 0.0.6
- **Since** 0.0.6

OTS_FileNode|null function IOTS_FileCache::readCache(\$md5) [*line 28*]

Function Parameters:

- *string* **\$md5** MD5 hash of file.

Returns cache.

Returns cache.

- **Version** 0.0.6
- **Since** 0.0.6
- **Access** public

void function IOTS_FileCache::writeCache(\$md5, \$root) [*line 35*]

Function Parameters:

- *string* **\$md5** MD5 checksum of current file.
- [*OTS_FileNode*](#) **\$root** Root node of file which should be cached.

Writes node cache.

Writes node cache.

- **Version** 0.0.6
- **Since** 0.0.6
- **Access** public

Class IOTS_GuildAction

[line 32]

Guild action interface.

Guild action interface.

This interface indicates that class can handle OTServ guild action.

You can use it for example to handle invites or membership requests.

If you want to serialise (for example save in session) your guild obejcts with assigned drivers you need to implement also `__sleep()` and `__wakeup()` methods in your drivers, as assigned drivers are also serialised.

- **Package** POT
- **Version** 0.0.4
- **Since** 0.0.4

Constructor *void* function IOTS_GuildAction::__construct(\$guild) [line 41]

Function Parameters:

- [OTS Guild](#) **\$guild** Guild that this driver is assigned to.

Objects are initialized with a guild that they are assigned to.

Objects are initialized with a guild that they are assigned to.

It is recommeded that your implementations calls assignment functions of \$guild to automaticly assign itself as action handler.

- **Version** 0.0.4
- **Since** 0.0.4
- **Access** public

void function IOTS_GuildAction::addRequest(\$player) [line 54]

Function Parameters:

- [*OTS_Player*](#) **\$player** Player which is object of request.

Adds new request.

Adds new request.

- **Version** 0.0.4
- **Since** 0.0.4
- **Access** public

void function IOTS_GuildAction::deleteRequest(\$player) [line 60]

Function Parameters:

- [*OTS_Player*](#) **\$player** Player which is object of request.

Deletes request.

Deletes request.

- **Version** 0.0.4
- **Since** 0.0.4
- **Access** public

array function IOTS_GuildAction::listRequests() [line 48]

List of saved pending actions.

List of saved pending actions.

- **Version** 0.0.4
- **Since** 0.0.4
- **Access** public

void function IOTS_GuildAction::submitRequest(\$player) [*line 66*]

Function Parameters:

- [*OTS_Player*](#) **\$player** Player which is object of request.

Finalizes request.

Finalizes request.

- **Version** 0.0.4
- **Since** 0.0.4
- **Access** public

Class IOTS_ItemsCache

[*line 20*]

This interface defines items.

This interface defines items.xml cache handler as an standard file cache extender.

- **Package** POT
- **Version** 0.0.8
- **Since** 0.0.8

array|null function IOTS_ItemsCache::readItems(\$md5) [*line 28*]

Function Parameters:

- *string* **\$md5** MD5 hash of file.

Returns cache.

Returns cache.

- **Version** 0.0.8
- **Since** 0.0.8
- **Access** public

void function IOTS_ItemsCache::writeItems(\$md5, \$items) [*line 35*]

Function Parameters:

- *string* **\$md5** MD5 checksum of current file.
- *array* **\$items** List of items to be saved.

Writes items cache.

Writes items cache.

- **Version** 0.0.8
- **Since** 0.0.8

- **Access** public

Class OTS_Account

[line 30]

OTServ account abstraction.

OTServ account abstraction.

- **Package** POT
- **Property** string \$password: Password.
- **Property** string \$eMail: Email address.
- **Property** bool \$blocked: Blocked flag state.
- **Property** bool \$banned: Ban state.
- **Property-read** int \$id: Account number.
- **Property-read** bool \$loaded: Loaded state.
- **Property-read** [OTS Players List](#) \$playersList: Characters of this account.
- **Property-read** int \$access: Access level.
- **Version** 0.1.1
- **Version** 0.0.1
- **Since** 0.0.1

void function OTS_Account::ban([\$time = 0]) *[line 472]*

Function Parameters:

- **int \$time** Time for time until expires (0 - forever).

Bans current account.

Bans current account.

- **Version** 0.0.5
- **Version** 0.0.1
- **Since** 0.0.1
- **Since** 0.0.5
- **Access** public

void function OTS_Account::block() [*line 318*]

Blocks account.

Blocks account.

- **Version** 0.0.1
- **Since** 0.0.1
- **Access** public

int function OTS_Account::count() [*line 614*]

Returns number of player within.

Returns number of player within.

- **Version** 0.0.5
- **Version** 0.0.1
- **Throws** E_OTS_NotLoaded If account is not loaded.
- **Since** 0.0.5

- **Since** 0.0.1
- **Access** public

int function `OTS_Account::create([$min = 1], [$max = 9999999])` [*line 55*]
account.php

```

1      <?php
2
3      /**
4       * @ignore
5       * @package examples
6       * @author Wrzasq <wrzasq@gmail.com>
7       * @copyright 2007 (C) by Wrzasq
8       * @license http://www.gnu.org/licenses/lgpl-3.0.txt GNU Lesser General Public License, Version 3
9       */
10
11     // to not repeat all that stuff
12     include('quickstart.php');
13
14     // creates new OTS_Account object
15     $account = new OTS_Account();
16
17     // generates new account number
18     $number = $account->create();
19
20     /*
21     to generate number from 111111 to 999999 use:
22     $number = $account->create(111111, 999999);
23     */
24
25     // sets account info
26     $account->setPassword('secret'); // $account->setPassword( md5('secret') );
27     $account->setEmail('foo@example.com');
28     $account->unblock(); // remember to unblock!
29     $account->setPACCDays(0);
30     $account->save();
31
32     // give user his number
33     echo 'Your account number is: ', $number;
34
35     ?>

```

Function Parameters:

- *int* ***\$min*** Minimum number.
- *int* ***\$max*** Maximum number.

Creates new account.

Creates new account.

Create new account in given range (1 - 9999999 by default).

Remember! This method sets blocked flag to true after account creation!

- **Version** 0.1.1
- **Version** 0.0.1
- **Throws** E_OTS_Generic When there are no free account numbers.
- **Since** 0.0.1
- **Access** public
- **Example**

int function OTS_Account::createEx(\$group, [\$min = 1], [\$max = 9999999]) [*line 123*]
Function Parameters:

- [OTS_Group](#) **\$group** Group to be assigned to account.
- *int* **\$min** Minimum number.
- *int* **\$max** Maximum number.

Creates new account.

Creates new account.

Create new account in given range (1 - 9999999 by default) in given group.

Remember! This method sets blocked flag to true after account creation!

IMPORTANT: Since 0.0.6 there isn't group_id field which this method was created for. You should use [create\(\)](#) method.

- **Version** 0.0.6
- **Version** 0.0.1
- **Deprecated** 0.0.6 There is no more group_id field in database, use create().
- **Since** 0.0.4
- **Since** 0.0.1
- **Access** public

void function OTS_Account::delete() [line 526]

Deletes account.

Deletes account.

- **Version** 0.0.5
- **Version** 0.0.1
- **Throws** E_OTS_NotLoaded If account is not loaded.
- **Since** 0.0.5
- **Since** 0.0.1
- **Access** public

void function OTS_Account::find(\$email) [line 147]

Function Parameters:

- *string* **\$email** Account's e-mail address.

Loads account by it's e-mail address.

Loads account by it's e-mail address.

- **Version** 0.0.5
- **Version** 0.0.1
- **Since** 0.0.1
- **Since** 0.0.2
- **Access** public

int function OTS_Account::getAccess() [*line 545*]

Checks highest access level of account.

Checks highest access level of account.

- **Version** 0.0.1
- **Since** 0.0.1
- **Access** public

string function OTS_Account::getCustomField(\$field) [*line 365*]

Function Parameters:

- *string* **\$field** Field name.

Reads custom field.

Reads custom field.

Reads field by it's name. Can read any field of given record that exists in database.

Note: You should use this method only for fields that are not provided in standard setters/getters (SVN fields). This method runs SQL query each time you call it so it highly overloads used resources.

- **Version** 0.0.5
- **Version** 0.0.1
- **Throws** E_OTS_NotLoaded If account is not loaded.
- **Since** 0.0.3
- **Since** 0.0.1
- **Access** public

string function OTS_Account::getEmail() [*line 270*]

E-mail address.

E-mail address.

- **Version** 0.0.3
- **Version** 0.0.1
- **Throws** E_OTS_NotLoaded If account is not loaded.
- **Since** 0.0.1
- **Access** public

OTS_Group function OTS_Account::getGroup() [*line 212*]

Returns group of this account.

Returns group of this account.

- **Version** 0.1.0
- **Version** 0.0.1
- **Deprecated** 0.0.6 There is no more group_id field in database.
- **Throws** E_OTS_NotLoaded If account is not loaded.
- **Since** 0.0.1
- **Since** 0.0.4
- **Access** public

int function OTS_Account::getGuildAccess(\$guild) [*line 571*]

Function Parameters:

- [*OTS_Guild*](#) **\$guild** Guild in which access should be checked.

Checks highest access level of account in given guild.

Checks highest access level of account in given guild.

- **Version** 0.0.1
- **Since** 0.0.1
- **Access** public

int function OTS_Account::getId() [*line 193*]

Account number.

Account number.

- **Version** 0.0.3
- **Version** 0.0.1
- **Throws** E_OTS_NotLoaded If account is not loaded.
- **Since** 0.0.1
- **Access** public

Iterator function OTS_Account::getIterator() [*line 601*]

Returns players iterator.

Returns players iterator.

There is no need to implement entire Iterator interface since we have [players list class](#) for it.

- **Version** 0.0.5
- **Version** 0.0.1
- **Throws** E_OTS_NotLoaded If account is not loaded.
- **Since** 0.0.5
- **Since** 0.0.1

- **Access** public

int function OTS_Account::getPACCDays() [*line 331*]

PACC days.

PACC days.

- **Version** 0.0.4
- **Version** 0.0.1
- **Deprecated** 0.0.3 There is no more premdays field in accounts table.
- **Since** 0.0.1
- **Throws** E_OTS_NotLoaded If account is not loaded.
- **Access** public

string function OTS_Account::getPassword() [*line 243*]

Account's password.

Account's password.

- **Version** 0.0.3
- **Version** 0.0.1
- **Throws** E_OTS_NotLoaded If account is not loaded.
- **Since** 0.0.1
- **Access** public

array function OTS_Account::getPlayers() [*line 415*]

List of characters on account.

List of characters on account.

- **Version** 0.1.0
- **Version** 0.0.1
- **Deprecated** 0.0.5 Use `getPlayersList()`.
- **Since** 0.0.1
- **Throws** `E_OTS_NotLoaded` If account is not loaded.
- **Access** public

OTS_Players_List function `OTS_Account::getPlayersList()` [*line 445*]

List of characters on account.

List of characters on account.

In difference to [getPlayers\(\) method](#) this method returns filtered [OTS_Players_List](#) object instead of array of [OTS_Player](#) objects. It is more effective since `OTS_Player_List` doesn't perform all rows loading at once.

- **Version** 0.1.0
- **Version** 0.0.1
- **Throws** `E_OTS_NotLoaded` If account is not loaded.
- **Since** 0.0.5
- **Since** 0.0.1
- **Access** public

bool function `OTS_Account::isBanned()` [*line 507*]

Checks if account is banned.

Checks if account is banned.

- **Version** 0.0.5
- **Version** 0.0.1
- **Since** 0.0.1
- **Since** 0.0.5
- **Access** public

bool function OTS_Account::isBlocked() [*line 297*]

Checks if account is blocked.

Checks if account is blocked.

- **Version** 0.0.3
- **Version** 0.0.1
- **Throws** E_OTS_NotLoaded If account is not loaded.
- **Since** 0.0.1
- **Access** public

bool function OTS_Account::isLoaded() [*line 164*]

Checks if object is loaded.

Checks if object is loaded.

- **Version** 0.0.1
- **Since** 0.0.1
- **Access** public

void function OTS_Account::load(\$id) [*line 134*]

Function Parameters:

- *int* **\$id** Account number.

Loads account with given number.

Loads account with given number.

- **Version** 0.0.6
- **Version** 0.0.1
- **Since** 0.0.1
- **Access** public

void function OTS_Account::save() [*line 175*]

Updates account in database.

Updates account in database.

- **Version** 0.0.6
- **Version** 0.0.1
- **Throws** E_OTS_NotLoaded False if account doesn't have ID assigned.
- **Since** 0.0.1
- **Access** public

void function OTS_Account::setCustomField(\$field, \$value) [*line 391*]

Function Parameters:

- *string* **\$field** Field name.
- *mixed* **\$value** Field value.

Writes custom field.

Writes custom field.

Write field by it's name. Can write any field of given record that exists in database.

Note: You should use this method only for fields that are not provided in standard setters/getters (SVN fields). This method runs SQL query each time you call it so it highly overloads used resources.

Note: Make sure that you pass \$value argument of correct type. This method determinates whether to quote field name. It is safe - it makes you sure that no unproper queries that could lead to SQL injection will be executed, but it can make your code working wrong way. For example: \$object->setCustomField('foo', '1'); will quote 1 as as string ('1') instead of passing it as a integer.

- **Version** 0.0.5
- **Version** 0.0.1
- **Throws** E_OTS_NotLoaded If account is not loaded.
- **Since** 0.0.3
- **Since** 0.0.1
- **Access** public

void function OTS_Account::setEMail(\$email) [line 285]

Function Parameters:

- *string* **\$email** E-mail address.

Sets account's email.

Sets account's email.

- **Version** 0.0.1
- **Since** 0.0.1
- **Access** public

void function OTS_Account::setGroup(\$group) [line 232]

Function Parameters:

- [OTS_Group](#) **\$group** Group to be a member.

Assigns account to group.

Assigns account to group.

- **Version** 0.0.6
- **Version** 0.0.1
- **Deprecated** 0.0.6 There is no more group_id field in database.
- **Since** 0.0.1
- **Access** public

void function OTS_Account::setPACCDays(\$premdays, \$pacc) [line 348]

Function Parameters:

- *int* **\$pacc** PACC days.
- **\$premdays**

Sets PACC days count.

Sets PACC days count.

- **Version** 0.0.4
- **Version** 0.0.1
- **Deprecated** 0.0.3 There is no more premdays field in accounts table.

- **Since** 0.0.1
- **Access** public

void function OTS_Account::setPassword(\$password) [line 258]

Function Parameters:

- *string* **\$password** Password.

Sets account's password.

Sets account's password.

- **Version** 0.0.1
- **Since** 0.0.1
- **Access** public

void function OTS_Account::unban() [line 489]

Deletes ban from current account.

Deletes ban from current account.

- **Version** 0.0.5
- **Version** 0.0.1
- **Since** 0.0.1
- **Since** 0.0.5
- **Access** public

void function OTS_Account::unblock() [line 310]

Unblocks account.

Unblocks account.

- **Version** 0.0.1
- **Since** 0.0.1
- **Access** public

mixed function OTS_Account::__get(\$name) [*line 628*]

Function Parameters:

- *string* **\$name** Property name.

Magic PHP5 method.

Magic PHP5 method.

- **Version** 0.1.0
- **Version** 0.0.1
- **Throws** OutOfBoundsException For non-supported properties.
- **Since** 0.1.0
- **Since** 0.0.1
- **Access** public

void function OTS_Account::__set(\$name, \$value) [*line 670*]

Function Parameters:

- *string* **\$name** Property name.
- *mixed* **\$value** Property value.

Magic PHP5 method.

Magic PHP5 method.

- **Version** 0.1.0
- **Version** 0.0.1
- **Throws** OutOfBoundsException For non-supported properties.
- **Since** 0.1.0
- **Since** 0.0.1
- **Access** public

string function OTS_Account::__toString() [*line 718*]

Returns string representation of object.

Returns string representation of object.

If any display driver is currently loaded then it uses it's method. Otherwise just returns account number.

- **Version** 0.1.0
- **Version** 0.0.1
- **Since** 0.0.1
- **Since** 0.1.0
- **Access** public

Class OTS_Accounts_List

[*line 21*]

List of accounts.

List of accounts.

- **Package** POT
- **Version** 0.1.0
- **Since** 0.0.1

void function OTS_Accounts_List::deleteAccount(\$account) [*line 30*]

Function Parameters:

- [OTS_Account](#) **\$account** Account to be deleted.

Deletes account.

Deletes account.

- **Version** 0.0.5
- **Deprecated** 0.0.5 Use OTS_Account->delete().
- **Since** 0.0.1
- **Access** public

void function OTS_Accounts_List::init() [*line 43*]

Sets list parameters.

Sets list parameters.

This method is called at object creation.

- **Version** 0.0.5

- **Since** 0.0.1
- **Since** 0.0.5
- **Access** public

string function OTS_Accounts_List::__toString() [*line 58*]

Returns string representation of object.

Returns string representation of object.

If any display driver is currently loaded then it uses it's method.

- **Version** 0.1.0
- **Since** 0.0.1
- **Since** 0.1.0
- **Access** public

Class OTS_Admin

[*line 20*]

OTAdmin protocol client.

OTAdmin protocol client.

- **Package** POT
- **Version** 0.1.2
- **Since** 0.1.2

OTS_Admin::COMMAND_BAN_MANAGER

= 10 [*line 131*]

Not supported in current OTAdmin implementation.
Not supported in current OTAdmin implementation.

- **Version** 0.1.2
- **Since** 0.1.2

OTS_Admin::COMMAND_BROADCAST

= 1 [*line 95*]

Broadcast message.
Broadcast message.

- **Version** 0.1.2
- **Since** 0.1.2

OTS_Admin::COMMAND_CLOSE_SERVER

= 2 [*line 99*]

Closes server.
Closes server.

- **Version** 0.1.2
- **Since** 0.1.2

OTS_Admin::COMMAND_GETHOUSE

= 12 [*line 139*]

Not supported in current OTAdmin implementation.

Not supported in current OTAdmin implementation.

- **Version** 0.1.2
- **Since** 0.1.2

OTS_Admin::COMMAND_GETONLINE

= 8 [*line 123*]

Not supported in current OTAdmin implementation.

Not supported in current OTAdmin implementation.

- **Version** 0.1.2
- **Since** 0.1.2

OTS_Admin::COMMAND_KICK

= 9 [*line 127*]

Not supported in current OTAdmin implementation.

Not supported in current OTAdmin implementation.

- **Version** 0.1.2
- **Since** 0.1.2

OTS_Admin::COMMAND_OPEN_SERVER

= 4 [*line 107*]

Not supported in current OTAdmin implementation.

Not supported in current OTAdmin implementation.

- **Version** 0.1.2
- **Since** 0.1.2

OTS_Admin::COMMAND_PAY_HOUSES

= 3 [*line 103*]

Pays all rented shouses.

Pays all rented shouses.

- **Version** 0.1.2
- **Since** 0.1.2

OTS_Admin::COMMAND_PLAYER_INFO

= 7 [*line 119*]

Not supported in current OTAdmin implementation.

Not supported in current OTAdmin implementation.

- **Version** 0.1.2
- **Since** 0.1.2

OTS_Admin::COMMAND_RELOAD_SCRIPTS

= 6 *[line 115]*

Not supported in current OTAdmin implementation.

Not supported in current OTAdmin implementation.

- **Version** 0.1.2
- **Since** 0.1.2

OTS_Admin::COMMAND_SERVER_INFO

= 11 *[line 135]*

Not supported in current OTAdmin implementation.

Not supported in current OTAdmin implementation.

- **Version** 0.1.2
- **Since** 0.1.2

OTS_Admin::COMMAND_SHUTDOWN_SERVER

= 5 *[line 111]*

Shutdowns the server.

Shutdowns the server.

- **Version** 0.1.2
- **Since** 0.1.2

OTS_Admin::ENCRYPTION_RSA1024XTEA

= 1 *[line 153]*

Server uses XTEA encryption, XTEA key is being sent in 1024bit RSA encrypted packet.

Server uses XTEA encryption, XTEA key is being sent in 1024bit RSA encrypted packet.

- **Version 0.1.2**
- **Since 0.1.2**

OTS_Admin::REQUEST_COMMAND

= 4 *[line 37]*

OTAdmin commnd.

OTAdmin commnd.

- **Version 0.1.2**
- **Since 0.1.2**

OTS_Admin::REQUEST_ENCRYPTION

= 2 *[line 29]*

Encryption packet.

Encryption packet.

- **Version 0.1.2**

- **Since 0.1.2**

OTS_Admin::REQUEST_KEY_EXCHANGE

= 3 *[line 33]*

RSA key exchange.
RSA key exchange.

- **Version 0.1.2**
- **Since 0.1.2**

OTS_Admin::REQUEST_LOGIN

= 1 *[line 25]*

User login.
User login.

- **Version 0.1.2**
- **Since 0.1.2**

OTS_Admin::REQUEST_PING

= 5 *[line 41]*

Ping.
Ping.

- **Version 0.1.2**

- **Since 0.1.2**

OTS_Admin::REQUIRE_ENCRYPTION

= 2 [*line 148*]

Server requires encryption.
Server requires encryption.

- **Version 0.1.2**
- **Since 0.1.2**

OTS_Admin::REQUIRE_LOGIN

= 1 [*line 144*]

Server requires login.
Server requires login.

- **Version 0.1.2**
- **Since 0.1.2**

OTS_Admin::RESPOND_COMMAND_FAILED

= 7 [*line 70*]

Command failed.
Command failed.

- **Version** 0.1.2
- **Since** 0.1.2

OTS_Admin::RESPOND_COMMAND_OK

= 6 [*line 66*]

Command success.
Command success.

- **Version** 0.1.2
- **Since** 0.1.2

OTS_Admin::RESPOND_ENCRYPTION_FAILED

= 9 [*line 78*]

Encryption initialization failed.
Encryption initialization failed.

- **Version** 0.1.2
- **Since** 0.1.2

OTS_Admin::RESPOND_ENCRYPTION_OK

= 8 [*line 74*]

Encryption initialization success.
Encryption initialization success.

- **Version** 0.1.2
- **Since** 0.1.2

OTS_Admin::RESPOND_ERROR

= 12 [*line 90*]

Error.

Error.

- **Version** 0.1.2
- **Since** 0.1.2

OTS_Admin::RESPOND_HELLO

= 1 [*line 46*]

Hello respond.

Hello respond.

- **Version** 0.1.2
- **Since** 0.1.2

OTS_Admin::RESPOND_KEY_EXCHANGE_FAILED

= 3 [*line 54*]

Keys exchange failed.

Keys exchange failed.

- **Version** 0.1.2
- **Since** 0.1.2

OTS_Admin::RESPOND_KEY_EXCHANGE_OK

= 2 [*line 50*]

Keys exchange success.
Keys exchange success.

- **Version** 0.1.2
- **Since** 0.1.2

OTS_Admin::RESPOND_LOGIN_FAILED

= 5 [*line 62*]

Login incorrect.
Login incorrect.

- **Version** 0.1.2
- **Since** 0.1.2

OTS_Admin::RESPOND_LOGIN_OK

= 4 [*line 58*]

Login success.
Login success.

- **Version** 0.1.2
- **Since** 0.1.2

OTS_Admin::RESPOND_MESSAGE

= 11 [*line 86*]

Message.

Message.

- **Version** 0.1.2
- **Since** 0.1.2

OTS_Admin::RESPOND_PING_OK

= 10 [*line 82*]

Ping success.

Ping success.

- **Version** 0.1.2
- **Since** 0.1.2

Constructor *void* function OTS_Admin::__construct(\$host, [\$port = 7171]) [*line 204*]

Function Parameters:

- *string* **\$host** Target server.
- *int* **\$port** Port (7171 by default).

Creates new connection to OTServ administration backend.

Creates new connection to OTServ administration backend.

- **Version** 0.1.2
- **Throws** E_OTServ_ErrorCode When receive failed respond or unexpected message.
- **Since** 0.1.2
- **Access** public

void function OTS_Admin::broadcast(\$message) [line 498]

Function Parameters:

- *string* **\$message** Broadcast to be sent.

Sends broadcast message to all players.

Sends broadcast message to all players.

Sends COMMAND_BROADCAST command with given parameter.

- **Version** 0.1.2
- **Since** 0.1.2
- **Access** public

void function OTS_Admin::close() [line 512]

Closes server.

Closes server.

Sends COMMAND_CLOSE_SERVER command. This command closes server for connections to enable maintenance but doesn't shut it down.

- **Version** 0.1.2
- **Since** 0.1.2
- **Access** public

void function OTS_Admin::login(\$password) [line 409]

Function Parameters:

- *string* **\$password** Admin password.

Logs into server.

Logs into server.

Call this method if after connection is established login required flag is set.

- **Version** 0.1.2
- **Throws** E_OTS_ErrorCode When receive failed respond or unexpected message.
- **Since** 0.1.2
- **Access** public

void function OTS_Admin::payHouses() [line 525]

Takes fees for all rented houses.

Takes fees for all rented houses.

Sends COMMAND_PAY_HOUSES command.

- **Version** 0.1.2
- **Since** 0.1.2
- **Access** public

int function OTS_Admin::ping() [*line 436*]

Ping command.

Ping command.

Note: This methods calculates ping time based on [OTS_Admin::send\(\)](#) sub-call. This means ping time will be time used for entire sending operation including packet encryption, packing, unpacking and decryption.

- **Version** 0.1.2
- **Throws** E_OTS_ErrorCode When receive failed respond or unexpected message.
- **Since** 0.1.2
- **Access** public

bool function OTS_Admin::requiresEncryption() [*line 312*]

Checks if protocol requires encryption.

Checks if protocol requires encryption.

- **Version** 0.1.2
- **Since** 0.1.2
- **Access** public

bool function OTS_Admin::requiresLogin() [*line 302*]

Checks if protocol requires login.

Checks if protocol requires login.

- **Version** 0.1.2

- **Since** 0.1.2
- **Access** public

OTS_Buffer function OTS_Admin::send(\$message) [*line 334*]

Function Parameters:

- [*OTS_Buffer*](#) **\$message** Packet to be sent.

Sends OTAdmin packet.

Sends OTAdmin packet.

- **Version** 0.1.2
- **Throws** E_OTS_ErrorCode When receive RESPOND_ERROR message.
- **Since** 0.1.2
- **Access** public

void function OTS_Admin::shutdown() [*line 538*]

Shutdowns server.

Shutdowns server.

Sends COMMAND_SHUTDOWN_SERVER command. This command closes server thread.

- **Version** 0.1.2
- **Since** 0.1.2
- **Access** public

bool function OTS_Admin::usesRSA1024XTEA() [*line 322*]

Checks if protocol requires XTEA encryption with RSA-encrypted key.

Checks if protocol requires XTEA encryption with RSA-encrypted key.

- **Version** 0.1.2
- **Since** 0.1.2
- **Access** public

void function OTS_Admin::__clone() [line 564]

Magic PHP5 method.

Magic PHP5 method.

Creates new socket connection to server.

- **Version** 0.1.2
- **Since** 0.1.2
- **Access** public

void function OTS_Admin::__destruct() [line 375]

Closes connection.

Closes connection.

- **Version** 0.1.2
- **Since** 0.1.2
- **Access** public

void function OTS_Admin::__set_state(\$properties) [line 554]

Function Parameters:

- **array \$properties** List of object properties.

Magic PHP5 method.

Magic PHP5 method.

Allows object importing from [var_export\(\)](#).

- **Version** 0.1.2
- **Static**
- **Since** 0.1.2
- **Access** public

array function OTS_Admin::__sleep() [*line 577*]

Magic PHP5 method.

Magic PHP5 method.

Allows object serialisation.

- **Version** 0.1.2
- **Since** 0.1.2
- **Access** public

void function OTS_Admin::__wakeup() [*line 589*]

Magic PHP5 method.

Magic PHP5 method.

Allows object unserialisation.

- **Version** 0.1.2

- **Since** 0.1.2
- **Access** public

Class OTS_Base_DAO

[line 22]

Basic data access object routines.

Basic data access object routines.

- **Package** POT
- **Version** 0.1.0
- **Version** 0.0.5
- **Abstract Element**
- **Since** 0.0.5

OTS_Base_DAO::\$db

PDO = [line 29]

Database connection.

Database connection.

- **Version** 0.0.5
- **Since** 0.0.5
- **Access** protected

Constructor *void* function `OTS_Base_DAO::__construct()` [*line 36*]

Sets database connection handler.

Sets database connection handler.

- **Version** 0.1.0
- **Version** 0.0.5
- **Since** 0.0.5
- **Access** public

void function `OTS_Base_DAO::__clone()` [*line 73*]

Creates clone of object.

Creates clone of object.

Copy of object needs to have different ID.

- **Version** 0.0.5
- **Since** 0.0.5
- **Access** public

void function `OTS_Base_DAO::__set_state($properties)` [*line 87*]

Function Parameters:

- *array* **\$properties** List of object properties.

Magic PHP5 method.

Magic PHP5 method.

Allows object importing from [var export\(\)](#).

- **Version** 0.1.0
- **Version** 0.0.5
- **Static**
- **Since** 0.0.5
- **Access** public

array function OTS_Base_DAO::__sleep() [*line 49*]

Magic PHP5 method.

Magic PHP5 method.

Allows object serialisation.

- **Version** 0.0.5
- **Since** 0.0.5
- **Access** public

void function OTS_Base_DAO::__wakeup() [*line 61*]

Magic PHP5 method.

Magic PHP5 method.

Allows object unserialisation.

- **Version** 0.0.5
- **Since** 0.0.5
- **Access** public

Class OTS_Base_List

[line 25]

Basic list class routines.

Basic list class routines.

- **Package** POT
- **Property-write** int \$limit: Sets LIMIT clause.
- **Property-write** int \$offset: Sets OFFSET clause.
- **Property-write** [OTS_SQLFilter](#) \$filter: Sets filter for list SQL query.
- **Version** 0.1.0
- **Version** 0.0.5
- **Abstract Element**
- **Since** 0.0.5

OTS_Base_List::\$class

string = [line 81]

Class of generated objects.

Class of generated objects.

- **Version** 0.0.5
- **Since** 0.0.5
- **Access** protected

OTS_Base_List::\$table

string = [line 74]

Default table name for queries.

Default table name for queries.

- **Version** 0.0.5
- **Since** 0.0.5
- **Access** protected

Constructor *void* function OTS_Base_List::__construct() [line 88]

Sets database connection handler.

Sets database connection handler.

- **Version** 0.1.0
- **Version** 0.0.5
- **Since** 0.0.5
- **Access** public

int function OTS_Base_List::count() [line 245]

Returns number of rows on list in current criterium.

Returns number of rows on list in current criterium.

- **Version** 0.0.5
- **Version** 0.0.5
- **Since** 0.0.5
- **Access** public

LOTS_DAO function *OTS_Base_List::current()* [*line 193*]

Returns current row.

Returns current row.

- **Version** 0.1.0
- **Version** 0.0.5
- **Since** 0.0.5
- **Access** public

void function *OTS_Base_List::init()* [*line 97*]

Sets list parameters.

Sets list parameters.

- **Version** 0.0.5
- **Abstract Element**
- **Since** 0.0.5
- **Access** public

mixed function *OTS_Base_List::key()* [*line 224*]

Current cursor position.

Current cursor position.

- **Version** 0.0.5
- **Since** 0.0.5

- **Access** public

void function OTS_Base_List::next() [line 214]

Moves to next row.

Moves to next row.

- **Version** 0.0.5
- **Since** 0.0.5
- **Access** public

void function OTS_Base_List::orderBy(\$field, [\$order = POT::ORDER_ASC]) [line 278]

Function Parameters:

- [*OTS_SQLField*](#)|string **\$field** Field name.
- *int* **\$order** Sorting order (ascending by default).

Appends sorting rule.

Appends sorting rule.

- **Version** 0.0.7
- **Version** 0.0.5
- **Since** 0.0.5
- **Access** public

void function OTS_Base_List::resetOrder() [line 266]

Clears ORDER BY clause.

Clears ORDER BY clause.

- **Version** 0.0.5
- **Since** 0.0.5
- **Access** public

void function OTS_Base_List::rewind() [line 206]

Select rows from database.

Select rows from database.

- **Version** 0.0.5
- **Since** 0.0.5
- **Access** public

void function OTS_Base_List::setFilter([\$filter = null]) [line 258]

Function Parameters:

- [OTS_SQLFilter](#)|*null* **\$filter** Filter for list.

Sets filter on list.

Sets filter on list.

Call without argument to reset filter.

- **Version** 0.0.5
- **Since** 0.0.5
- **Access** public

void function OTS_Base_List::setLimit([\$limit = false]) [line 158]

Function Parameters:

- *int|bool* **\$limit** Limit for SELECT (false to reset).

Sets LIMIT.

Sets LIMIT.

- **Version** 0.0.5
- **Since** 0.0.5
- **Access** public

void function OTS_Base_List::setOffset([\$offset = false]) [line 175]

Function Parameters:

- *int|bool* **\$offset** Offset for SELECT (false to reset).

Sets OFFSET.

Sets OFFSET.

- **Version** 0.0.5
- **Since** 0.0.5
- **Access** public

bool function OTS_Base_List::valid() [line 234]

Checks if there are any rows left.

Checks if there are any rows left.

- **Version** 0.0.5
- **Since** 0.0.5
- **Access** public

void function OTS_Base_List::__set(\$name, \$value) [line 388]

Function Parameters:

- *string* **\$name** Property name.
- *mixed* **\$value** Property value.

Magic PHP5 method.

Magic PHP5 method.

- **Version** 0.1.0
- **Version** 0.0.5
- **Throws** OutOfBoundsException For non-supported properties.
- **Since** 0.1.0
- **Since** 0.0.5
- **Access** public

void function OTS_Base_List::__set_state(\$properties) [line 133]

Function Parameters:

- *array* **\$properties** List of object properties.

Magic PHP5 method.

Magic PHP5 method.

Allows object importing from [var_export\(\)](#).

- **Version** 0.0.6
- **Version** 0.0.5
- **Static**
- **Since** 0.0.5
- **Access** public

array function OTS_Base_List::__sleep() [*line 107*]

Magic PHP5 method.

Magic PHP5 method.

Allows object serialisation.

- **Version** 0.0.5
- **Since** 0.0.5
- **Access** public

void function OTS_Base_List::__wakeup() [*line 119*]

Magic PHP5 method.

Magic PHP5 method.

Allows object unserialisation.

- **Version** 0.0.5
- **Since** 0.0.5

- **Access** public

Class OTS_BinaryTools

[line 20]

This is class similar to OTS_Toolbox except that this class contains routines for binary number operations.

This is class similar to OTS_Toolbox except that this class contains routines for binary number operations.

- **Package** POT
- **Version** 0.1.2
- **Since** 0.1.2

string function OTS_BinaryTools::bin2Int(\$string) *[line 113]*

Function Parameters:

- *string* **\$string** Binary string.

Transforms binary representation of large integer into string.

Transforms binary representation of large integer into string.

- **Version** 0.1.2
- **Static**
- **Since** 0.1.2

- **Access** public

string function OTS_BinaryTools::int2Bin(\$number) [*line 133*]

Function Parameters:

- *string* **\$number** Large integer.

Transforms large integer into binary string.

Transforms large integer into binary string.

- **Version** 0.1.2
- **Static**
- **Since** 0.1.2
- **Access** public

int function OTS_BinaryTools::unsignedAdd(\$a, \$b) [*line 73*]

Function Parameters:

- *int* **\$a** First number.
- *int* **\$b** Second number.

Handle proper unsigned add, dealing with PHP's signed add.

Handle proper unsigned add, dealing with PHP's signed add.

- **Version** 0.1.2
- **Static**
- **Since** 0.1.2

- **Access** public

int function OTS_BinaryTools::unsignedRightShift(\$integer, \$n) [*line 29*]

Function Parameters:

- *int* **\$integer** Number to be shifted.
- *int* **\$n** Number of bits to shift.

Handle proper unsigned right shift, dealing with PHP's signed shift.

Handle proper unsigned right shift, dealing with PHP's signed shift.

- **Version** 0.1.2
- **Static**
- **Since** 0.1.2
- **Access** public

Class OTS_Buffer

[*line 30*]

Binary buffer container.

Binary buffer container.

- **Package** POT
- **Property** string \$buffer: Properties binary string.

- **Property** int \$char: getChar()/putChar() method wrapper.
- **Property** int \$short: getShort()/putShort() method wrapper.
- **Property** int \$long: getLong()/putLong() method wrapper.
- **Property** string \$string: getString(false)/putString(, true) call wrapper.
- **Property** int \$pos: getPos()/setPos() method wrapper.
- **Property-read** bool \$valid: isValid() method wrapper.
- **Property-read** int \$size: getSize() method wrapper.
- **Version** 0.1.2
- **Since** 0.1.2

OTS_Buffer::\$buffer

string = [line 37]

Node properties stream.

Node properties stream.

- **Version** 0.1.2
- **Since** 0.1.2
- **Access** protected

OTS_Buffer::\$pos

int = [line 43]

Properties stream pointer.

Properties stream pointer.

- **Version** 0.1.2
- **Since** 0.1.2

- **Access** protected

Constructor *void* function OTS_Buffer::__construct([\$buffer = ""]) [*line 50*]

Function Parameters:

- *string* **\$buffer** Buffer content.

Initializes new buffered reader.

Initializes new buffered reader.

- **Version** 0.1.2
- **Since** 0.1.2
- **Access** public

void function OTS_Buffer::check([\$size = 1]) [*line 114*]

Function Parameters:

- *int* **\$size** Amount of bytes that are going to be read.

Checks stream end state.

Checks stream end state.

- **Version** 0.1.2
- **Throws** E_OTS_OutOfBuffer When there is read attempt after end of stream.
- **Since** 0.1.2
- **Access** protected

string function OTS_Buffer::getBuffer() [*line 82*]

Returns properties stream.

Returns properties stream.

- **Version** 0.1.2
- **Since** 0.1.2
- **Access** public

int function OTS_Buffer::getChar() [*line 127*]

Returns single byte.

Returns single byte.

- **Version** 0.1.2
- **Since** 0.1.2
- **Access** public

int function OTS_Buffer::getLong() [*line 177*]

Returns quater byte.

Returns quater byte.

- **Version** 0.1.2
- **Since** 0.1.2
- **Access** public

int function OTS_Buffer::getPos() [*line 252*]

Returns current read position.

Returns current read position.

- **Version** 0.1.2
- **Since** 0.1.2
- **Access** public

int function OTS_Buffer::getShort() [*line 152*]

Returns double byte.

Returns double byte.

- **Version** 0.1.2
- **Since** 0.1.2
- **Access** public

int function OTS_Buffer::getSize() [*line 272*]

Returns buffer size.

Returns buffer size.

- **Version** 0.1.2
- **Since** 0.1.2
- **Access** public

string function OTS_Buffer::getString([\$length = false]) [*line 205*]

Function Parameters:

- *int|bool* **\$length** String length.

Returns string from buffer.

Returns string from buffer.

If length is not given then treats first byte from current buffer as string length.

- **Version** 0.1.2
- **Since** 0.1.2
- **Access** public

bool function OTS_Buffer::isValid() [*line 103*]

Checks if there is anything left in stream.

Checks if there is anything left in stream.

- **Version** 0.1.2
- **Since** 0.1.2
- **Access** public

void function OTS_Buffer::putChar(\$char) [*line 142*]

Function Parameters:

- *int* **\$char** Byte (char) value.

Appends single byte to buffer.

Appends single byte to buffer.

- **Version** 0.1.2
- **Since** 0.1.2
- **Access** public

void function OTS_Buffer::putLong(\$long) [line 192]

Function Parameters:

- *int* **\$long** Double word (long) value.

Appends quater byte to buffer.

Appends quater byte to buffer.

- **Version** 0.1.2
- **Since** 0.1.2
- **Access** public

void function OTS_Buffer::putShort(\$short) [line 167]

Function Parameters:

- *int* **\$short** Word (short) value.

Appends double byte to buffer.

Appends double byte to buffer.

- **Version** 0.1.2

- **Since** 0.1.2
- **Access** public

void function OTS_Buffer::putString(\$string, [\$dynamic = true]) [line 228]

Function Parameters:

- *string* **\$string** Binary length.
- *bool* **\$dynamic** Whether if string length is fixed or not (if it is dynamic then length will be inserted as short before string chunk).

Appends string to buffer.

Appends string to buffer.

- **Version** 0.1.2
- **Since** 0.1.2
- **Access** public

void function OTS_Buffer::reset() [line 242]

Empties buffer.

Empties buffer.

- **Version** 0.1.2
- **Since** 0.1.2
- **Access** public

void function OTS_Buffer::setBuffer(\$buffer) [line 92]

Function Parameters:

- *string* **\$buffer** Properties stream.

Sets properties stream.

Sets properties stream.

- **Version** 0.1.2
- **Since** 0.1.2
- **Access** public

void function OTS_Buffer::setPos(\$pos) [*line 262*]

Function Parameters:

- *int* **\$pos** Read position.

Seeks current reading position.

Seeks current reading position.

- **Version** 0.1.2
- **Since** 0.1.2
- **Access** public

void function OTS_Buffer::skip(\$n) [*line 282*]

Function Parameters:

- *int* **\$n** Bytes to skip.

Skips given amount of bytes.
Skips given amount of bytes.

- **Version** 0.1.2
- **Since** 0.1.2
- **Access** public

void function OTS_Buffer::__clone() [line 390]

Resets pointer of cloned object.
Resets pointer of cloned object.

- **Version** 0.1.2
- **Since** 0.1.2
- **Access** public

mixed function OTS_Buffer::__get(\$name) [line 295]

Function Parameters:

- *string* **\$name** Property name.

Magic PHP5 method.
Magic PHP5 method.

- **Version** 0.1.2
- **Throws** OutOfBoundsException For non-supported properties.

- **Since** 0.1.2
- **Access** public

void function OTS_Buffer::__set(\$name, \$value) [*line 343*]

Function Parameters:

- *string* **\$name** Property name.
- *mixed* **\$value** Property value.

Magic PHP5 method.

Magic PHP5 method.

- **Version** 0.1.2
- **Throws** OutOfBoundsException For non-supported properties.
- **Since** 0.1.2
- **Access** public

void function OTS_Buffer::__set_state(\$properties) [*line 64*]

Function Parameters:

- *array* **\$properties** List of object properties.

Magic PHP5 method.

Magic PHP5 method.

Allows object importing from [var_export\(\)](#).

- **Version** 0.1.2

- **Static**
- **Since** 0.1.2
- **Access** public

string function OTS_Buffer::__toString() [*line 382*]

Returns string representation of buffer object.
Returns string representation of buffer object.

- **Version** 0.1.2
- **Since** 0.1.2
- **Access** public

Class OTS_Container

[*line 22*]

Container item representation.
Container item representation.

- **Package** POT
- **Version** 0.1.0
- **Version** 0.0.3
- **Since** 0.0.3

void function OTS_Container::addItem(\$item) [*line 36*]

Function Parameters:

- [OTS_Item](#) \$item Item.

Adds item to container.

Adds item to container.

- **Version** 0.0.3
- **Since** 0.0.3
- **Access** public

int function OTS_Container::count() [*line 67*]

Number of items inside container.

Number of items inside container.

OTS_Container implementation of Countable interface differs from OTS_Item implementation. [OTS_Item::count\(\)](#) returns count of given item, OTS_Container::count() returns number of items inside container. If somehow it would be possible to make container items with more than 1 in one place, you can use [OTS_Item::getCount\(\)](#) and [OTS_Item::setCount\(\)](#) in code where you are not sure if working with regular item, or container.

- **Version** 0.0.3
- **Since** 0.0.3
- **Access** public

OTS_Item function OTS_Container::current() [*line 78*]

Returns current item.

Returns current item.

- **Version** 0.0.3
- **Deprecated** 0.1.0 Use getIterator().
- **Since** 0.0.3
- **Access** public

ArrayIterator function OTS_Container::getIterator() [*line 132*]

Returns iterator handle for loops.

Returns iterator handle for loops.

- **Version** 0.1.0
- **Version** 0.0.3
- **Since** 0.0.3
- **Since** 0.1.0
- **Access** public

mixed function OTS_Container::key() [*line 99*]

Current cursor position.

Current cursor position.

- **Version** 0.0.3
- **Deprecated** 0.1.0 Use getIterator().
- **Since** 0.0.3
- **Access** public

void function OTS_Container::next() [*line 88*]

Moves to next item.

Moves to next item.

- **Version** 0.0.3
- **Deprecated** 0.1.0 Use getIterator().
- **Since** 0.0.3
- **Access** public

void function OTS_Container::removeItem(\$item) [line 48]

Function Parameters:

- [*OTS_Item*](#) **\$item** Item.

Removes given item from current container.

Removes given item from current container.

Passed item must be exactly instance of item which is stored in container, not its copy.

- **Version** 0.0.3
- **Since** 0.0.3
- **Access** public

void function OTS_Container::rewind() [line 120]

Resets internal items array pointer.

Resets internal items array pointer.

- **Version** 0.0.3

- **Deprecated** 0.1.0 Use getIterator().
- **Since** 0.0.3
- **Access** public

bool function OTS_Container::valid() [*line 110*]

Checks if there are any items left.

Checks if there are any items left.

- **Version** 0.0.3
- **Deprecated** 0.1.0 Use getIterator().
- **Since** 0.0.3
- **Access** public

Class OTS_DB_MySQL

[*line 22*]

MySQL connection interface.

MySQL connection interface.

- **Package** POT
- **Version** 0.0.6
- **Version** 0.0.1
- **Since** 0.0.1

Constructor *void* function OTS_DB_MySQL::__construct(\$params) [line 50]

Function Parameters:

- *array* **\$params** Connection parameters.

Creates database connection.

Creates database connection.

Connects to MySQL database on given arguments.

List of parameters for this drivers:

- *host* - database server.
- *port* - port (optional, also it is possible to use host:port in *host* parameter).
- *database* - database name.
- *user* - user login.
- *password* - user password.

- **Version** 0.0.6
- **Version** 0.0.1
- **See** [POT::connect\(\)](#)
- **Since** 0.0.1
- **Access** public

string function OTS_DB_MySQL::fieldName(\$name) [line 105]

Function Parameters:

- *string* **\$name** Field name.

Query-quoted field name.

Query-quoted field name.

- **Version** 0.0.1
- **Since** 0.0.1
- **Access** public

string function OTS_DB_MySQL::limit([\$limit = false], [\$offset = false]) [*line 159*]

Function Parameters:

- *int|bool* **\$limit** Limit of rows to be affected by query (false if no limit).
- *int|bool* **\$offset** Number of rows to be skipped before applying query effects (false if no offset).

LIMIT/OFFSET clause for queries.

LIMIT/OFFSET clause for queries.

- **Version** 0.0.1
- **Since** 0.0.1
- **Access** public

PDOStatement|bool function OTS_DB_MySQL::SQLquery(\$query) [*line 147*]

Function Parameters:

- *string* **\$query** SQL query.

IOTS_DB method.

IOTS_DB method.

Overwrites PDO method.

- **Version** 0.0.1

- **Deprecated** 0.0.5 Use PDO::query().
- **Since** 0.0.1
- **Access** public

string function OTS_DB_MySQL::SQLquote(\$string) [*line 132*]

Function Parameters:

- *string* **\$string** String to be quoted.

IOTS_DB method.

IOTS_DB method.

Overwrites PDO method - we won't use quoting againsts other values.

- **Version** 0.0.7
- **Version** 0.0.1
- **Deprecated** 0.0.5 Use PDO::quote().
- **Since** 0.0.1
- **Access** public

string function OTS_DB_MySQL::tableName(\$name) [*line 116*]

Function Parameters:

- *string* **\$name** Table name.

Query-quoted table name.

Query-quoted table name.

- **Version** 0.0.1
- **Since** 0.0.1
- **Access** public

Class OTS_DB_ODBC

[line 22]

ODBC connection interface.

ODBC connection interface.

- **Package** POT
- **Version** 0.0.6
- **Version** 0.0.4
- **Since** 0.0.4

Constructor *void* function OTS_DB_ODBC::__construct(\$params) [line 50]

Function Parameters:

- *array* **\$params** Connection parameters.

Creates database connection.

Creates database connection.

Connects to ODBC data source on given arguments.

List of parameters for this drivers:

- *host* - database host.
- *port* - ODBC driver.
- *database* - database name.
- *user* - user login.
- *password* - user password.

- **Version** 0.0.6
- **Version** 0.0.4
- **See** [POT::connect\(\)](#)
- **Since** 0.0.4
- **Access** public

string function OTS_DB_ODBC::fieldName(\$name) [*line 98*]

Function Parameters:

- *string* **\$name** Field name.

Query-quoted field name.

Query-quoted field name.

- **Version** 0.0.4
- **Since** 0.0.4
- **Access** public

string function OTS_DB_ODBC::limit([\$limit = false], [\$offset = false]) [*line 152*]

Function Parameters:

- *int|bool* **\$limit** Limit of rows to be affected by query (false if no limit).
- *int|bool* **\$offset** Number of rows to be skipped before applying query effects (false if no offset).

LIMIT/OFFSET clause for queries.

LIMIT/OFFSET clause for queries.

- **Version** 0.0.4
- **Since** 0.0.4
- **Access** public

PDOStatement|bool function OTS_DB_ODBC::SQLquery(\$query) [*line 140*]

Function Parameters:

- *string* **\$query** SQL query.

IOTS_DB method.

IOTS_DB method.

Overwrites PDO method.

- **Version** 0.0.4
- **Deprecated** 0.0.5 Use PDO::query().
- **Since** 0.0.4
- **Access** public

string function OTS_DB_ODBC::SQLquote(\$string) [*line 125*]

Function Parameters:

- *string* **\$string** String to be quoted.

IOTS_DB method.

IOTS_DB method.

Overwrites PDO method - we won't use quoting against other values.

- **Version** 0.0.7
- **Version** 0.0.4
- **Deprecated** 0.0.5 Use PDO::quote().
- **Since** 0.0.4
- **Access** public

string function OTS_DB_ODBC::tableName(\$name) [*line 109*]

Function Parameters:

- *string* **\$name** Table name.

Query-quoted table name.

Query-quoted table name.

- **Version** 0.0.4
- **Since** 0.0.4
- **Access** public

Class OTS_DB_PostgreSQL

[*line 22*]

PostgreSQL connection interface.

PostgreSQL connection interface.

- **Package** POT
- **Version** 0.0.6
- **Version** 0.0.4
- **Since** 0.0.4

Constructor *void* function OTS_DB_PostgreSQL::__construct(\$params) [line 50]

Function Parameters:

- *array* **\$params** Connection parameters.

Creates database connection.

Creates database connection.

Connects to PgSQL database on given arguments.

List of parameters for this drivers:

- *host* - database server.
- *port* - port (optional, also it is possible to use host:port in *host* parameter).
- *database* - database name.
- *user* - user login.
- *password* - user password.

- **Version** 0.0.6
- **Version** 0.0.4
- **See** [POT::connect\(\)](#)
- **Since** 0.0.4
- **Access** public

string function OTS_DB_PostgreSQL::fieldName(\$name) [line 105]

Function Parameters:

- *string* **\$name** Field name.

Query-quoted field name.

Query-quoted field name.

- **Version** 0.0.4
- **Since** 0.0.4
- **Access** public

string function OTS_DB_PostgreSQL::limit([\$limit = false], [\$offset = false]) [*line 159*]

Function Parameters:

- *int|bool* **\$limit** Limit of rows to be affected by query (false if no limit).
- *int|bool* **\$offset** Number of rows to be skipped before applying query effects (false if no offset).

LIMIT/OFFSET clause for queries.

LIMIT/OFFSET clause for queries.

- **Version** 0.0.4
- **Since** 0.0.4
- **Access** public

PDOStatement|bool function OTS_DB_PostgreSQL::SQLquery(\$query) [*line 147*]

Function Parameters:

- *string* **\$query** SQL query.

IOTS_DB method.

IOTS_DB method.
Overwrites PDO method.

- **Version** 0.0.4
- **Deprecated** 0.0.5 Use PDO::query().
- **Since** 0.0.4
- **Access** public

string function OTS_DB_PostgreSQL::SQLquote(\$string) [*line 132*]

Function Parameters:

- *string* **\$string** String to be quoted.

IOTS_DB method.

IOTS_DB method.
Overwrites PDO method - we won't use quoting againsts other values.

- **Version** 0.0.7
- **Version** 0.0.4
- **Deprecated** 0.0.5 Use PDO::quote().
- **Since** 0.0.4
- **Access** public

string function OTS_DB_PostgreSQL::tableName(\$name) [*line 116*]

Function Parameters:

- *string* **\$name** Table name.

Query-quoted table name.

Query-quoted table name.

- **Version** 0.0.4
- **Since** 0.0.4
- **Access** public

Class OTS_DB_SQLite

[line 22]

SQLite connection interface.

SQLite connection interface.

- **Package** POT
- **Version** 0.0.6
- **Version** 0.0.1
- **Since** 0.0.1

Constructor *void* function OTS_DB_SQLite::__construct(\$params) [line 46]

Function Parameters:

- *array* **\$params** Connection parameters.

Creates database connection.

Creates database connection.

Connects to SQLite database on given arguments.

List of parameters for this drivers:

- *database* - database name.

- **Version** 0.0.7
- **Version** 0.0.1
- **See** [POT::connect\(\)](#)
- **Since** 0.0.1
- **Access** public

string function OTS_DB_SQLite::fieldName(\$name) [*line 67*]

Function Parameters:

- *string* **\$name** Field name.

Query-quoted field name.

Query-quoted field name.

- **Version** 0.0.1
- **Since** 0.0.1
- **Access** public

string function OTS_DB_SQLite::limit([\$limit = false], [\$offset = false]) [*line 121*]

Function Parameters:

- *int|bool* **\$limit** Limit of rows to be affected by query (false if no limit).
- *int|bool* **\$offset** Number of rows to be skipped before applying query effects (false if no offset).

LIMIT/OFFSET clause for queries.

LIMIT/OFFSET clause for queries.

- **Version** 0.0.1
- **Since** 0.0.1
- **Access** public

PDOStatement|bool function OTS_DB_SQLite::SQLquery(\$query) [*line 109*]

Function Parameters:

- *string* **\$query** SQL query.

IOTS_DB method.

IOTS_DB method.

Overwrites PDO method.

- **Version** 0.0.1
- **Deprecated** 0.0.5 Use PDO::query().
- **Since** 0.0.1
- **Access** public

string function OTS_DB_SQLite::SQLquote(\$string) [*line 94*]

Function Parameters:

- *string* **\$string** String to be quoted.

IOTS_DB method.

IOTS_DB method.

Overwrites PDO method - we won't use quoting against other values.

- **Version** 0.0.7
- **Version** 0.0.1
- **Deprecated** 0.0.5 Use PDO::quote().
- **Since** 0.0.1
- **Access** public

string function OTS_DB_SQLite::tableName(\$name) [*line 78*]

Function Parameters:

- *string* **\$name** Table name.

Query-quoted table name.

Query-quoted table name.

- **Version** 0.0.1
- **Since** 0.0.1
- **Access** public

Class OTS_FileLoader

[line 25]

Universal OTServ binary formats reader.

Universal OTServ binary formats reader.

- **Package** POT
- **Property-write** [OTS_FileCache](#) \$cacheDriver: Cache driver.
- **Version** 0.1.0
- **Version** 0.0.6
- **Since** 0.0.6

OTS_FileLoader::ESCAPE_CHAR

= 0xFD [line 38]

Escape another special byte.

Escape another special byte.

- **Version** 0.0.6
- **Since** 0.0.6

OTS_FileLoader::NODE_END

= 0xFF [line 34]

End of node.

End of node.

- **Version** 0.0.6
- **Since** 0.0.6

OTS_FileLoader::NODE_START

= 0xFE *[line 30]*

Start of node.

Start of node.

- **Version** 0.0.6
- **Since** 0.0.6

OTS_FileLoader::\$cache

OTS_FileCache = [line 59]

Cache handler.

Cache handler.

- **Version** 0.0.6
- **Since** 0.0.6
- **Access** protected

OTS_FileLoader::\$root

OTS_FileNode = [line 52]

Root node.

Root node.

- **Version** 0.0.6
- **Since** 0.0.6
- **Access** protected

void function OTS_FileLoader::loadFile(\$file) [line 131]

Function Parameters:

- *string* **\$file** Filepath.

Opens file.

Opens file.

- **Version** 0.1.0
- **Version** 0.0.6
- **Throws** E_OTS_FileLoaderError When error occurs during file operation.
- **Since** 0.0.6
- **Access** public

void function OTS_FileLoader::setCacheDriver([\$cache = null]) [line 119]

Function Parameters:

- [*IOTS_FileCache*](#) **\$cache** Cache handler (leave this parameter if you want to unset caching).

Sets cache handler.

Sets cache handler.

- **Version** 0.0.6
- **Since** 0.0.6
- **Access** public

void function OTS_FileLoader::__clone() [line 85]

Creates clone of object.

Creates clone of object.

Copy of object needs to have different ID.

- **Version** 0.0.6
- **Version** 0.0.6
- **Since** 0.0.6
- **Since** 0.0.6
- **Access** public

void function OTS_FileLoader::__set(\$name, \$value) [line 329]

Function Parameters:

- *string* **\$name** Property name.
- *mixed* **\$value** Property value.

Magic PHP5 method.

Magic PHP5 method.

- **Version** 0.1.0
- **Version** 0.0.6
- **Throws** OutOfBoundsException For non-supported properties.
- **Since** 0.1.0
- **Since** 0.0.6
- **Access** public

void function OTS_FileLoader::__set_state(\$properties) [*line 101*]

Function Parameters:

- *array* **\$properties** List of object properties.

Magic PHP5 method.

Magic PHP5 method.

Allows object importing from [var_export\(\)](#).

- **Version** 0.0.6
- **Version** 0.0.6
- **Static**
- **Since** 0.0.6
- **Since** 0.0.6
- **Access** public

array function OTS_FileLoader::__sleep() [*line 71*]

Magic PHP5 method.

Magic PHP5 method.

Allows object serialisation.

- **Version** 0.0.6
- **Version** 0.0.6
- **Since** 0.0.6
- **Since** 0.0.6
- **Access** public

Class OTS_FileNode

[line 27]

OTServ binary file node representation.

OTServ binary file node representation.

- **Package** POT
- **Property** [OTS_FileNode](#) \$next: Next sibling node.
- **Property** [OTS_FileNode](#) \$child: First child node.
- **Property** int \$type: Node type.
- **Version** 0.1.2
- **Version** 0.0.6
- **Since** 0.0.6

OTS_FileNode function OTS_FileNode::getChild() *[line 94]*

Returns first child.

Returns first child.

- **Version** 0.0.6
- **Since** 0.0.6
- **Access** public

OTS_FileNode function *OTS_FileNode::getNext()* [*line 74*]

Returns next sibling.

Returns next sibling.

- **Version** 0.0.6
- **Since** 0.0.6
- **Access** public

int function *OTS_FileNode::getType()* [*line 114*]

Returns node type.

Returns node type.

- **Version** 0.0.6
- **Since** 0.0.6
- **Access** public

void function *OTS_FileNode::setChild(\$child)* [*line 104*]

Function Parameters:

- [*OTS_FileNode*](#) **\$child** Child node.

Sets first child.

Sets first child.

- **Version** 0.0.6
- **Since** 0.0.6
- **Access** public

void function OTS_FileNode::setNext(\$next) [line 84]

Function Parameters:

- [*OTS_FileNode*](#) **\$next** Sibling node.

Sets next sibling.

Sets next sibling.

- **Version** 0.0.6
- **Since** 0.0.6
- **Access** public

void function OTS_FileNode::setType(\$type) [line 124]

Function Parameters:

- *int* **\$type** Node type.

Sets node type.

Sets node type.

- **Version** 0.0.6
- **Since** 0.0.6
- **Access** public

void function OTS_FileNode::__clone() [*line 55*]

Creates clone of object.

Creates clone of object.

Copy of object needs to have different ID.

- **Version** 0.0.6
- **Since** 0.0.6
- **Access** public

mixed function OTS_FileNode::__get(\$name) [*line 138*]

Function Parameters:

- *string* **\$name** Property name.

Magic PHP5 method.

Magic PHP5 method.

- **Version** 0.1.2
- **Version** 0.0.6
- **Throws** OutOfBoundsException For non-supported properties.
- **Since** 0.1.0
- **Since** 0.0.6

- **Access** public

void function OTS_FileNode::__set(\$name, \$value) [line 162]

Function Parameters:

- *string* **\$name** Property name.
- *mixed* **\$value** Property value.

Magic PHP5 method.

Magic PHP5 method.

- **Version** 0.1.2
- **Version** 0.0.6
- **Throws** OutOfBoundsException For non-supported properties.
- **Since** 0.1.0
- **Since** 0.0.6
- **Access** public

Class OTS_Group

[line 30]

OTServ user group abstraction.

OTServ user group abstraction.

- **Package** POT

- **Property** string \$name: Group name.
- **Property** int \$flags: Access flags.
- **Property** int \$access: Access level.
- **Property** int \$maxDepotItems: Maximum count of items in depot.
- **Property** int \$maxVIPList: Maximum count of entries in VIP list.
- **Property-read** bool \$loaded: Loaded state check.
- **Property-read** int \$id: Row ID.
- **Property-read** [OTS_Players_List](#) \$playersList: List of members of this group.
- **Version** 0.1.1
- **Version** 0.0.1
- **Since** 0.0.1

int function OTS_Group::count() [*line 412*]

Returns number of player within.

Returns number of player within.

- **Version** 0.0.5
- **Version** 0.0.1
- **Throws** E_OTS_NotLoaded If group is not loaded.
- **Since** 0.0.5
- **Since** 0.0.1
- **Access** public

void function OTS_Group::delete() [*line 375*]

Deletes group.

Deletes group.

- **Version** 0.0.5
- **Version** 0.0.1
- **Throws** E_OTS_NotLoaded If group is not loaded.
- **Since** 0.0.5
- **Since** 0.0.1
- **Access** public

void function OTS_Group::find(\$name) [*line 58*]

Function Parameters:

- *string* **\$name** Group name.

Loads group by it's name.

Loads group by it's name.

- **Version** 0.1.1
- **Version** 0.0.1
- **Since** 0.0.1
- **Since** 0.1.1
- **Access** public

int function OTS_Group::getAccess() [*line 181*]

Access level.

Access level.

- **Version** 0.0.3

- **Version** 0.0.1
- **Throws** E_OTS_NotLoaded If group is not loaded.
- **Since** 0.0.1
- **Access** public

string function OTS_Group::getCustomField(\$field) [*line 268*]

Function Parameters:

- *string* **\$field** Field name.

Reads custom field.

Reads custom field.

Reads field by it's name. Can read any field of given record that exists in database.

Note: You should use this method only for fields that are not provided in standard setters/getters (SVN fields). This method runs SQL query each time you call it so it highly overloads used resources.

- **Version** 0.0.5
- **Version** 0.0.1
- **Throws** E_OTS_NotLoaded If group is not loaded.
- **Since** 0.0.3
- **Since** 0.0.1
- **Access** public

int function OTS_Group::getFlags() [*line 154*]

Rights flags.

Rights flags.

- **Version** 0.0.3
- **Version** 0.0.1
- **Throws** E_OTS_NotLoaded If group is not loaded.
- **Since** 0.0.1
- **Access** public

int function OTS_Group::getId() [*line 110*]

Group ID.

Group ID.

- **Version** 0.0.3
- **Version** 0.0.1
- **Throws** E_OTS_NotLoaded If group is not loaded.
- **Since** 0.0.1
- **Access** public

Iterator function OTS_Group::getIterator() [*line 399*]

Returns players iterator.

Returns players iterator.

There is no need to implement entire Iterator interface since we have [players list class](#) for it.

- **Version** 0.0.5
- **Version** 0.0.1
- **Throws** E_OTS_NotLoaded If group is not loaded.
- **Since** 0.0.5
- **Since** 0.0.1

- **Access** public

int function OTS_Group::getMaxDepotItems() [*line 208*]

Maximum count of items in depot.

Maximum count of items in depot.

- **Version** 0.0.3
- **Version** 0.0.1
- **Throws** E_OTS_NotLoaded If group is not loaded.
- **Since** 0.0.1
- **Access** public

int function OTS_Group::getMaxVIPList() [*line 235*]

Maximum count of players in VIP list.

Maximum count of players in VIP list.

- **Version** 0.0.3
- **Version** 0.0.1
- **Throws** E_OTS_NotLoaded If group is not loaded.
- **Since** 0.0.1
- **Access** public

string function OTS_Group::getName() [*line 127*]

Group name.

Group name.

- **Version** 0.0.3
- **Version** 0.0.1
- **Throws** E_OTS_NotLoaded If group is not loaded.
- **Since** 0.0.1
- **Access** public

array function OTS_Group::getPlayers() [*line 318*]

List of characters in given group.

List of characters in given group.

- **Version** 0.1.0
- **Version** 0.0.1
- **Deprecated** 0.0.5 Use getPlayersList().
- **Since** 0.0.1
- **Throws** E_OTS_NotLoaded If group is not loaded.
- **Access** public

OTS_Players_List function OTS_Group::getPlayersList() [*line 348*]

List of characters in group.

List of characters in group.

In difference to [getPlayers\(\) method](#) this method returns filtered [OTS Players List](#) object instead of array of [OTS Player](#) objects. It is more effective since OTS_Player_List doesn't perform all rows loading at once.

- **Version** 0.1.0
- **Version** 0.0.1
- **Throws** E_OTS_NotLoaded If group is not loaded.

- **Since** 0.0.5
- **Since** 0.0.1
- **Access** public

bool function OTS_Group::isLoaded() [*line 75*]

Checks if object is loaded.

Checks if object is loaded.

- **Version** 0.0.1
- **Since** 0.0.1
- **Access** public

void function OTS_Group::load(\$id) [*line 45*]

Function Parameters:

- *int* **\$id** Group number.

Loads group with given id.

Loads group with given id.

- **Version** 0.0.5
- **Version** 0.0.1
- **Since** 0.0.1
- **Access** public

void function OTS_Group::save() [*line 85*]

Saves account in database.

Saves account in database.

- **Version** 0.0.5
- **Version** 0.0.1
- **Since** 0.0.1
- **Access** public

void function OTS_Group::setAccess(\$access) [*line 196*]

Function Parameters:

- *int* **\$access** Access level.

Sets access level.

Sets access level.

- **Version** 0.0.1
- **Since** 0.0.1
- **Access** public

void function OTS_Group::setCustomField(\$field, \$value) [*line 294*]

Function Parameters:

- *string* **\$field** Field name.
- *mixed* **\$value** Field value.

Writes custom field.

Writes custom field.

Write field by it's name. Can write any field of given record that exists in database.

Note: You should use this method only for fields that are not provided in standard setters/getters (SVN fields). This method runs SQL query each time you call it so it highly overloads used resources.

Note: Make sure that you pass \$value argument of correct type. This method determinates whether to quote field name. It is safe - it makes you sure that no unproper queries that could lead to SQL injection will be executed, but it can make your code working wrong way. For example: \$object->setCustomField('foo', '1'); will quote 1 as as string ('1') instead of passing it as a integer.

- **Version** 0.0.5
- **Version** 0.0.1
- **Throws** E_OTS_NotLoaded If group is not loaded.
- **Since** 0.0.3
- **Since** 0.0.1
- **Access** public

void function OTS_Group::setFlags(\$flags) [*line 169*]

Function Parameters:

- *int* **\$flags** Flags.

Sets rights flags.

Sets rights flags.

- **Version** 0.0.1
- **Since** 0.0.1
- **Access** public

void function OTS_Group::setMaxDepotItems(\$maxdepotitems) [line 223]

Function Parameters:

- *int* **\$maxdepotitems** Maximum value.

Sets maximum count of items in depot.

Sets maximum count of items in depot.

- **Version** 0.0.1
- **Since** 0.0.1
- **Access** public

void function OTS_Group::setMaxVIPList(\$maxviplist) [line 250]

Function Parameters:

- *int* **\$maxviplist** Maximum value.

Sets maximum count of players in VIP list.

Sets maximum count of players in VIP list.

- **Version** 0.0.1
- **Since** 0.0.1
- **Access** public

void function OTS_Group::setName(\$name) [line 142]

Function Parameters:

- *string* **\$name** Name.

Sets group's name.

Sets group's name.

- **Version** 0.0.1
- **Since** 0.0.1
- **Access** public

mixed function OTS_Group::__get(\$name) [*line 426*]

Function Parameters:

- *string* **\$name** Property name.

Magic PHP5 method.

Magic PHP5 method.

- **Version** 0.1.0
- **Version** 0.0.1
- **Throws** OutOfBoundsException For non-supported properties.
- **Since** 0.1.0
- **Since** 0.0.1
- **Access** public

void function OTS_Group::__set(\$name, \$value) [*line 468*]

Function Parameters:

- *string* **\$name** Property name.
- *mixed* **\$value** Property value.

Magic PHP5 method.

Magic PHP5 method.

- **Version** 0.1.0
- **Version** 0.0.1
- **Throws** OutOfBoundsException For non-supported properties.
- **Since** 0.1.0
- **Since** 0.0.1
- **Access** public

string function OTS_Group::__toString() [*line 506*]

Returns string representation of object.

Returns string representation of object.

If any display driver is currently loaded then it uses it's method. Else it returns group name.

- **Version** 0.1.0
- **Version** 0.0.1
- **Since** 0.0.1
- **Since** 0.1.0
- **Access** public

Class OTS_Groups_List

[line 21]

List of groups.

List of groups.

- **Package** POT
- **Version** 0.1.0
- **Since** 0.0.1

void function OTS_Groups_List::deleteGroup(\$group) [line 30]

Function Parameters:

- [OTS_Group](#) **\$group** Group to be deleted.

Deletes group.

Deletes group.

- **Version** 0.0.5
- **Deprecated** 0.0.5 Use OTS_Group->delete().
- **Since** 0.0.1
- **Access** public

void function OTS_Groups_List::init() [line 43]

Sets list parameters.

Sets list parameters.

This method is called at object creation.

- **Version** 0.0.5
- **Since** 0.0.1
- **Since** 0.0.5
- **Access** public

string function OTS_Groups_List::__toString() [*line 58*]

Returns string representation of object.

Returns string representation of object.

If any display driver is currently loaded then it uses it's method.

- **Version** 0.1.0
- **Since** 0.0.1
- **Since** 0.1.0
- **Access** public

Class OTS_Guild

[*line 31*]

OTServ guild abstraction.

OTServ guild abstraction.

- **Package** POT
- **Property** string \$read: Guild name.
- **Property** [OTS_Player](#) \$owner: Guild founder.

- **Property** int \$creationData: Guild creation data (mostly timestamp).
- **Property-read** int \$id: Guild ID.
- **Property-read** [OTS_GuildRanks_List](#) \$guildRanksList: Ranks in this guild.
- **Property-read** array \$invites: List of invited players.
- **Property-read** array \$requests: List of players that requested invites.
- **Property-write** [IOTS_GuildAction](#) \$invitesDriver: Invitations handler.
- **Property-write** [IOTS_GuildAction](#) \$requestsDriver: Membership requests handler.
- **Version** 0.1.1
- **Version** 0.0.4
- **Since** 0.0.4

void function OTS_Guild::acceptInvite(\$player) [line 452]

Function Parameters:

- [OTS_Player](#) **\$player** Player to be joined.

Finalise invitation.

Finalise invitation.

- **Version** 0.0.4
- **Throws** E_OTS_NotLoaded If guild is not loaded.
- **Throws** E_OTS_NoDriver If there is no invites driver assigned.
- **Since** 0.0.4
- **Access** public

void function OTS_Guild::acceptRequest(\$player) [line 544]

Function Parameters:

- [OTS_Player](#) \$player Player to be accepted.

Accepts player.

Accepts player.

- **Version** 0.0.4
- **Throws** E_OTS_NotLoaded If guild is not loaded.
- **Throws** E_OTS_NoDriver If there is no requests driver assigned.
- **Since** 0.0.4
- **Access** public

int function OTS_Guild::count() [*line 604*]

Returns number of ranks within.

Returns number of ranks within.

- **Version** 0.0.5
- **Version** 0.0.4
- **Throws** E_OTS_NotLoaded If guild is not loaded.
- **Since** 0.0.5
- **Since** 0.0.4
- **Access** public

void function OTS_Guild::delete() [*line 567*]

Deletes guild.

Deletes guild.

- **Version** 0.0.5
- **Version** 0.0.4
- **Throws** E_OTS_NotLoaded If guild is not loaded.
- **Since** 0.0.5
- **Since** 0.0.4
- **Access** public

void function OTS_Guild::deleteInvite(\$player) [line 429]

Function Parameters:

- [*OTS_Player*](#) **\$player** Player to be un-invited.

Deletes invitation for player to guild.

Deletes invitation for player to guild.

- **Version** 0.0.4
- **Throws** E_OTS_NotLoaded If guild is not loaded.
- **Throws** E_OTS_NoDriver If there is no invites driver assigned.
- **Since** 0.0.4
- **Access** public

void function OTS_Guild::deleteRequest(\$player) [line 521]

Function Parameters:

- [*OTS_Player*](#) **\$player** Player to be rejected.

Deletes request from player.

Deletes request from player.

- **Version** 0.0.4
- **Throws** E_OTS_NotLoaded If guild is not loaded.
- **Throws** E_OTS_NoDriver If there is no requests driver assigned.
- **Since** 0.0.4
- **Access** public

void function OTS_Guild::find(\$name) [*line 123*]

Function Parameters:

- *string* **\$name** Guild's name.

Loads guild by it's name.

Loads guild by it's name.

- **Version** 0.0.5
- **Version** 0.0.4
- **Since** 0.0.4
- **Access** public

int function OTS_Guild::getCreationData() [*line 245*]

Guild creation data.

Guild creation data.

- **Version** 0.0.4
- **Throws** E_OTTS_NotLoaded If guild is not loaded.
- **Since** 0.0.4
- **Access** public

string function OTS_Guild::getCustomField(\$field) [*line 277*]

Function Parameters:

- *string* **\$field** Field name.

Reads custom field.

Reads custom field.

Reads field by it's name. Can read any field of given record that exists in database.

Note: You should use this method only for fields that are not provided in standard setters/getters (SVN fields). This method runs SQL query each time you call it so it highly overloads used resources.

- **Version** 0.0.8
- **Version** 0.0.4
- **Throws** E_OTTS_NotLoaded If guild is not loaded.
- **Since** 0.0.4
- **Access** public

array function OTS_Guild::getGuildRanks() [*line 326*]

Reads all ranks that are in this guild.

Reads all ranks that are in this guild.

- **Version** 0.1.0

- **Version** 0.0.4
- **Deprecated** 0.0.5 Use getGuildRanksList().
- **Since** 0.0.4
- **Throws** E_OTS_NotLoaded If guild is not loaded.
- **Access** public

OTS_GuildRanks_List function OTS_Guild::getGuildRanksList() [*line 356*]

List of ranks in guild.

List of ranks in guild.

In difference to [getGuildRanks\(\) method](#) this method returns filtered [OTS_GuildRanks_List](#) object instead of array of [OTS_GuildRank](#) objects. It is more effective since OTS_GuildRanks_List doesn't perform all rows loading at once.

- **Version** 0.1.0
- **Version** 0.0.4
- **Throws** E_OTS_NotLoaded If guild is not loaded.
- **Since** 0.0.5
- **Since** 0.0.4
- **Access** public

int function OTS_Guild::getId() [*line 174*]

Guild ID.

Guild ID.

- **Version** 0.0.4
- **Throws** E_OTS_NotLoaded If guild is not loaded.
- **Since** 0.0.4
- **Access** public

Iterator function OTS_Guild::getIterator() [*line 591*]

Returns ranks iterator.

Returns ranks iterator.

There is no need to implement entire Iterator interface since we have [ranks list class](#) for it.

- **Version** 0.0.5
- **Version** 0.0.4
- **Throws** E_OTS_NotLoaded If guild is not loaded.
- **Since** 0.0.5
- **Since** 0.0.4
- **Access** public

string function OTS_Guild::getName() [*line 190*]

Guild name.

Guild name.

- **Version** 0.0.4
- **Throws** E_OTS_NotLoaded If guild is not loaded.
- **Since** 0.0.4
- **Access** public

OTS_Player function OTS_Guild::getOwner() [*line 217*]

Returns owning player of this player.

Returns owning player of this player.

- **Version** 0.1.0
- **Version** 0.0.4
- **Throws** E_OTTS_NotLoaded If guild is not loaded.
- **Since** 0.0.4
- **Access** public

void function OTS_Guild::invite(\$player) [*line 406*]
Function Parameters:

- [OTS_Player](#) **\$player** Player to be invited.

Invites player to guild.
 Invites player to guild.

- **Version** 0.0.4
- **Throws** E_OTTS_NotLoaded If guild is not loaded.
- **Throws** E_OTTS_NoDriver If there is no invites driver assigned.
- **Since** 0.0.4
- **Access** public

bool function OTS_Guild::isLoading() [*line 140*]

Checks if object is loaded.
 Checks if object is loaded.

- **Version** 0.0.4

- **Since** 0.0.4
- **Access** public

array function OTS_Guild::listInvites() [*line 383*]

Returns list of invited players.

Returns list of invited players.

- **Version** 0.0.4
- **Throws** E_OTS_NotLoaded If guild is not loaded.
- **Throws** E_OTS_NoDriver If there is no invites driver assigned.
- **Since** 0.0.4
- **Access** public

array function OTS_Guild::listRequests() [*line 475*]

Returns list of players that requested membership.

Returns list of players that requested membership.

- **Version** 0.0.4
- **Throws** E_OTS_NotLoaded If guild is not loaded.
- **Throws** E_OTS_NoDriver If there is no requests driver assigned.
- **Since** 0.0.4
- **Access** public

void function OTS_Guild::load(\$id) [*line 111*]

Function Parameters:

- *int* **\$id** Guild's ID.

Loads guild with given id.

Loads guild with given id.

- **Version** 0.0.5
- **Version** 0.0.4
- **Since** 0.0.4
- **Access** public

void function OTS_Guild::request(\$player) [line 498]

Function Parameters:

- [*OTS_Player*](#) **\$player** Player that requested membership.

Requests membership in guild for player player.

Requests membership in guild for player player.

- **Version** 0.0.4
- **Throws** E_OTS_NotLoaded If guild is not loaded.
- **Throws** E_OTS_NoDriver If there is no requests driver assigned.
- **Since** 0.0.4
- **Access** public

void function OTS_Guild::save() [line 150]

Saves guild in database.

Saves guild in database.

- **Version** 0.0.5
- **Version** 0.0.4
- **Since** 0.0.4
- **Access** public

void function OTS_Guild::setCreationData(\$creationdata) [line 260]

Function Parameters:

- *int* **\$creationdata** Guild creation data.

Sets guild creation data.

Sets guild creation data.

- **Version** 0.0.4
- **Since** 0.0.4
- **Access** public

void function OTS_Guild::setCustomField(\$field, \$value) [line 302]

Function Parameters:

- *string* **\$field** Field name.
- *mixed* **\$value** Field value.

Writes custom field.

Writes custom field.

Write field by it's name. Can write any field of given record that exists in database.

Note: You should use this method only for fields that are not provided in standard

setters/getters (SVN fields). This method runs SQL query each time you call it so it highly overloads used resources.

Note: Make sure that you pass \$value argument of correct type. This method determinates whether to quote field value. It is safe - it makes you sure that no unproper queries that could lead to SQL injection will be executed, but it can make your code working wrong way. For example: \$object->setCustomField('foo', '1'); will quote 1 as as string ('1') instead of passing it as a integer.

- **Version** 0.0.5
- **Version** 0.0.4
- **Throws** E_OTS_NotLoaded If guild is not loaded.
- **Since** 0.0.4
- **Access** public

void function OTS_Guild::setInvitesDriver([\$invites = null]) [line 90]

Function Parameters:

- [*IOTS_GuildAction*](#) **\$invites** Invites driver (don't pass it to clear driver).

Assigns invites handler.

Assigns invites handler.

- **Version** 0.0.4
- **Since** 0.0.4
- **Access** public

void function OTS_Guild::setName(\$name) [line 205]

Function Parameters:

- *string* **\$name** Name.

Sets players's name.

Sets players's name.

- **Version** 0.0.4
- **Since** 0.0.4
- **Access** public

void function OTS_Guild::setOwner(\$owner) [*line 234*]

Function Parameters:

- [*OTS_Player*](#) **\$owner** Owning player.

Assigns guild to owner.

Assigns guild to owner.

- **Version** 0.0.4
- **Since** 0.0.4
- **Access** public

void function OTS_Guild::setRequestsDriver([\$requests = null]) [*line 100*]

Function Parameters:

- [*IOTS_GuildAction*](#) **\$requests** Membership requests driver (don't pass it to clear driver).

Assigns requests handler.

Assigns requests handler.

- **Version** 0.0.4
- **Since** 0.0.4
- **Access** public

void function OTS_Guild::__clone() [*line 74*]

Creates clone of object.

Creates clone of object.

Copy of object needs to have different ID.

- **Version** 0.0.4
- **Since** 0.0.4
- **Access** public

mixed function OTS_Guild::__get(\$name) [*line 618*]

Function Parameters:

- *string* **\$name** Property name.

Magic PHP5 method.

Magic PHP5 method.

- **Version** 0.1.0
- **Version** 0.0.4

- **Throws** OutOfBoundsException For non-supported properties.
- **Since** 0.1.0
- **Since** 0.0.4
- **Access** public

void function OTS_Guild::__set(\$name, \$value) [*line 657*]

Function Parameters:

- *string* **\$name** Property name.
- *mixed* **\$value** Property value.

Magic PHP5 method.

Magic PHP5 method.

- **Version** 0.1.0
- **Version** 0.0.4
- **Throws** OutOfBoundsException For non-supported properties.
- **Since** 0.1.0
- **Since** 0.0.4
- **Access** public

array function OTS_Guild::__sleep() [*line 62*]

Magic PHP5 method.

Magic PHP5 method.

Allows object serialisation.

- **Version** 0.0.4

- **Since** 0.0.4
- **Access** public

string function OTS_Guild::__toString() [*line 695*]

Returns string representation of object.

Returns string representation of object.

If any display driver is currently loaded then it uses it's method. Else it returns guild name.

- **Version** 0.1.0
- **Version** 0.0.4
- **Since** 0.0.4
- **Since** 0.1.0
- **Access** public

Class OTS_GuildRank

[*line 28*]

OTServ guild rank abstraction.

OTServ guild rank abstraction.

- **Package** POT
- **Property** string \$name: Rank title.
- **Property** [OTS_Guild](#) \$guild: Guild in which rank exists.
- **Property** int \$level: Guild access level.
- **Property-read** bool \$loaded: Loaded state check.

- **Property-read** int \$id: Row ID.
- **Property-read** [OTS_Players_List](#) \$playersList: List of members with this rank.
- **Version** 0.1.1
- **Version** 0.0.4
- **Since** 0.0.4

int function OTS_GuildRank::count() [*line 363*]

Returns number of player within.

Returns number of player within.

- **Version** 0.0.5
- **Version** 0.0.4
- **Throws** E_OTS_NotLoaded If rank is not loaded.
- **Since** 0.0.5
- **Since** 0.0.4
- **Access** public

void function OTS_GuildRank::delete() [*line 326*]

Deletes guild rank.

Deletes guild rank.

- **Version** 0.0.5
- **Version** 0.0.4
- **Throws** E_OTS_NotLoaded If guild rank is not loaded.
- **Since** 0.0.5
- **Since** 0.0.4
- **Access** public

void function OTS_GuildRank::find(\$name, [\$guild = null]) [*line 58*]

Function Parameters:

- *string* **\$name** Rank's name.
- [*OTS_Guild*](#) **\$guild** Guild in which rank should be found.

Loads rank by it's name.

Loads rank by it's name.

As there can be several ranks with same name in different guilds you can pass optional second parameter to specify in which guild script should look for rank.

- **Version** 0.0.5
- **Version** 0.0.4
- **Since** 0.0.4
- **Access** public

string function OTS_GuildRank::getCustomField(\$field) [*line 220*]

Function Parameters:

- *string* **\$field** Field name.

Reads custom field.

Reads custom field.

Reads field by it's name. Can read any field of given record that exists in database.

Note: You should use this method only for fields that are not provided in standard setters/getters (SVN fields). This method runs SQL query each time you call it so it highly overloads used resources.

- **Version** 0.0.5
- **Version** 0.0.4
- **Throws** E_OTs_NotLoaded If rank is not loaded.
- **Since** 0.0.4
- **Access** public

OTS_Guild function OTS_GuildRank::getGuild() [*line 160*]

Returns guild of this rank.

Returns guild of this rank.

- **Version** 0.1.0
- **Version** 0.0.4
- **Throws** E_OTs_NotLoaded If rank is not loaded.
- **Since** 0.0.4
- **Access** public

int function OTS_GuildRank::getId() [*line 117*]

Rank ID.

Rank ID.

- **Version** 0.0.4
- **Throws** E_OTs_NotLoaded If rank is not loaded.
- **Since** 0.0.4
- **Access** public

Iterator function OTS_GuildRank::getIterator() [*line 350*]

Returns players iterator.

Returns players iterator.

There is no need to implement entire Iterator interface since we have [players list class](#) for it.

- **Version** 0.0.5
- **Version** 0.0.4
- **Throws** E_OTs_NotLoaded If rank is not loaded.
- **Since** 0.0.5
- **Since** 0.0.4
- **Access** public

int function OTS_GuildRank::getLevel() [*line 188*]

Rank's access level.

Rank's access level.

- **Version** 0.0.4
- **Throws** E_OTs_NotLoaded If rank is not loaded.
- **Since** 0.0.4
- **Access** public

string function OTS_GuildRank::getName() [*line 133*]

Rank name.

Rank name.

- **Version** 0.0.4
- **Throws** E_OTs_NotLoaded If rank is not loaded.

- **Since** 0.0.4
- **Access** public

array function OTS_GuildRank::getPlayers() [*line 269*]

Reads all players who has this rank set.

Reads all players who has this rank set.

- **Version** 0.1.0
- **Version** 0.0.4
- **Deprecated** 0.0.5 Use getPlayersList().
- **Since** 0.0.4
- **Throws** E_OTS_NotLoaded If rank is not loaded.
- **Access** public

OTS_Players_List function OTS_GuildRank::getPlayersList() [*line 299*]

List of characters with current rank.

List of characters with current rank.

In difference to [getPlayers\(\) method](#) this method returns filtered [OTS_Players_List](#) object instead of array of [OTS_Player](#) objects. It is more effective since OTS_Player_List doesn't perform all rows loading at once.

- **Version** 0.1.0
- **Version** 0.0.4
- **Throws** E_OTS_NotLoaded If rank is not loaded.
- **Since** 0.0.5
- **Since** 0.0.4
- **Access** public

bool function OTS_GuildRank::isLoaded() [*line 83*]

Checks if object is loaded.

Checks if object is loaded.

- **Version** 0.0.4
- **Since** 0.0.4
- **Access** public

void function OTS_GuildRank::load(\$id) [*line 43*]

Function Parameters:

- *int* **\$id** Rank's ID.

Loads rank with given id.

Loads rank with given id.

- **Version** 0.0.5
- **Version** 0.0.4
- **Since** 0.0.4
- **Access** public

void function OTS_GuildRank::save() [*line 93*]

Saves rank in database.

Saves rank in database.

- **Version** 0.0.8
- **Version** 0.0.4
- **Since** 0.0.4
- **Access** public

void function OTS_GuildRank::setCustomField(\$field, \$value) [line 245]

Function Parameters:

- *string* **\$field** Field name.
- *mixed* **\$value** Field value.

Writes custom field.

Writes custom field.

Write field by it's name. Can write any field of given record that exists in database.

Note: You should use this method only for fields that are not provided in standard setters/getters (SVN fields). This method runs SQL query each time you call it so it highly overloads used resources.

Note: Make sure that you pass \$value argument of correct type. This method determinates whether to quote field value. It is safe - it makes you sure that no unproper queries that could lead to SQL injection will be executed, but it can make your code working wrong way. For example: \$object->setCustomField('foo', '1'); will quote 1 as as string ('1') instead of passing it as a integer.

- **Version** 0.0.5
- **Version** 0.0.4
- **Throws** E_OTS_NotLoaded If rank is not loaded.
- **Since** 0.0.4
- **Access** public

void function OTS_GuildRank::setGuild(\$guild) [line 177]

Function Parameters:

- [*OTS_Guild*](#) **\$guild** Owing guild.

Assigns rank to guild.

Assigns rank to guild.

- **Version** 0.0.4
- **Since** 0.0.4
- **Access** public

void function OTS_GuildRank::setLevel(\$level) [line 203]

Function Parameters:

- *int* **\$level** access level within guild.

Sets rank's access level within guild.

Sets rank's access level within guild.

- **Version** 0.0.4
- **Since** 0.0.4
- **Access** public

void function OTS_GuildRank::setName(\$name) [line 148]

Function Parameters:

- *string* **\$name** Name.

Sets rank's name.

Sets rank's name.

- **Version** 0.0.4
- **Since** 0.0.4
- **Access** public

mixed function OTS_GuildRank::__get(\$name) [*line 377*]

Function Parameters:

- *string* **\$name** Property name.

Magic PHP5 method.

Magic PHP5 method.

- **Version** 0.1.0
- **Version** 0.0.4
- **Throws** OutOfBoundsException For non-supported properties.
- **Since** 0.1.0
- **Since** 0.0.4
- **Access** public

void function OTS_GuildRank::__set(\$name, \$value) [*line 413*]

Function Parameters:

- *string* **\$name** Property name.
- *mixed* **\$value** Property value.

Magic PHP5 method.

Magic PHP5 method.

- **Version** 0.1.0
- **Version** 0.0.4
- **Throws** OutOfBoundsException For non-supported properties.
- **Since** 0.1.0
- **Since** 0.0.4
- **Access** public

string function OTS_GuildRank::__toString() [*line 443*]

Returns string representation of object.

Returns string representation of object.

If any display driver is currently loaded then it uses it's method. Else it returns rank name.

- **Version** 0.1.0
- **Version** 0.0.4
- **Since** 0.0.4
- **Since** 0.1.0
- **Access** public

Class OTS_GuildRanks_List

[line 21]

List of guild ranks.

List of guild ranks.

- **Package** POT
- **Version** 0.1.0
- **Since** 0.0.4

void function OTS_GuildRanks_List::deleteGuildRank(\$guildRank) [line 30]

Function Parameters:

- [OTS_GuildRank](#) **\$guildRank** Rank to be deleted.

Deletes guild rank.

Deletes guild rank.

- **Version** 0.0.5
- **Deprecated** 0.0.5 Use OTS_GuildRank->delete().
- **Since** 0.0.4
- **Access** public

void function OTS_GuildRanks_List::init() [line 43]

Sets list parameters.

Sets list parameters.

This method is called at object creation.

- **Version** 0.0.5
- **Since** 0.0.4
- **Since** 0.0.5
- **Access** public

string function OTS_GuildRanks_List::__toString() [*line 58*]

Returns string representation of object.

Returns string representation of object.

If any display driver is currently loaded then it uses it's method.

- **Version** 0.1.0
- **Since** 0.0.4
- **Since** 0.1.0
- **Access** public

Class OTS_Guilds_List

[*line 21*]

List of guilds.

List of guilds.

- **Package** POT
- **Version** 0.1.0
- **Since** 0.0.4

void function OTS_Guilds_List::deleteGuild(\$guild) [*line 30*]

Function Parameters:

- [*OTS_Guild*](#) **\$guild** Guild to be deleted.

Deletes guild.

Deletes guild.

- **Version** 0.0.5
- **Deprecated** 0.0.5 Use OTS_Guild->delete().
- **Since** 0.0.4
- **Access** public

void function OTS_Guilds_List::init() [*line 43*]

Sets list parameters.

Sets list parameters.

This method is called at object creation.

- **Version** 0.0.5
- **Since** 0.0.4
- **Since** 0.0.5
- **Access** public

string function OTS_Guilds_List::__toString() [*line 58*]

Returns string representation of object.

Returns string representation of object.

If any display driver is currently loaded then it uses it's method.

- **Version** 0.1.0
- **Since** 0.0.4
- **Since** 0.1.0
- **Access** public

Class OTS_House

[line 31]

Wrapper for house information.

Wrapper for house information.

- **Package** POT
- **Property** [OTS_Player](#) \$owner: House owner.
- **Property** int \$paid: Paid time.
- **Property** int \$warnings: Warnings message.
- **Property-read** int \$id: House ID.
- **Property-read** string \$name: House name.
- **Property-read** int \$townId: ID of town where house is located.
- **Property-read** string \$townName: Name of town where house is located.
- **Property-read** int \$rent: Rent cost.
- **Property-read** int \$size: House size.
- **Property-read** [OTS_MapCoords](#) \$entry: Entry point.
- **Property-read** array \$tiles: List of tile points which house uses.
- **Version** 0.1.0

- **Since** 0.1.0

Constructor *void* function OTS_House::__construct(\$element) [*line 59*]

Function Parameters:

- *DOMElement* **\$element** House information.

Creates wrapper for given house element.

Creates wrapper for given house element.

- **Version** 0.1.0
- **Since** 0.1.0
- **Access** public

void function OTS_House::addTile(\$tile) [*line 273*]

Function Parameters:

- [*OTS_MapCoords*](#) **\$tile** Tile to be added.

Adds tile to house.

Adds tile to house.

- **Version** 0.1.0
- **Since** 0.1.0
- **Access** public

void function OTS_House::delete() [*line 101*]

Deletes house info from database.

Deletes house info from database.

- **Version** 0.1.0
- **Since** 0.1.0
- **Access** public

OTS_MapCoords function *OTS_House::getEntry()* [*line 175*]

Returns entry position.

Returns entry position.

- **Version** 0.1.0
- **Since** 0.1.0
- **Access** public

int function *OTS_House::getId()* [*line 115*]

Returns house's ID.

Returns house's ID.

- **Version** 0.1.0
- **Since** 0.1.0
- **Access** public

string function *OTS_House::getName()* [*line 125*]

Return house's name.

Return house's name.

- **Version** 0.1.0
- **Since** 0.1.0
- **Access** public

OTS_Player|null function OTS_House::getOwner() [*line 185*]

Returns current house owner.

Returns current house owner.

- **Version** 0.1.0
- **Since** 0.1.0
- **Access** public

int|false function OTS_House::getPaid() [*line 215*]

Returns paid date.

Returns paid date.

- **Version** 0.1.0
- **Since** 0.1.0
- **Access** public

int function OTS_House::getRent() [*line 155*]

Returns house rent cost.

Returns house rent cost.

- **Version** 0.1.0
- **Since** 0.1.0
- **Access** public

int function OTS_House::getSize() [*line 165*]

Returns house size.

Returns house size.

- **Version** 0.1.0
- **Since** 0.1.0
- **Access** public

array function OTS_House::getTiles() [*line 283*]

Returns tiles list.

Returns tiles list.

- **Version** 0.1.0
- **Since** 0.1.0
- **Access** public

int function OTS_House::getTownId() [*line 135*]

Returns town ID in which house is located.

Returns town ID in which house is located.

- **Version** 0.1.0
- **Since** 0.1.0
- **Access** public

string function OTS_House::getTownName() [*line 145*]

Returns town name.

Returns town name.

- **Version** 0.1.0
- **Since** 0.1.0
- **Access** public

int|false function OTS_House::getWarnings() [*line 244*]

Returns house warnings.

Returns house warnings.

- **Version** 0.1.2
- **Version** 0.1.0
- **Since** 0.1.0
- **Access** public

void function OTS_House::save() [*line 84*]

Saves info in database.

Saves info in database.

- **Version** 0.1.0
- **Since** 0.1.0
- **Access** public

void function OTS_House::setOwner(\$player) [*line 205*]

Function Parameters:

- [*OTS_Player*](#) **\$player** House owner to be set.

Sets house owner.

Sets house owner.

- **Version** 0.1.0
- **Since** 0.1.0
- **Access** public

void function OTS_House::setPaid(\$paid) [*line 233*]

Function Parameters:

- *int* **\$paid** Sets paid timestamp to passed one.

Sets paid date.

Sets paid date.

- **Version** 0.1.0
- **Since** 0.1.0
- **Access** public

void function OTS_House::setWarnings(\$warnings) [*line 263*]

Function Parameters:

- *int* **\$warnings** Sets house warnings.

Sets house warnings.

Sets house warnings.

- **Version** 0.1.2
- **Version** 0.1.0
- **Since** 0.1.0
- **Access** public

mixed function OTS_House::__get(\$name) [*line 295*]

Function Parameters:

- *string* **\$name** Property name.

Magic PHP5 method.

Magic PHP5 method.

- **Version** 0.1.0
- **Throws** OutOfBoundsException For non-supported properties.

- **Since** 0.1.0
- **Access** public

void function OTS_House::__set(\$name, \$value) [*line 344*]

Function Parameters:

- *string* **\$name** Property name.
- *mixed* **\$value** Property value.

Magic PHP5 method.

Magic PHP5 method.

- **Version** 0.1.0
- **Throws** OutOfBoundsException For non-supported properties.
- **Since** 0.1.0
- **Access** public

array function OTS_House::__sleep() [*line 76*]

Magic PHP5 method.

Magic PHP5 method.

Allows object serialisation.

- **Version** 0.1.0
- **Since** 0.1.0
- **Access** public

Class OTS_HousesList

[line 20]

Wrapper for houses list.
Wrapper for houses list.

- **Package** POT
- **Version** 0.1.0
- **Since** 0.1.0

Constructor *void* function OTS_HousesList::__construct(\$path) [line 34]

Function Parameters:

- *string* **\$path** Houses file.

Loads houses information.
Loads houses information.

- **Version** 0.1.0
- **Since** 0.1.0
- **Access** public

int function OTS_HousesList::count() [line 109]

Returns amount of houses.
Returns amount of houses.

- **Version** 0.1.0
- **Since** 0.1.0
- **Access** public

OTS_House|null function *OTS_HousesList::getHouse(\$id)* [line 72]

Function Parameters:

- *int* **\$id** House ID.

Returns house information.

Returns house information.

- **Version** 0.1.0
- **Since** 0.1.0
- **Access** public

int|bool function *OTS_HousesList::getHouseId(\$name)* [line 90]

Function Parameters:

- *string* **\$name** House name.

Returns ID of house with given name.

Returns ID of house with given name.

- **Version** 0.1.0
- **Since** 0.1.0

- **Access** public

ArrayIterator function `OTS_HousesList::getIterator()` [*line 119*]

Returns iterator handle for loops.

Returns iterator handle for loops.

- **Version** 0.1.0
- **Since** 0.1.0
- **Access** public

bool function `OTS_HousesList::offsetExists($offset)` [*line 130*]

Function Parameters:

- *string|int* **\$offset** Array key.

Checks if given element exists.

Checks if given element exists.

- **Version** 0.1.0
- **Since** 0.1.0
- **Access** public

mixed function `OTS_HousesList::offsetGet($offset)` [*line 150*]

Function Parameters:

- *string|int* **\$offset** Array key.

Returns item from given position.

Returns item from given position.

- **Version** 0.1.0
- **Since** 0.1.0
- **Access** public

`void function OTS_HousesList::offsetSet($offset, $value) [line 179]`

Function Parameters:

- *string|int* **\$offset** Array key.
- *mixed* **\$value** Field value.

This method is implemented for **ArrayAccess** interface.

This method is implemented for **ArrayAccess** interface. In fact you can't write/append to houses list. Any call to this method will cause **E_OTS_ReadOnly** raise.

- **Version** 0.1.0
- **Throws** **E_OTS_ReadOnly** Always - this class is read-only.
- **Since** 0.1.0
- **Access** public

`void function OTS_HousesList::offsetUnset($offset) [line 190]`

Function Parameters:

- *string|int* **\$offset** Array key.

This method is implemented for `ArrayAccess` interface.

This method is implemented for `ArrayAccess` interface. In fact you can't write/append to houses list. Any call to this method will cause `E_OTS_ReadOnly` raise.

- **Version** 0.1.0
- **Throws** `E_OTS_ReadOnly` Always - this class is read-only.
- **Since** 0.1.0
- **Access** public

`void function OTS_HousesList::__set_state($properties) [line 53]`

Function Parameters:

- `array $properties` List of object properties.

Magic PHP5 method.

Magic PHP5 method.

Allows object importing from [var_export\(\)](#).

- **Version** 0.1.0
- **Since** 0.1.0
- **Access** public

Class `OTS_InfoRespond`
[line 45]

Wrapper for 'info' respond's DOMDocument.

Wrapper for 'info' respond's DOMDocument.

Note: as this class extends DOMDocument class and contains exactly respond XML tree you can work on it as on normal DOM tree.

- **Package** POT
- **Property-read** string \$tspqVersion: Root element version.
- **Property-read** int \$uptime: Uptime.
- **Property-read** string \$ip: IP number.
- **Property-read** string \$name: Server name.
- **Property-read** int \$port: Server port.
- **Property-read** string \$location: Server physical location.
- **Property-read** string \$url: Website URL.
- **Property-read** string \$server: What the hell...?
- **Property-read** string \$serverVersion: Server version.
- **Property-read** string \$clientVersion: Client version.
- **Property-read** string \$owner: Owner name.
- **Property-read** string \$eMail: Owner's e-mail.
- **Property-read** int \$onlinePlayers: Players online count.
- **Property-read** int \$maxPlayers: Maximum allowed players count.
- **Property-read** int \$playersPeak: Record of players online.
- **Property-read** int \$monstersCount: Number of monsters on map.
- **Property-read** string \$mapName: Map name.
- **Property-read** string \$mapAuthor: Map author.
- **Property-read** int \$mapWidth: Map width.
- **Property-read** int \$mapHeight: Map height.
- **Property-read** string \$motd: Message Of The Day.
- **Version** 0.1.0
- **Version** 0.0.2

- **Since** 0.0.2

string function OTS_InfoRespond::getClientVersion() [*line 144*]

Returns dedicated version of client.

Returns dedicated version of client.

- **Version** 0.0.2
- **Since** 0.0.2
- **Access** public

string function OTS_InfoRespond::getEmail() [*line 164*]

Returns owner e-mail.

Returns owner e-mail.

- **Version** 0.0.2
- **Since** 0.0.2
- **Access** public

string function OTS_InfoRespond::getIP() [*line 72*]

Returns server IP.

Returns server IP.

- **Version** 0.0.2
- **Since** 0.0.2
- **Access** public

string function OTS_InfoRespond::getLocation() [*line 102*]

Returns server location.

Returns server location.

- **Version** 0.0.2
- **Since** 0.0.2
- **Access** public

string function OTS_InfoRespond::getMapAuthor() [*line 225*]

Returns map author.

Returns map author.

- **Version** 0.0.2
- **Since** 0.0.2
- **Access** public

int function OTS_InfoRespond::getMapHeight() [*line 245*]

Returns map height.

Returns map height.

- **Version** 0.0.2
- **Since** 0.0.2
- **Access** public

string function OTS_InfoRespond::getMapName() [*line 214*]

Returns map name.

Returns map name.

- **Version** 0.0.2
- **Since** 0.0.2
- **Access** public

int function OTS_InfoRespond::getMapWidth() [*line 235*]

Returns map width.

Returns map width.

- **Version** 0.0.2
- **Since** 0.0.2
- **Access** public

int function OTS_InfoRespond::getMaxPlayers() [*line 184*]

Returns maximum amount of players online.

Returns maximum amount of players online.

- **Version** 0.0.2
- **Since** 0.0.2
- **Access** public

int function OTS_InfoRespond::getMonstersCount() [*line 204*]

Returns number of all monsters on map.

Returns number of all monsters on map.

- **Version** 0.0.2
- **Since** 0.0.2
- **Access** public

string function OTS_InfoRespond::getMOTD() [*line 256*]

Returns server's Message Of The Day

Returns server's Message Of The Day

- **Version** 0.1.0
- **Version** 0.0.2
- **Since** 0.0.2
- **Access** public

string function OTS_InfoRespond::getName() [*line 82*]

Returns server name.

Returns server name.

- **Version** 0.0.2
- **Since** 0.0.2
- **Access** public

int function OTS_InfoRespond::getOnlinePlayers() [*line 174*]

Returns current amount of players online.

Returns current amount of players online.

- **Version** 0.0.2
- **Since** 0.0.2
- **Access** public

string function OTS_InfoRespond::getOwner() [*line 154*]

Returns owner name.

Returns owner name.

- **Version** 0.0.2
- **Since** 0.0.2
- **Access** public

int function OTS_InfoRespond::getPlayersPeak() [*line 194*]

Returns record of online players.

Returns record of online players.

- **Version** 0.0.2
- **Since** 0.0.2
- **Access** public

int function OTS_InfoRespond::getPort() [*line 92*]

Returns server port.

Returns server port.

- **Version** 0.0.2
- **Since** 0.0.2
- **Access** public

string function OTS_InfoRespond::getServer() [*line 124*]

Returns server attribute.

Returns server attribute.

I have no idea what the hell is it representing :P.

- **Version** 0.0.2
- **Since** 0.0.2
- **Access** public

string function OTS_InfoRespond::getServerVersion() [*line 134*]

Returns server version.

Returns server version.

- **Version** 0.0.2
- **Since** 0.0.2
- **Access** public

string function OTS_InfoRespond::getTSPQVersion() [*line 52*]

Returns version of root element.

Returns version of root element.

- **Version** 0.0.2
- **Since** 0.0.2
- **Access** public

int function OTS_InfoRespond::getUptime() [*line 62*]

Returns server uptime.

Returns server uptime.

- **Version** 0.0.2
- **Since** 0.0.2
- **Access** public

string function OTS_InfoRespond::getURL() [*line 112*]

Returns server website.

Returns server website.

- **Version** 0.0.2
- **Since** 0.0.2
- **Access** public

mixed function OTS_InfoRespond::__get(\$name) [*line 281*]

Function Parameters:

- *string* **\$name** Property name.

Magic PHP5 method.

Magic PHP5 method.

- **Version** 0.1.0
- **Version** 0.0.2
- **Throws** OutOfBoundsException For non-supported properties.
- **Since** 0.1.0
- **Since** 0.0.2
- **Access** public

string function OTS_InfoRespond::__toString() [*line 360*]

Returns string representation of XML.

Returns string representation of XML.

- **Version** 0.1.0
- **Version** 0.0.2
- **Since** 0.0.2
- **Since** 0.1.0
- **Access** public

Class OTS_Item

[line 26]

Single item representation.

Single item representation.

- **Package** POT
- **Property** int \$count: Amount of item.
- **Property** string \$attributes: Attributes binary string.
- **Property-read** int \$id: Item type ID.
- **Property-read** [OTS_ItemType](#)|null \$itemType: Item type instance.
- **Version** 0.1.0
- **Version** 0.0.3
- **Since** 0.0.3

Constructor *void* function OTS_Item::__construct(\$id) [line 54]

Function Parameters:

- *int* **\$id** Item ID.

Creates item of given ID.

Creates item of given ID.

- **Version** 0.0.3
- **Since** 0.0.3
- **Access** public

int function OTS_Item::count() [line 126]

Count value for current item.

Count value for current item.

- **Version** 0.0.3
- **Since** 0.0.3
- **Access** public

string function OTS_Item::getAttributes() [*line 94*]

Returns item custom attributes.

Returns item custom attributes.

- **Version** 0.0.3
- **Since** 0.0.3
- **Access** public

int function OTS_Item::getCount() [*line 74*]

Returns count of item.

Returns count of item.

- **Version** 0.0.3
- **Since** 0.0.3
- **Access** public

int function OTS_Item::getId() [*line 64*]

Returns item type.

Returns item type.

- **Version** 0.0.3
- **Since** 0.0.3
- **Access** public

OTS_ItemType|null function OTS_Item::getItemType() [*line 116*]

Returns type of item.

Returns type of item.

- **Version** 0.1.0
- **Version** 0.0.3
- **Since** 0.0.3
- **Since** 0.1.0
- **Access** public

void function OTS_Item::setAttributes(\$attributes) [*line 104*]

Function Parameters:

- *string* **\$attributes** Item Attributes.

Sets item attributes.

Sets item attributes.

- **Version** 0.0.3
- **Since** 0.0.3

- **Access** public

void function OTS_Item::setCount(\$count) [*line 84*]

Function Parameters:

- *int* **\$count** Count.

Sets count of item.

Sets count of item.

- **Version** 0.0.3
- **Since** 0.0.3
- **Access** public

mixed function OTS_Item::__get(\$name) [*line 140*]

Function Parameters:

- *string* **\$name** Property name.

Magic PHP5 method.

Magic PHP5 method.

- **Version** 0.1.0
- **Version** 0.0.3
- **Throws** OutOfBoundsException For non-supported properties.
- **Since** 0.1.0
- **Since** 0.0.3

- **Access** public

void function OTS_Item::__set(\$name, \$value) [line 170]

Function Parameters:

- *string* **\$name** Property name.
- *mixed* **\$value** Property value.

Magic PHP5 method.

Magic PHP5 method.

- **Version** 0.1.0
- **Version** 0.0.3
- **Throws** OutOfBoundsException For non-supported properties.
- **Since** 0.1.0
- **Since** 0.0.3
- **Access** public

Class OTS_ItemsList

[line 27]

Items list loader.

Items list loader.

- **Package** POT
- **Property-read** int \$otbVersion: OTB file version.
- **Property-read** int \$clientVersion: Dedicated client version.
- **Property-read** int \$buildVersion: File build version.
- **Version** 0.1.0
- **Version** 0.0.8
- **Since** 0.0.8

OTS_ItemsList::CLIENT_VERSION_750

= 1 [*line 37*]

Tibia client 7.

Tibia client 7.5 version.

- **Version** 0.0.8
- **Since** 0.0.8

OTS_ItemsList::CLIENT_VERSION_755

= 2 [*line 41*]

Tibia client 7.

Tibia client 7.55 version.

- **Version** 0.0.8
- **Since** 0.0.8

OTS_ItemsList::CLIENT_VERSION_760

= 3 *[line 45]*

Tibia client 7.

Tibia client 7.6 version.

- **Version** 0.0.8
- **Since** 0.0.8

OTS_ItemsList::CLIENT_VERSION_770

= 3 *[line 49]*

Tibia client 7.

Tibia client 7.7 version.

- **Version** 0.0.8
- **Since** 0.0.8

OTS_ItemsList::CLIENT_VERSION_780

= 4 *[line 53]*

Tibia client 7.

Tibia client 7.8 version.

- **Version** 0.0.8
- **Since** 0.0.8

OTS_ItemsList::CLIENT_VERSION_790

= 5 *[line 57]*

Tibia client 7.

Tibia client 7.9 version.

- **Version** 0.0.8
- **Since** 0.0.8

OTS_ItemsList::CLIENT_VERSION_792

= 6 *[line 61]*

Tibia client 7.

Tibia client 7.92 version.

- **Version** 0.0.8
- **Since** 0.0.8

OTS_ItemsList::CLIENT_VERSION_800

= 7 *[line 65]*

Tibia client 8.

Tibia client 8.0 version.

- **Version** 0.0.8
- **Since** 0.0.8

OTS_ItemsList::ITEM_ATTR_CLIENTID

= 17 [*line 74*]

Client ID.

Client ID.

- **Version** 0.0.8
- **Since** 0.0.8

OTS_ItemsList::ITEM_ATTR_LIGHT2

= 42 [*line 82*]

Light.

Light.

- **Version** 0.0.8
- **Since** 0.0.8

OTS_ItemsList::ITEM_ATTR_SERVERID

= 16 [*line 70*]

Server ID.

Server ID.

- **Version** 0.0.8
- **Since** 0.0.8

OTS_ItemsList::ITEM_ATTR_SPEED

= 20 [*line 78*]

Speed.

Speed.

- **Version** 0.0.8
- **Since** 0.0.8

OTS_ItemsList::ITEM_ATTR_TOPORDER

= 43 [*line 86*]

Always-on-top order.

Always-on-top order.

- **Version** 0.0.8
- **Since** 0.0.8

OTS_ItemsList::ROOT_ATTR_VERSION

= 1 [*line 32*]

Root file attribute.

Root file attribute.

- **Version** 0.0.8
- **Since** 0.0.8

int function OTS_ItemsList::count() [*line 481*]

Returns amount of items loaded.

Returns amount of items loaded.

- **Version** 0.0.8
- **Since** 0.0.8
- **Access** public

string function OTS_ItemsList::current() [*line 492*]

Returns item at current position in iterator.

Returns item at current position in iterator.

- **Version** 0.0.8
- **Deprecated** 0.1.0 Use getIterator().
- **Since** 0.0.8
- **Access** public

int function OTS_ItemsList::getBuildVersion() [*line 417*]

Returns build version.

Returns build version.

- **Version** 0.0.8
- **Since** 0.0.8
- **Access** public

int function OTS_ItemsList::getClientVersion() [*line 407*]

Returns client version.

Returns client version.

- **Version** 0.0.8
- **Since** 0.0.8
- **Access** public

OTS_ItemType|null function OTS_ItemsList::getItemType(\$id) [*line 428*]

Function Parameters:

- *int* **\$id** Item type (server) ID.

Returns given item type.

Returns given item type.

- **Version** 0.0.8
- **Since** 0.0.8
- **Access** public

int|bool function OTS_ItemsList::getItemTypeId(\$name) [*line 450*]

Function Parameters:

- *string* **\$name** Item type name.

Finds item type by it's name.

Finds item type by it's name.

Note: If there are more then one items with same name this function will return first found server ID. It doesn't also mean that it will be the lowest ID - item types are ordered in order that they were loaded from items.xml file.

- **Version** 0.0.8
- **Since** 0.0.8
- **Access** public

array function OTS_ItemsList::getItemTypesList() [*line 471*]

Returns all loaded items.

Returns all loaded items.

- **Version** 0.0.8
- **Deprecated** 0.1.0 Use this class object as array for iterations, counting and methods for field fetching.
- **Since** 0.0.8
- **Access** public

ArrayIterator function OTS_ItemsList::getIterator() [*line 546*]

Returns iterator handle for loops.

Returns iterator handle for loops.

- **Version** 0.1.0
- **Version** 0.0.8
- **Since** 0.0.8

- **Since** 0.1.0
- **Access** public

int function OTS_ItemsList::getOTBVersion() [*line 397*]

Returns OTB file version.

Returns OTB file version.

- **Version** 0.0.8
- **Since** 0.0.8
- **Access** public

int function OTS_ItemsList::key() [*line 513*]

Returns ID of current position.

Returns ID of current position.

- **Version** 0.0.8
- **Deprecated** 0.1.0 Use getIterator().
- **Since** 0.0.8
- **Access** public

void function OTS_ItemsList::loadItems(\$path) [*line 155*]

Function Parameters:

- *string* **\$path** Path to data/items directory.

Loads items.

Loads items.xml and items.otb files.

- **Version** 0.0.8
- **Since** 0.0.8
- **Access** public

void function OTS_ItemsList::next() [*line 502*]

Moves to next iterator item.

Moves to next iterator item.

- **Version** 0.0.8
- **Deprecated** 0.1.0 Use getIterator().
- **Since** 0.0.8
- **Access** public

bool function OTS_ItemsList::offsetExists(\$offset) [*line 559*]

Function Parameters:

- *string|int* **\$offset** Array key.

Checks if given element exists.

Checks if given element exists.

- **Version** 0.1.0
- **Version** 0.0.8
- **Since** 0.0.8

- **Since** 0.1.0
- **Access** public

mixed function OTS_ItemsList::offsetGet(\$offset) [*line 581*]

Function Parameters:

- *string|int* **\$offset** Array key.

Returns item from given position.

Returns item from given position.

- **Version** 0.1.0
- **Version** 0.0.8
- **Since** 0.0.8
- **Since** 0.1.0
- **Access** public

void function OTS_ItemsList::offsetSet(\$offset, \$value) [*line 612*]

Function Parameters:

- *string|int* **\$offset** Array key.
- *mixed* **\$value** Field value.

This method is implemented for ArrayAccess interface.

This method is implemented for ArrayAccess interface. In fact you can't write/append to items list. Any call to this method will cause E_OTs_ReadOnly raise.

- **Version** 0.1.0
- **Version** 0.0.8
- **Throws** E_OTS_ReadOnly Always - this class is read-only.
- **Since** 0.1.0
- **Since** 0.0.8
- **Access** public

void function OTS_ItemsList::offsetUnset(\$offset) [line 625]
Function Parameters:

- *string|int* **\$offset** Array key.

This method is implemented for ArrayAccess interface.

This method is implemented for ArrayAccess interface. In fact you can't write/append to items list. Any call to this method will cause E_OTS_ReadOnly raise.

- **Version** 0.1.0
- **Version** 0.0.8
- **Throws** E_OTS_ReadOnly Always - this class is read-only.
- **Since** 0.1.0
- **Since** 0.0.8
- **Access** public

void function OTS_ItemsList::rewind() [line 534]
Resets iterator index.
 Resets iterator index.

- **Version** 0.0.8
- **Deprecated** 0.1.0 Use getIterator().
- **Since** 0.0.8
- **Access** public

bool function OTS_ItemsList::valid() [*line 524*]

Checks if there is anything more in iterator.

Checks if there is anything more in iterator.

- **Version** 0.0.8
- **Deprecated** 0.1.0 Use getIterator().
- **Since** 0.0.8
- **Access** public

mixed function OTS_ItemsList::__get(\$name) [*line 639*]

Function Parameters:

- *string* **\$name** Property name.

Magic PHP5 method.

Magic PHP5 method.

- **Version** 0.1.0
- **Version** 0.0.8
- **Throws** OutOfBoundsException For non-supported properties.
- **Since** 0.1.0

- **Since** 0.0.8
- **Access** public

void function OTS_ItemsList::__set_state(\$properties) [line 137]

Function Parameters:

- *array* **\$properties** List of object properties.

Magic PHP5 method.

Magic PHP5 method.

Allows object importing from [var_export\(\)](#).

- **Version** 0.0.8
- **Static**
- **Since** 0.0.8
- **Access** public

void function OTS_ItemsList::__wakeup() [line 123]

Magic PHP5 method.

Magic PHP5 method.

Allows object unserialisation.

- **Version** 0.0.8
- **Since** 0.0.8
- **Access** public

Class OTS_ItemType

[line 43]

Item type info.

Item type info.

- **Package** POT
- **Property** int \$clientId: Client ID.
- **Property** string \$name: Item name.
- **Property** int \$group: Group.
- **Property** int \$type: Item type.
- **Property-read** int \$id: Item type ID.
- **Property-read** array \$attributesList: List of all attributes.
- **Property-read** bool \$blocking: Is item blocking move.
- **Property-read** bool \$hasHeight: Does item have height.
- **Property-read** bool \$usable: Is item usable.
- **Property-read** bool \$pickupable: Is player able to pick it up.
- **Property-read** bool \$movable: Can be moved.
- **Property-read** bool \$stackable: Can be stacked.
- **Property-read** bool \$alwaysOnTop: Is always on top of stack.
- **Property-read** bool \$readable: Has readable sign.
- **Property-read** bool \$rotatable: Can be rotated.
- **Property-read** bool \$hangable: Can be hang.
- **Property-read** bool \$vertical: Is vertically oriented.
- **Property-read** bool \$horizontal: Is horizontaly oriented.
- **Property-write** int \$flags: Special flags.
- **Version** 0.1.0
- **Version** 0.0.8

- **Since** 0.0.8

OTS_ItemType::FLAG_ALLOWDISTREAD

= 1048576 [*line 228*]

Can be read from distance.

Can be read from distance.

- **Version** 0.0.8
- **Since** 0.0.8

OTS_ItemType::FLAG_ALWAYSONTOP

= 8192 [*line 200*]

Is always over other items in stack.

Is always over other items in stack.

- **Version** 0.0.8
- **Since** 0.0.8

OTS_ItemType::FLAG_BLOCK_PATHFIND

= 4 [*line 156*]

Can block searching for path.

Can block searching for path.

- **Version** 0.0.8
- **Since** 0.0.8

OTS_ItemType::FLAG_BLOCK_PROJECTILE

= 2 [*line 152*]

BLOCK_PROJECTILE flag(?).
BLOCK_PROJECTILE flag(?).

- **Version** 0.0.8
- **Since** 0.0.8

OTS_ItemType::FLAG_BLOCK_SOLID

= 1 [*line 148*]

Can block characters from walking.
Can block characters from walking.

- **Version** 0.0.8
- **Since** 0.0.8

OTS_ItemType::FLAG_CANNOTDECAY

= 524288 [*line 224*]

Doesn't decay.
Doesn't decay.

- **Version** 0.0.8
- **Since** 0.0.8

OTS_ItemType::FLAG_FLOORCHANGEDOWN

= 256 [*line 180*]

Changes floor under it.

Changes floor under it.

- **Version** 0.0.8
- **Since** 0.0.8

OTS_ItemType::FLAG_FLOORCHANGEEST

= 1024 [*line 188*]

Changes floor east from it's position.

Changes floor east from it's position.

- **Version** 0.0.8
- **Since** 0.0.8

OTS_ItemType::FLAG_FLOORCHANGENORTH

= 512 [*line 184*]

Changes floor north from it's position.

Changes floor north from it's position.

- **Version** 0.0.8
- **Since** 0.0.8

OTS_ItemType::FLAG_FLOORCHANGESOUTH

= 2048 [*line 192*]

Changes floor south from it's position.
Changes floor south from it's position.

- **Version** 0.0.8
- **Since** 0.0.8

OTS_ItemType::FLAG_FLOORCHANGEWEST

= 4096 [*line 196*]

Changes floor west from it's position.
Changes floor west from it's position.

- **Version** 0.0.8
- **Since** 0.0.8

OTS_ItemType::FLAG_HANGABLE

= 65536 [*line 212*]

Can be hang(?).
Can be hang(?).

- **Version** 0.0.8
- **Since** 0.0.8

OTS_ItemType::FLAG_HAS_HEIGHT

= 8 [*line 160*]

Does item rises stack height on it's field.
Does item rises stack height on it's field.

- **Version** 0.0.8
- **Since** 0.0.8

OTS_ItemType::FLAG_HORIZONTAL

= 262144 [*line 220*]

Is oriented horizontally.
Is oriented horizontally.

- **Version** 0.0.8
- **Since** 0.0.8

OTS_ItemType::FLAG_MOVEABLE

= 64 [*line 172*]

Can be moved by player.
Can be moved by player.

- **Version** 0.0.8
- **Since** 0.0.8

OTS_ItemType::FLAG_PICKUPABLE

= 32 *[line 168]*

Can be picked up by player.
Can be picked up by player.

- **Version** 0.0.8
- **Since** 0.0.8

OTS_ItemType::FLAG_READABLE

= 16384 *[line 204]*

Has readable sign.
Has readable sign.

- **Version** 0.0.8
- **Since** 0.0.8

OTS_ItemType::FLAG_ROTABLE

= 32768 *[line 208]*

Can be rotated by player.
Can be rotated by player.

- **Version** 0.0.8
- **Since** 0.0.8

OTS_ItemType::FLAG_STACKABLE

= 128 [*line 176*]

Can be grouped with another items.
Can be grouped with another items.

- **Version** 0.0.8
- **Since** 0.0.8

OTS_ItemType::FLAG_USEABLE

= 16 [*line 164*]

Can be used by players.
Can be used by players.

- **Version** 0.0.8
- **Since** 0.0.8

OTS_ItemType::FLAG_VERTICAL

= 131072 [*line 216*]

Is oriented vertically.
Is oriented vertically.

- **Version** 0.0.8
- **Since** 0.0.8

OTS_ItemType::ITEM_GROUP_AMMUNITION

= 4 *[line 64]*

Ammunition.

Ammunition.

- **Version** 0.0.8
- **Since** 0.0.8

OTS_ItemType::ITEM_GROUP_ARMOR

= 5 *[line 68]*

Armor.

Armor.

- **Version** 0.0.8
- **Since** 0.0.8

OTS_ItemType::ITEM_GROUP_CONTAINER

= 2 *[line 56]*

Container.

Container.

- **Version** 0.0.8
- **Since** 0.0.8

OTS_ItemType::ITEM_GROUP_DEPRECATED

= 14 [*line 107*]

Deprecated item.
Deprecated item.

- **Version** 0.1.0
- **Version** 0.0.8
- **Since** 0.0.8
- **Since** 0.1.0

OTS_ItemType::ITEM_GROUP_DOOR

= 13 [*line 100*]

Door.
Door.

- **Version** 0.0.8
- **Since** 0.0.8

OTS_ItemType::ITEM_GROUP_FLUID

= 12 [*line 96*]

Liquid thing.
Liquid thing.

- **Version** 0.0.8
- **Since** 0.0.8

OTS_ItemType::ITEM_GROUP_GROUND

= 1 [*line 52*]

Ground tile.
Ground tile.

- **Version** 0.0.8
- **Since** 0.0.8

OTS_ItemType::ITEM_GROUP_KEY

= 10 [*line 88*]

Key.
Key.

- **Version** 0.0.8
- **Since** 0.0.8

OTS_ItemType::ITEM_GROUP_MAGICFIELD

= 8 *[line 80]*

Magic field.

Magic field.

- **Version** 0.0.8
- **Since** 0.0.8

OTS_ItemType::ITEM_GROUP_NONE

= 0 *[line 48]*

No group speciffied.

No group speciffied.

- **Version** 0.0.8
- **Since** 0.0.8

OTS_ItemType::ITEM_GROUP_RUNE

= 6 *[line 72]*

Rune.

Rune.

- **Version** 0.0.8
- **Since** 0.0.8

OTS_ItemType::ITEM_GROUP_SPLASH

= 11 [*line 92*]

Splash effect.

Splash effect.

- **Version** 0.0.8
- **Since** 0.0.8

OTS_ItemType::ITEM_GROUP_TELEPORT

= 7 [*line 76*]

Teleport field.

Teleport field.

- **Version** 0.0.8
- **Since** 0.0.8

OTS_ItemType::ITEM_GROUP_WEAPON

= 3 [*line 60*]

Weapon.

Weapon.

- **Version** 0.0.8
- **Since** 0.0.8

OTS_ItemType::ITEM_GROUP_WRITEABLE

= 9 [*line 84*]

Item that can store editable sign.

Item that can store editable sign.

- **Version** 0.0.8
- **Since** 0.0.8

OTS_ItemType::ITEM_TYPE_CONTAINER

= 4 [*line 128*]

Container.

Container.

- **Version** 0.0.8
- **Since** 0.0.8

OTS_ItemType::ITEM_TYPE_DEPOT

= 1 [*line 116*]

Depot locker.

Depot locker.

- **Version** 0.0.8
- **Since** 0.0.8

OTS_ItemType::ITEM_TYPE_DOOR

= 5 [*line 132*]

Door.

Door.

- **Version** 0.0.8
- **Since** 0.0.8

OTS_ItemType::ITEM_TYPE_MAGICFIELD

= 6 [*line 136*]

Magic field.

Magic field.

- **Version** 0.0.8
- **Since** 0.0.8

OTS_ItemType::ITEM_TYPE_MAILBOX

= 2 [*line 120*]

Mailbox.

Mailbox.

- **Version** 0.0.8

- **Since** 0.0.8

OTS_ItemType::ITEM_TYPE_NONE

= 0 [*line 112*]

No special type.
No special type.

- **Version** 0.0.8
- **Since** 0.0.8

OTS_ItemType::ITEM_TYPE_TELEPORT

= 7 [*line 143*]

Teleport.
Teleport.

- **Version** 0.1.0
- **Version** 0.0.8
- **Since** 0.0.8
- **Since** 0.1.0

OTS_ItemType::ITEM_TYPE_TRASHHOLDER

= 3 [*line 124*]

Trash can.
Trash can.

- **Version** 0.0.8
- **Since** 0.0.8

Constructor *void* function OTS_ItemType::__construct(\$id) [*line 284*]

Function Parameters:

- *int* **\$id** Server ID.

Initializes new item type object.

Initializes new item type object.

- **Version** 0.0.8
- **Since** 0.0.8
- **Access** public

OTS_Item function OTS_ItemType::createItem() [*line 574*]

Creates instance of this type.

Creates instance of this type.

- **Version** 0.0.8
- **Since** 0.0.8
- **Access** public

string|null function OTS_ItemType::getAttribute(\$name, \$attribyte) [*line 366*]

Function Parameters:

- *string* **\$attribbyte** Attribute name.
- **\$name**

Returns given attribute.

Returns given attribute.

- **Version** 0.0.8
- **Since** 0.0.8
- **Access** public

array function `OTS_ItemType::getAttributesList()` [*line 394*]

Returns all attributes list.

Returns all attributes list.

- **Version** 0.0.8
- **Since** 0.0.8
- **Access** public

int function `OTS_ItemType::getClientId()` [*line 325*]

Returns item type client ID.

Returns item type client ID.

- **Version** 0.0.8
- **Since** 0.0.8

- **Access** public

int function OTS_ItemType::getGroup() [*line 404*]

Returns group.

Returns group.

- **Version** 0.0.8
- **Since** 0.0.8
- **Access** public

int function OTS_ItemType::getId() [*line 315*]

Returns item type server ID.

Returns item type server ID.

- **Version** 0.0.8
- **Since** 0.0.8
- **Access** public

string function OTS_ItemType::getName() [*line 345*]

Returns item name.

Returns item name.

- **Version** 0.0.8
- **Since** 0.0.8
- **Access** public

int function OTS_ItemType::getType() [*line 424*]

Returns item type.

Returns item type.

- **Version** 0.0.8
- **Since** 0.0.8
- **Access** public

bool function OTS_ItemType::hasHeight() [*line 464*]

Checks if item has height.

Checks if item has height.

- **Version** 0.0.8
- **Since** 0.0.8
- **Access** public

bool function OTS_ItemType::isAlwaysOnTop() [*line 514*]

Checks if item is always on top.

Checks if item is always on top.

- **Version** 0.0.8
- **Since** 0.0.8
- **Access** public

bool function OTS_ItemType::isBlocking() [*line 454*]

Checks if item is blocking.

Checks if item is blocking.

- **Version** 0.0.8
- **Since** 0.0.8
- **Access** public

bool function OTS_ItemType::isHangable() [*line 544*]

Checks if item can be hanged.

Checks if item can be hanged.

- **Version** 0.0.8
- **Since** 0.0.8
- **Access** public

bool function OTS_ItemType::isHorizontal() [*line 564*]

Checks if item is horizontal.

Checks if item is horizontal.

- **Version** 0.0.8
- **Since** 0.0.8
- **Access** public

bool function OTS_ItemType::isMovable() [*line 494*]

Checks if item is movable.

Checks if item is movable.

- **Version** 0.0.8
- **Since** 0.0.8
- **Access** public

bool function OTS_ItemType::isPickupable() [line 484]

Checks if item is pickupable.

Checks if item is pickupable.

- **Version** 0.0.8
- **Since** 0.0.8
- **Access** public

bool function OTS_ItemType::isReadable() [line 524]

Checks if item is readable.

Checks if item is readable.

- **Version** 0.0.8
- **Since** 0.0.8
- **Access** public

bool function OTS_ItemType::isRotable() [line 534]

Checks if item can be rotated.

Checks if item can be rotated.

- **Version** 0.0.8
- **Since** 0.0.8
- **Access** public

bool function OTS_ItemType::isStackable() [*line 504*]

Checks if item is stackable.

Checks if item is stackable.

- **Version** 0.0.8
- **Since** 0.0.8
- **Access** public

bool function OTS_ItemType::isUsable() [*line 474*]

Checks if item is usable.

Checks if item is usable.

- **Version** 0.0.8
- **Since** 0.0.8
- **Access** public

bool function OTS_ItemType::isVertical() [*line 554*]

Checks if item is vertical.

Checks if item is vertical.

- **Version** 0.0.8
- **Since** 0.0.8
- **Access** public

void function OTS_ItemType::setAttribute(\$name, \$value, \$attribute) [line 384]

Function Parameters:

- *string* **\$attribute** Attribute name.
- *string* **\$value** Attribute value.
- **\$name**

Sets given attribute.

Sets given attribute.

- **Version** 0.0.8
- **Since** 0.0.8
- **Access** public

void function OTS_ItemType::setClientId(\$clientId) [line 335]

Function Parameters:

- *int* **\$clientId** Client ID.

Sets client side ID.

Sets client side ID.

- **Version** 0.0.8
- **Since** 0.0.8
- **Access** public

void function OTS_ItemType::setFlags(\$flags) [line 444]

Function Parameters:

- *int* **\$flags** Flags.

Sets type flags.

Sets type flags.

- **Version** 0.0.8
- **Since** 0.0.8
- **Access** public

void function OTS_ItemType::setGroup(\$group) [line 414]

Function Parameters:

- *int* **\$group** Group.

Sets item group.

Sets item group.

- **Version** 0.0.8
- **Since** 0.0.8
- **Access** public

void function OTS_ItemType::setName(\$name) [*line 355*]

Function Parameters:

- *string* **\$name** Name.

Sets item type name.

Sets item type name.

- **Version** 0.0.8
- **Since** 0.0.8
- **Access** public

void function OTS_ItemType::setType(\$type) [*line 434*]

Function Parameters:

- *int* **\$type** Type.

Sets item type.

Sets item type.

- **Version** 0.0.8
- **Since** 0.0.8
- **Access** public

mixed function OTS_ItemType::__get(\$name) [*line 597*]

Function Parameters:

- *string* **\$name** Property name.

Magic PHP5 method.

Magic PHP5 method.

- **Version** 0.1.0
- **Version** 0.0.8
- **Throws** OutOfBoundsException For non-supported properties.
- **Since** 0.1.0
- **Since** 0.0.8
- **Access** public

void function OTS_ItemType::__set(\$name, \$value) [*line 669*]

Function Parameters:

- *string* **\$name** Property name.
- *mixed* **\$value** Property value.

Magic PHP5 method.

Magic PHP5 method.

- **Version** 0.1.0
- **Version** 0.0.8

- **Throws** OutOfBoundsException For non-supported properties.
- **Since** 0.1.0
- **Since** 0.0.8
- **Access** public

void function OTS_ItemType::__set_state(\$properties) [*line 297*]

Function Parameters:

- **array \$properties** List of object properties.

Magic PHP5 method.

Magic PHP5 method.

Allows object importing from [var_export\(\)](#).

- **Version** 0.0.8
- **Static**
- **Since** 0.0.8
- **Access** public

Class OTS_MapCoords

[*line 27*]

Map position point.

Map position point.

- **Package** POT
- **Property-read** int \$x: X coord.
- **Property-read** int \$y: Y coord.
- **Property-read** int \$z: Z coord.
- **Version** 0.1.0
- **Version** 0.0.6
- **Since** 0.0.6

Constructor *void* function `OTS_MapCoords::__construct($x, $y, $z)` [line 57]

Function Parameters:

- *int* **\$x** X.
- *int* **\$y** Y.
- *int* **\$z** Z.

Sets coords for point.

Sets coords for point.

- **Version** 0.0.6
- **Since** 0.0.6
- **Access** public

int function `OTS_MapCoords::getX()` [line 82]

Returns X.

Returns X.

- **Version** 0.0.6
- **Since** 0.0.6
- **Access** public

int function OTS_MapCoords::getY() [*line 92*]

Returns Y.
Returns Y.

- **Version** 0.0.6
- **Since** 0.0.6
- **Access** public

int function OTS_MapCoords::getZ() [*line 102*]

Returns Z.
Returns Z.

- **Version** 0.0.6
- **Since** 0.0.6
- **Access** public

mixed function OTS_MapCoords::__get(\$name) [*line 116*]

Function Parameters:

- *string* **\$name** Property name.

Magic PHP5 method.
Magic PHP5 method.

- **Version** 0.1.0
- **Version** 0.0.6
- **Throws** OutOfBoundsException For non-supported properties.
- **Since** 0.1.0
- **Since** 0.0.6
- **Access** public

void function OTS_MapCoords::__set_state(\$properties) [*line 72*]

Function Parameters:

- **array \$properties** List of object properties.

Magic PHP5 method.

Magic PHP5 method.

Allows object importing from [var_export\(\)](#).

- **Version** 0.0.6
- **Static**
- **Since** 0.0.6
- **Access** public

Class OTS_Monster

[*line 38*]

Wrapper for monsters files DOMDocument.

Wrapper for monsters files DOMDocument.

Note: as this class extends DOMDocument class and contains exactly file XML tree you can work on it as on normal DOM tree.

- **Package** POT
- **Property-read** string \$name: Monster name.
- **Property-read** string \$race: Monster race.
- **Property-read** int \$experience: Experience for killing monster.
- **Property-read** int \$speed: Monster speed.
- **Property-read** int|bool \$manaCost: Mana required (false if not possible).
- **Property-read** int \$health: Hit points.
- **Property-read** array \$flags: Flags.
- **Property-read** array \$voices: List of sounds.
- **Property-read** array \$items: List of possible loot.
- **Property-read** array \$immunities: List of immunities.
- **Property-read** int \$defense: Defense rate.
- **Property-read** int \$armor: Armor rate.
- **Property-read** array \$defenses: List of defenses.
- **Property-read** array \$attacks: List of attacks.
- **Version** 0.1.0
- **Version** 0.0.6
- **Since** 0.0.6

int function OTS_Monster::getArmor() [*line 320*]

Returns monster armor.

Returns monster armor.

- **Version** 0.0.6
- **Since** 0.0.6
- **Access** public

array function OTS_Monster::getAttacks() [*line 361*]

Returns list of monster attacks.

Returns list of monster attacks.

- **Version** 0.0.6
- **Since** 0.0.6
- **Access** public

int function OTS_Monster::getDefense() [*line 302*]

Returns monster defense rate.

Returns monster defense rate.

- **Version** 0.0.6
- **Since** 0.0.6
- **Access** public

array function OTS_Monster::getDefenses() [*line 338*]

Returns list of special defenses.

Returns list of special defenses.

- **Version** 0.0.6
- **Since** 0.0.6
- **Access** public

int function OTS_Monster::getExperience() [*line 65*]

Returns amount of experience for killing this monster.

Returns amount of experience for killing this monster.

- **Version** 0.0.6
- **Since** 0.0.6
- **Access** public

int|bool function OTS_Monster::getFlag(\$flag) [*line 134*]

Function Parameters:

- *string* **\$flag** Flag.

Returns specified flag value.

Returns specified flag value.

- **Version** 0.0.6
- **Since** 0.0.6
- **Access** public

array function OTS_Monster::getFlags() [*line 113*]

Returns all monster flags (in format flagname => value).

Returns all monster flags (in format flagname => value).

- **Version** 0.0.6
- **Since** 0.0.6
- **Access** public

int function OTS_Monster::getHealth() [*line 103*]

Returns monster HP.

Returns monster HP.

- **Version** 0.0.6
- **Since** 0.0.6
- **Access** public

array function OTS_Monster::getImmunities() [*line 245*]

Returns all monster immunities.

Returns all monster immunities.

- **Version** 0.0.6
- **Since** 0.0.6
- **Access** public

array function OTS_Monster::getItems() [*line 212*]

Returns all possible loot.

Returns all possible loot.

- **Version** 0.1.0
- **Version** 0.0.6
- **Since** 0.0.6
- **Since** 0.1.0
- **Access** public

array function OTS_Monster::getLoot() [*line 180*]

Returns all possible loot.

Returns all possible loot.

- **Version** 0.0.6
- **Deprecated** 0.1.0 Use getItem().
- **Since** 0.0.6
- **Access** public

int|bool function OTS_Monster::getManaCost() [*line 85*]

Returns amount of mana required to summon this monster.

Returns amount of mana required to summon this monster.

- **Version** 0.0.6
- **Since** 0.0.6
- **Access** public

string function OTS_Monster::getName() [*line 45*]

Returns monster name.

Returns monster name.

- **Version** 0.0.6
- **Since** 0.0.6
- **Access** public

string function OTS_Monster::getRace() [*line 55*]

Returns monster race.

Returns monster race.

- **Version** 0.0.6
- **Since** 0.0.6
- **Access** public

int function OTS_Monster::getSpeed() [*line 75*]

Returns monster speed.

Returns monster speed.

- **Version** 0.0.6
- **Since** 0.0.6
- **Access** public

array function OTS_Monster::getVoices() [*line 155*]

Returns voices that monster can sound.

Returns voices that monster can sound.

- **Version** 0.0.6
- **Since** 0.0.6
- **Access** public

bool function OTS_Monster::hasImmunity(\$name) [*line 276*]

Function Parameters:

- *string* **\$name** Immunity to check.

Checks if monster has given immunity.

Checks if monster has given immunity.

- **Version** 0.0.6
- **Since** 0.0.6
- **Access** public

mixed function OTS_Monster::__get(\$name) [*line 388*]

Function Parameters:

- *string* **\$name** Property name.

Magic PHP5 method.

Magic PHP5 method.

- **Version** 0.1.0
- **Version** 0.0.6
- **Throws** OutOfBoundsException For non-supported properties.
- **Since** 0.1.0
- **Since** 0.0.6
- **Access** public

string function OTS_Monster::__toString() [*line 446*]

Returns string representation of XML.

Returns string representation of XML.

- **Version** 0.1.0
- **Version** 0.0.6
- **Since** 0.0.6
- **Since** 0.1.0
- **Access** public

Class OTS_MonstersList

[*line 20*]

Wrapper for monsters list.

Wrapper for monsters list.

- **Package** POT

- **Version** 0.1.0
- **Since** 0.1.0

Constructor *void* function OTS_MonstersList::__construct(\$path) [*line 41*]

Function Parameters:

- *string* **\$path** Monsters directory.

Loads monsters mapping file.

Loads monsters mapping file.

- **Version** 0.1.0
- **Since** 0.1.0
- **Access** public

int function OTS_MonstersList::count() [*line 110*]

Returns amount of monsters loaded.

Returns amount of monsters loaded.

- **Version** 0.1.0
- **Since** 0.1.0
- **Access** public

OTS_Monster function OTS_MonstersList::current() [*line 120*]

Returns monster at current position in iterator.

Returns monster at current position in iterator.

- **Version** 0.1.0
- **Since** 0.1.0
- **Access** public

OTS_Monster|null function OTS_MonstersList::getMonster(\$name) [line 89]

Function Parameters:

- *string* **\$name** Monster name.

Returns loaded data of given monster.

Returns loaded data of given monster.

- **Version** 0.1.0
- **Since** 0.1.0
- **Access** public

string function OTS_MonstersList::key() [line 138]

Returns name of current position.

Returns name of current position.

- **Version** 0.1.0
- **Since** 0.1.0
- **Access** public

void function OTS_MonstersList::next() [*line 128*]

Moves to next iterator monster.

Moves to next iterator monster.

- **Version** 0.1.0
- **Since** 0.1.0
- **Access** public

bool function OTS_MonstersList::offsetExists(\$offset) [*line 167*]

Function Parameters:

- *string* **\$offset** Array key.

Checks if given element exists.

Checks if given element exists.

- **Version** 0.1.0
- **Since** 0.1.0
- **Access** public

OTS_Monster|bool function OTS_MonstersList::offsetGet(\$offset) [*line 178*]

Function Parameters:

- *string* **\$offset** Array key.

Returns item from given position.

Returns item from given position.

- **Version** 0.1.0
- **Since** 0.1.0
- **Access** public

void function OTS_MonstersList::offsetSet(\$offset, \$value) [*line 198*]

Function Parameters:

- *string|int* **\$offset** Array key.
- *mixed* **\$value** Field value.

This method is implemented for ArrayAccess interface.

This method is implemented for ArrayAccess interface. In fact you can't write/append to monsters list. Any call to this method will cause E_OTS_ReadOnly raise.

- **Version** 0.1.0
- **Throws** E_OTS_ReadOnly Always - this class is read-only.
- **Since** 0.1.0
- **Access** public

void function OTS_MonstersList::offsetUnset(\$offset) [*line 209*]

Function Parameters:

- *string|int* **\$offset** Array key.

This method is implemented for ArrayAccess interface.

This method is implemented for ArrayAccess interface. In fact you can't write/append to monsters list. Any call to this method will cause E_OTS_ReadOnly raise.

- **Version** 0.1.0
- **Throws** E_OTS_ReadOnly Always - this class is read-only.
- **Since** 0.1.0
- **Access** public

void function OTS_MonstersList::rewind() [*line 156*]

Resets iterator index.
Resets iterator index.

- **Version** 0.1.0
- **Since** 0.1.0
- **Access** public

bool function OTS_MonstersList::valid() [*line 148*]

Checks if there is anything more in interator.
Checks if there is anything more in interator.

- **Version** 0.1.0
- **Since** 0.1.0
- **Access** public

void function OTS_MonstersList::__set_state(\$properties) [*line 70*]

Function Parameters:

- **array \$properties** List of object properties.

Magic PHP5 method.

Magic PHP5 method.

Allows object importing from [var_export\(\)](#).

- **Version** 0.1.0
- **Since** 0.1.0
- **Access** public

Class OTS_OTBMFile

[line 30]

OTBM format reader.

OTBM format reader.

- **Package** POT
- **Property-read** [OTS_HousesList](#) \$housesList: Houses list loaded from associated houses file.
- **Property-read** int \$width: Map width.
- **Property-read** int \$height: Map height.
- **Property-read** string \$description: Map description.
- **Version** 0.1.0
- **Version** 0.0.6
- **Since** 0.0.6

OTS_OTBMFile::OTBM_ATTR_ACTION_ID

= 4 [*line 47*]

Action ID.

Action ID.

- **Version** 0.0.6
- **Since** 0.0.6

OTS_OTBMFile::OTBM_ATTR_DEPOT_ID

= 10 [*line 71*]

Depot ID.

Depot ID.

- **Version** 0.0.6
- **Since** 0.0.6

OTS_OTBMFile::OTBM_ATTR_DESC

= 7 [*line 59*]

Description.

Description.

- **Version** 0.0.6
- **Since** 0.0.6

OTS_OTBMFile::OTBM_ATTR_DESCRIPTION

= 1 [*line 35*]

Description attribute.

Description attribute.

- **Version** 0.0.6
- **Since** 0.0.6

OTS_OTBMFile::OTBM_ATTR_EXT_FILE

= 2 [*line 39*]

External file.

External file.

- **Version** 0.0.6
- **Since** 0.0.6

OTS_OTBMFile::OTBM_ATTR_EXT_HOUSE_FILE

= 13 [*line 83*]

External houses file.

External houses file.

- **Version** 0.0.6

- **Since** 0.0.6

OTS_OTBMFile::OTBM_ATTR_EXT_SPAWN_FILE

= 11 [*line 75*]

External spawns file.

External spawns file.

- **Version** 0.0.6
- **Since** 0.0.6

OTS_OTBMFile::OTBM_ATTR_HOUSEDOORID

= 14 [*line 87*]

ID of doors.

ID of doors.

- **Version** 0.0.6
- **Since** 0.0.6

OTS_OTBMFile::OTBM_ATTR_ITEM

= 9 [*line 67*]

Item.

Item.

- **Version** 0.0.6

- **Since** 0.0.6

OTS_OTBMFile::OTBM_ATTR_RUNE_CHARGES

= 12 [*line 79*]

Rune changes amount.
Rune changes amount.

- **Version** 0.0.6
- **Since** 0.0.6

OTS_OTBMFile::OTBM_ATTR_TELE_DEST

= 8 [*line 63*]

Teleport destination.
Teleport destination.

- **Version** 0.0.6
- **Since** 0.0.6

OTS_OTBMFile::OTBM_ATTR_TEXT

= 6 [*line 55*]

Text.
Text.

- **Version** 0.0.6
- **Since** 0.0.6

OTS_OTBMFile::OTBM_ATTR_TILE_FLAGS

= 3 [*line 43*]

Tile flags.
Tile flags.

- **Version** 0.0.6
- **Since** 0.0.6

OTS_OTBMFile::OTBM_ATTR_UNIQUE_ID

= 5 [*line 51*]

Unique ID.
Unique ID.

- **Version** 0.0.6
- **Since** 0.0.6

OTS_OTBMFile::OTBM_NODE_HOUSED_TILE

= 14 [*line 144*]

Tile of house.
Tile of house.

- **Version** 0.0.6
- **Since** 0.0.6

OTS_OTBMFile::OTBM_NODE_ITEM

= 6 [*line 112*]

Item.

Item.

- **Version** 0.0.6
- **Since** 0.0.6

OTS_OTBMFile::OTBM_NODE_ITEM_DEF

= 3 [*line 100*]

Item definition.

Item definition.

- **Version** 0.0.6
- **Since** 0.0.6

OTS_OTBMFile::OTBM_NODE_MAP_DATA

= 2 [*line 96*]

Map data container.

Map data container.

- **Version** 0.0.6
- **Since** 0.0.6

OTS_OTBMFile::OTBM_NODE_MONSTER

= 11 [*line 132*]

Monster.

Monster.

- **Version** 0.0.6
- **Since** 0.0.6

OTS_OTBMFile::OTBM_NODE_ROOTV1

= 1 [*line 92*]

Root node.

Root node.

- **Version** 0.0.6
- **Since** 0.0.6

OTS_OTBMFile::OTBM_NODE_SPAWNS

= 9 [*line 124*]

Spawns container.

Spawns container.

- **Version** 0.0.6
- **Since** 0.0.6

OTS_OTBMFile::OTBM_NODE_SPAWN_AREA

= 10 [*line 128*]

Spawn.

Spawn.

- **Version** 0.0.6
- **Since** 0.0.6

OTS_OTBMFile::OTBM_NODE_TILE

= 5 [*line 108*]

Single tile.

Single tile.

- **Version** 0.0.6
- **Since** 0.0.6

OTS_OTBMFile::OTBM_NODE_TILE_AREA

= 4 [*line 104*]

Map tiles fragment.

Map tiles fragment.

- **Version** 0.0.6
- **Since** 0.0.6

OTS_OTBMFile::OTBM_NODE_TILE_REF

= 8 [*line 120*]

Tile reference.

Tile reference.

- **Version** 0.0.6
- **Since** 0.0.6

OTS_OTBMFile::OTBM_NODE_TILE_SQUARE

= 7 [*line 116*]

Tile.

Tile.

- **Version** 0.0.6
- **Since** 0.0.6

OTS_OTBMFile::OTBM_NODE_TOWN

= 13 [*line 140*]

Town.

Town.

- **Version** 0.0.6
- **Since** 0.0.6

OTS_OTBMFile::OTBM_NODE_TOWNS

= 12 [*line 136*]

Towns container.

Towns container.

- **Version** 0.0.6
- **Since** 0.0.6

int function OTS_OTBMFile::count() [*line 482*]

Returns amount of towns loaded.

Returns amount of towns loaded.

- **Version** 0.0.8
- **Version** 0.0.6
- **Since** 0.0.6
- **Since** 0.0.8
- **Access** public

string function OTS_OTBMFile::current() [*line 495*]

Returns town at current position in iterator.

Returns town at current position in iterator.

- **Version** 0.0.8
- **Version** 0.0.6
- **Deprecated** 0.1.0 Use getIterator().
- **Since** 0.0.8
- **Since** 0.0.6
- **Access** public

string function OTS_OTBMFile::getDescription() [*line 412*]

Returns map description.

Returns map description.

- **Version** 0.0.6
- **Since** 0.0.6
- **Access** public

int function OTS_OTBMFile::getHeight() [*line 402*]

Returns map height.

Returns map height.

- **Version** 0.0.6
- **Since** 0.0.6
- **Access** public

OTS_HousesList function OTS_OTBMFile::getHousesList() [*line 382*]

Loads map's houses list.

Loads map's houses list.

- **Version** 0.1.0
- **Version** 0.0.6
- **Since** 0.0.6
- **Since** 0.1.0
- **Access** public

ArrayIterator function OTS_OTBMFile::getIterator() [*line 557*]

Returns iterator handle for loops.

Returns iterator handle for loops.

- **Version** 0.1.0
- **Version** 0.0.6
- **Since** 0.0.6
- **Since** 0.1.0
- **Access** public

int|bool function OTS_OTBMFile::getTownID(\$name) [*line 423*]

Function Parameters:

- *string* **\$name** Town.

Returns town's ID.

Returns town's ID.

- **Version** 0.0.6
- **Since** 0.0.6
- **Access** public

string|bool function OTS_OTBMFile::getTownName(\$id) [*line 434*]

Function Parameters:

- *int* **\$id** Town ID.

Returns name of given town's ID.

Returns name of given town's ID.

- **Version** 0.0.6
- **Since** 0.0.6
- **Access** public

array function OTS_OTBMFile::getTownsList() [*line 452*]

Returns list (id => name) of loaded towns.

Returns list (id => name) of loaded towns.

- **Version** 0.0.6
- **Deprecated** 0.1.0 Use this class object as array for iterations, counting and methods for field fetching.
- **Since** 0.0.6
- **Access** public

OTS_MapCoords|bool function OTS_OTBMFile::getTownTemple(\$id) [*line 463*]

Function Parameters:

- *int* **\$id** Town id.

Returns town's temple position.

Returns town's temple position.

- **Version** 0.0.6
- **Since** 0.0.6
- **Access** public

int function OTS_OTBMFile::getWidth() [*line 392*]

Returns map width.

Returns map width.

- **Version** 0.0.6
- **Since** 0.0.6
- **Access** public

int function OTS_OTBMFile::key() [*line 520*]

Returns ID of current position.

Returns ID of current position.

- **Version** 0.0.8
- **Version** 0.0.6
- **Deprecated** 0.1.0 Use getIterator().
- **Since** 0.0.8
- **Since** 0.0.6
- **Access** public

void function OTS_OTBMFile::loadFile(\$file) [line 235]
Function Parameters:

- *string* **\$file** Filename.

Loads OTBM file content.
 Loads OTBM file content.

- **Version** 0.1.0
- **Version** 0.0.6
- **Since** 0.0.6
- **Access** public

void function OTS_OTBMFile::next() [line 507]
Moves to next iterator town.
 Moves to next iterator town.

- **Version** 0.0.8
- **Version** 0.0.6

- **Deprecated** 0.1.0 Use getIterator().
- **Since** 0.0.8
- **Since** 0.0.6
- **Access** public

bool function OTS_OTBMFile::offsetExists(\$offset) [*line 570*]

Function Parameters:

- *string|int* **\$offset** Array key.

Checks if given element exists.

Checks if given element exists.

- **Version** 0.1.0
- **Version** 0.0.6
- **Since** 0.0.6
- **Since** 0.1.0
- **Access** public

mixed function OTS_OTBMFile::offsetGet(\$offset) [*line 592*]

Function Parameters:

- *string|int* **\$offset** Array key.

Returns item from given position.

Returns item from given position.

- **Version** 0.1.0
- **Version** 0.0.6
- **Since** 0.0.6
- **Since** 0.1.0
- **Access** public

void function OTS_OTBMFile::offsetSet(\$offset, \$value) [*line 623*]

Function Parameters:

- *string|int* **\$offset** Array key.
- *mixed* **\$value** Field value.

This method is implemented for ArrayAccess interface.

This method is implemented for ArrayAccess interface. In fact you can't write/append to towns list. Any call to this method will cause E_OTS_ReadOnly raise.

- **Version** 0.1.0
- **Version** 0.0.6
- **Throws** E_OTS_ReadOnly Always - this class is read-only.
- **Since** 0.1.0
- **Since** 0.0.6
- **Access** public

void function OTS_OTBMFile::offsetUnset(\$offset) [*line 636*]

Function Parameters:

- *string|int* **\$offset** Array key.

This method is implemented for ArrayAccess interface.

This method is implemented for ArrayAccess interface. In fact you can't write/append to towns list. Any call to this method will cause E_OTS_ReadOnly raise.

- **Version** 0.1.0
- **Version** 0.0.6
- **Throws** E_OTS_ReadOnly Always - this class is read-only.
- **Since** 0.1.0
- **Since** 0.0.6
- **Access** public

void function OTS_OTBMFile::rewind() [*line 545*]

Resets iterator index.

Resets iterator index.

- **Version** 0.0.8
- **Version** 0.0.6
- **Deprecated** 0.1.0 Use getIterator().
- **Since** 0.0.8
- **Since** 0.0.6
- **Access** public

bool function OTS_OTBMFile::valid() [*line 533*]

Checks if there is anything more in iterator.

Checks if there is anything more in iterator.

- **Version** 0.0.8
- **Version** 0.0.6
- **Deprecated** 0.1.0 Use getIterator().
- **Since** 0.0.8
- **Since** 0.0.6
- **Access** public

mixed function OTS_OTBMFile::__get(\$name) [*line 650*]

Function Parameters:

- *string* **\$name** Property name.

Magic PHP5 method.

Magic PHP5 method.

- **Version** 0.1.0
- **Version** 0.0.6
- **Throws** OutOfBoundsException For non-supported properties.
- **Since** 0.1.0
- **Since** 0.0.6
- **Access** public

void function OTS_OTBMFile::__set_state(\$properties) [*line 216*]

Function Parameters:

- *array* **\$properties** List of object properties.

Magic PHP5 method.

Magic PHP5 method.

Allows object importing from [var_export\(\)](#).

- **Version** 0.0.6
- **Static**
- **Since** 0.0.6
- **Access** public

void function OTS_OTBMFile::__wakeup() [line 202]

Magic PHP5 method.

Magic PHP5 method.

Allows object unserialisation.

- **Version** 0.0.6
- **Since** 0.0.6
- **Access** public

Class OTS_Player

[line 66]

OTServ character abstraction.

OTServ character abstraction.

- **Package** POT

- **Property** string \$name: Character name.
- **Property** [OTS_Account](#) \$account: Account to which character belongs.
- **Property** [OTS_Group](#) \$group: Group of which character is member.
- **Property** int \$premiumEnd: Timestamp of PACC end.
- **Property** int \$sex: Gender.
- **Property** int \$vocation: Vocation.
- **Property** int \$experience: Experience points.
- **Property** int \$level: Experience level.
- **Property** int \$magLevel: Magic level.
- **Property** int \$health: Hit points.
- **Property** int \$healthMax: Maximum hit points.
- **Property** int \$mana: Mana.
- **Property** int \$manaMax: Maximum mana.
- **Property** int \$manaSpent: Spent mana.
- **Property** int \$soul: Soul points.
- **Property** int \$direction: Looking direction.
- **Property** int \$lookBody: Body color.
- **Property** int \$lookFeet: Feet color.
- **Property** int \$lookHead: Hairs color.
- **Property** int \$lookLegs: Legs color.
- **Property** int \$lookType: Outfit type.
- **Property** int \$lookAddons: Addons.
- **Property** int \$posX: Spawn X coord.
- **Property** int \$posY: Spawn Y coord.
- **Property** int \$posZ: Spawn Z coord.
- **Property** int \$cap: Capacity.
- **Property** int \$lastLogin: Last login timestamp.
- **Property** int \$lastIP: Last login IP number.
- **Property** string \$conditions: Binary conditions.
- **Property** int \$redSkullTime: Timestamp for which red skull will last.

- **Property** string \$guildNick:
- **Property** [OTS_GuildRank](#) \$rank:
- **Property** int \$townId:
- **Property** int \$lossExperience:
- **Property** int \$lossMana:
- **Property** int \$lossSkills:
- **Property** int \$balance: Bank balance.
- **Property** bool \$save: Player save flag.
- **Property** bool \$redSkull: Player red skull flag.
- **Property** bool \$banned: Player banned state.
- **Property-read** int \$id: Player ID.
- **Property-read** bool \$loaded: Loaded state.
- **Property-read** string \$townName: Name of town in which player resides.
- **Property-read** [OTS_House](#) \$house: House which player rents.
- **Version** 0.1.2
- **Version** 0.0.1
- **Since** 0.0.1

void function OTS_Player::ban([\$time = 0]) [line 1803]

Function Parameters:

- *int* **\$time** Time for time until expires (0 - forever).

Bans current player.

Bans current player.

- **Version** 0.0.5
- **Version** 0.0.1

- **Since** 0.0.1
- **Since** 0.0.5
- **Access** public

void function OTS_Player::delete() [line 1857]

Deletes player.

Deletes player.

- **Version** 0.0.5
- **Version** 0.0.1
- **Throws** E_OTS_NotLoaded If player is not loaded.
- **Since** 0.0.5
- **Since** 0.0.1
- **Access** public

void function OTS_Player::find(\$name) [line 128]

Function Parameters:

- *string* **\$name** Player's name.

Loads player by it's name.

Loads player by it's name.

- **Version** 0.0.5
- **Version** 0.0.1
- **Since** 0.0.1

- **Since** 0.0.2
- **Access** public

OTS_Account function OTS_Player::getAccount() [*line 230*]

Returns account of this player.

Returns account of this player.

- **Version** 0.1.0
- **Version** 0.0.1
- **Throws** E_OTS_NotLoaded If player is not loaded.
- **Since** 0.0.1
- **Access** public

int function OTS_Player::getBalance() [*line 1316*]

Bank balance.

Bank balance.

- **Version** 0.1.2
- **Version** 0.0.1
- **Throws** E_OTS_NotLoaded If player is not loaded.
- **Since** 0.1.2
- **Since** 0.0.1
- **Access** public

int function OTS_Player::getCap() [*line 885*]

Capacity.

Capacity.

- **Version** 0.0.3
- **Version** 0.0.1
- **Throws** E_OTS_NotLoaded If player is not loaded.
- **Since** 0.0.1
- **Access** public

string function OTS_Player::getConditions() [*line 1023*]

Conditions.

Conditions.

- **Version** 0.0.3
- **Version** 0.0.1
- **Throws** E_OTS_NotLoaded If player is not loaded.
- **Since** 0.0.1
- **Access** public

string function OTS_Player::getCustomField(\$field) [*line 1353*]

Function Parameters:

- *string* **\$field** Field name.

Reads custom field.

Reads custom field.

Reads field by it's name. Can read any field of given record that exists in database.

Note: You should use this method only for fields that are not provided in standard setters/getters (SVN fields). This method runs SQL query each time you call it so it highly overloads used resources.

- **Version** 0.0.5
- **Version** 0.0.1
- **Throws** E_OTS_NotLoaded If player is not loaded.
- **Since** 0.0.3
- **Since** 0.0.1
- **Access** public

OTS_Item|null function OTS_Player::getDepot(\$depot) [*line 1682*]

Function Parameters:

- *int* **\$depot** Depot ID to get items.

Returns items tree from given depot.

Returns items tree from given depot.

Note: OTS_Player class has no information about item types. It returns all items as OTS_Item, unless they have any contained items in database, so empty container will be instanced as OTS_Item object, not OTS_Container.

- **Version** 0.1.2
- **Version** 0.0.1
- **Throws** E_OTS_NotAContainer If item which is not of type container contains sub items.
- **Throws** E_OTS_NotLoaded If player is not loaded.
- **Since** 0.0.1
- **Since** 0.0.3
- **Access** public

int function OTS_Player::getDirection() [*line 615*]

Looking direction.

Looking direction.

- **Version** 0.0.3
- **Version** 0.0.1
- **Throws** E_OTS_NotLoaded If player is not loaded.
- **Since** 0.0.1
- **Access** public

int function OTS_Player::getExperience() [*line 372*]

Experience points.

Experience points.

- **Version** 0.0.3
- **Version** 0.0.1
- **Throws** E_OTS_NotLoaded If player is not loaded.
- **Since** 0.0.1
- **Access** public

OTS_Group function OTS_Player::getGroup() [*line 259*]

Returns group of this player.

Returns group of this player.

- **Version** 0.1.0
- **Version** 0.0.1
- **Throws** E_OTS_NotLoaded If player is not loaded.
- **Since** 0.0.1
- **Access** public

string function OTS_Player::getGuildNick() [*line 1110*]

Guild nick.

Guild nick.

- **Version** 0.0.3
- **Version** 0.0.1
- **Throws** E_OTS_NotLoaded If player is not loaded.
- **Since** 0.0.1
- **Access** public

int function OTS_Player::getHealth() [*line 453*]

Current HP.

Current HP.

- **Version** 0.0.3
- **Version** 0.0.1
- **Throws** E_OTS_NotLoaded If player is not loaded.
- **Since** 0.0.1
- **Access** public

int function OTS_Player::getHealthMax() [*line 480*]

Maximum HP.

Maximum HP.

- **Version** 0.0.3
- **Version** 0.0.1
- **Throws** E_OTS_NotLoaded If player is not loaded.
- **Since** 0.0.1
- **Access** public

OTS_House|null function OTS_Player::getHouse() [*line 1915*]

Returns house rented by this player.

Returns house rented by this player.

- **Version** 0.1.0
- **Version** 0.0.1
- **Throws** E_OTS_NotLoaded If player is not loaded.
- **Since** 0.1.0
- **Since** 0.0.1
- **Access** public

int function OTS_Player::getId() [*line 186*]

Player ID.

Player ID.

- **Version** 0.0.3

- **Version** 0.0.1
- **Throws** E_OTS_NotLoaded If player is not loaded.
- **Since** 0.0.1
- **Access** public

int function OTS_Player::getLastIP() [*line 939*]

Last login IP.

Last login IP.

- **Version** 0.0.3
- **Version** 0.0.1
- **Throws** E_OTS_NotLoaded If player is not loaded.
- **Since** 0.0.1
- **Access** public

int function OTS_Player::getLastLogin() [*line 912*]

Last login timestamp.

Last login timestamp.

- **Version** 0.0.3
- **Version** 0.0.1
- **Throws** E_OTS_NotLoaded If player is not loaded.
- **Since** 0.0.1
- **Access** public

int function OTS_Player::getLevel() [*line 399*]

Experience level.

Experience level.

- **Version** 0.0.3
- **Version** 0.0.1
- **Throws** E_OTS_NotLoaded If player is not loaded.
- **Since** 0.0.1
- **Access** public

int function OTS_Player::getLookAddons() [*line 777*]

Addons.

Addons.

- **Version** 0.0.3
- **Version** 0.0.1
- **Throws** E_OTS_NotLoaded If player is not loaded.
- **Since** 0.0.1
- **Access** public

int function OTS_Player::getLookBody() [*line 642*]

Body color.

Body color.

- **Version** 0.0.3
- **Version** 0.0.1

- **Throws** E_OTS_NotLoaded If player is not loaded.
- **Since** 0.0.1
- **Access** public

int function OTS_Player::getLookFeet() [*line 669*]

Boots color.

Boots color.

- **Version** 0.0.3
- **Version** 0.0.1
- **Throws** E_OTS_NotLoaded If player is not loaded.
- **Since** 0.0.1
- **Access** public

int function OTS_Player::getLookHead() [*line 696*]

Hair color.

Hair color.

- **Version** 0.0.3
- **Version** 0.0.1
- **Throws** E_OTS_NotLoaded If player is not loaded.
- **Since** 0.0.1
- **Access** public

int function OTS_Player::getLookLegs() [*line 723*]

Legs color.

Legs color.

- **Version** 0.0.3
- **Version** 0.0.1
- **Throws** E_OTS_NotLoaded If player is not loaded.
- **Since** 0.0.1
- **Access** public

int function OTS_Player::getLookType() [*line 750*]

Outfit.

Outfit.

- **Version** 0.0.3
- **Version** 0.0.1
- **Throws** E_OTS_NotLoaded If player is not loaded.
- **Since** 0.0.1
- **Access** public

int function OTS_Player::getLossExperience() [*line 1234*]

Percentage of experience lost after dead.

Percentage of experience lost after dead.

- **Version** 0.0.3
- **Version** 0.0.1
- **Throws** E_OTS_NotLoaded If player is not loaded.

- **Since** 0.0.1
- **Access** public

int function OTS_Player::getLossMana() [*line 1261*]

Percentage of used mana lost after dead.

Percentage of used mana lost after dead.

- **Version** 0.0.3
- **Version** 0.0.1
- **Throws** E_OTS_NotLoaded If player is not loaded.
- **Since** 0.0.1
- **Access** public

int function OTS_Player::getLossSkills() [*line 1288*]

Percentage of skills lost after dead.

Percentage of skills lost after dead.

- **Version** 0.0.3
- **Version** 0.0.1
- **Throws** E_OTS_NotLoaded If player is not loaded.
- **Since** 0.0.1
- **Access** public

int function OTS_Player::getMagLevel() [*line 426*]

Magic level.

Magic level.

- **Version** 0.0.3
- **Version** 0.0.1
- **Throws** E_OTS_NotLoaded If player is not loaded.
- **Since** 0.0.1
- **Access** public

int function OTS_Player::getMana() [*line 507*]

Current mana.

Current mana.

- **Version** 0.0.3
- **Version** 0.0.1
- **Throws** E_OTS_NotLoaded If player is not loaded.
- **Since** 0.0.1
- **Access** public

int function OTS_Player::getManaMax() [*line 534*]

Maximum mana.

Maximum mana.

- **Version** 0.0.3
- **Version** 0.0.1
- **Throws** E_OTS_NotLoaded If player is not loaded.
- **Since** 0.0.1
- **Access** public

int function OTS_Player::getManaSpent() [*line 561*]

Mana spent.

Mana spent.

- **Version** 0.0.3
- **Version** 0.0.1
- **Throws** E_OTS_NotLoaded If player is not loaded.
- **Since** 0.0.1
- **Access** public

string function OTS_Player::getName() [*line 203*]

Player name.

Player name.

- **Version** 0.0.3
- **Version** 0.0.1
- **Throws** E_OTS_NotLoaded If player is not loaded.
- **Since** 0.0.1
- **Access** public

int function OTS_Player::getPosX() [*line 804*]

X map coordinate.

X map coordinate.

- **Version** 0.0.3
- **Version** 0.0.1
- **Throws** E_OTS_NotLoaded If player is not loaded.
- **Since** 0.0.1
- **Access** public

int function OTS_Player::getPosY() [*line 831*]

Y map coordinate.

Y map coordinate.

- **Version** 0.0.3
- **Version** 0.0.1
- **Throws** E_OTS_NotLoaded If player is not loaded.
- **Since** 0.0.1
- **Access** public

int function OTS_Player::getPosZ() [*line 858*]

Z map coordinate.

Z map coordinate.

- **Version** 0.0.3
- **Version** 0.0.1
- **Throws** E_OTS_NotLoaded If player is not loaded.
- **Since** 0.0.1
- **Access** public

int function OTS_Player::getPremiumEnd() [*line 289*]

Player's Premium Account expiration timestamp.

Player's Premium Account expiration timestamp.

- **Version** 0.0.3
- **Version** 0.0.1
- **Throws** E_OTS_NotLoaded If player is not loaded.
- **Since** 0.0.3
- **Since** 0.0.1
- **Access** public

OTS_GuildRank|null function OTS_Player::getRank() [*line 1155*]

Assigned guild rank.

Assigned guild rank.

- **Version** 0.1.0
- **Version** 0.0.1
- **Throws** E_OTS_NotLoaded If player is not loaded.
- **Since** 0.0.1
- **Access** public

int function OTS_Player::getRankId() [*line 1138*]

Guild rank ID.

Guild rank ID.

- **Version** 0.0.3
- **Version** 0.0.1
- **Deprecated** 0.0.4 Use getRank().
- **Since** 0.0.1
- **Throws** E_OTS_NotLoaded If player is not loaded.
- **Access** public

int function OTS_Player::getRedSkullTime() [*line 1050*]

Red skulled time remained.

Red skulled time remained.

- **Version** 0.0.3
- **Version** 0.0.1
- **Throws** E_OTS_NotLoaded If player is not loaded.
- **Since** 0.0.1
- **Access** public

int function OTS_Player::getSave() [*line 995*]

Save counter.

Save counter.

- **Version** 0.0.7
- **Version** 0.0.1
- **Deprecated** 0.0.7 Save field is back as flag not a counter.
- **Throws** E_OTS_NotLoaded If player is not loaded.
- **Since** 0.0.1
- **Since** 0.0.6

- **Access** public

int function OTS_Player::getSex() [*line 318*]

Player gender.

Player gender.

- **Version** 0.0.3
- **Version** 0.0.1
- **Throws** E_OTS_NotLoaded If player is not loaded.
- **Since** 0.0.1
- **Access** public

int function OTS_Player::getSkill(\$skill) [*line 1408*]

Function Parameters:

- *int* **\$skill** Skill ID.

Returns player's skill.

Returns player's skill.

- **Version** 0.0.2
- **Version** 0.0.1
- **Throws** E_OTS_NotLoaded If player is not loaded.
- **Since** 0.0.2
- **Since** 0.0.1
- **Access** public

int function OTS_Player::getSkillTries(\$skill) [*line 1440*]

Function Parameters:

- *int* **\$skill** Skill ID.

Returns player's skill's tries for next level.

Returns player's skill's tries for next level.

- **Version** 0.0.2
- **Version** 0.0.1
- **Throws** E_OTS_NotLoaded If player is not loaded.
- **Since** 0.0.2
- **Since** 0.0.1
- **Access** public

OTS_Item|null function OTS_Player::getSlot(\$slot) [*line 1544*]

Function Parameters:

- *int* **\$slot** Slot to get items.

Returns items tree from given slot.

Returns items tree from given slot.

Note: OTS_Player class has no information about item types. It returns all items as OTS_Item, unless they have any contained items in database, so empty container will be instanced as OTS_Item object, not OTS_Container.

- **Version** 0.1.2

- **Version** 0.0.1
- **Throws** E_OTS_NotAContainer If item which is not of type container contains sub items.
- **Throws** E_OTS_NotLoaded If player is not loaded.
- **Since** 0.0.1
- **Since** 0.0.3
- **Access** public

int function OTS_Player::getSoul() [*line 588*]

Soul points.

Soul points.

- **Version** 0.0.3
- **Version** 0.0.1
- **Throws** E_OTS_NotLoaded If player is not loaded.
- **Since** 0.0.1
- **Access** public

int|null function OTS_Player::getStorage(\$key) [*line 1472*]

Function Parameters:

- *int* **\$key** Storage key.

Returns value of storage record.

Returns value of storage record.

- **Version** 0.1.2
- **Version** 0.0.1

- **Throws** E_OTS_NotLoaded If player is not loaded.
- **Since** 0.1.2
- **Since** 0.0.1
- **Access** public

int function OTS_Player::getTownId() [*line 1207*]

Residence town's ID.

Residence town's ID.

- **Version** 0.0.3
- **Version** 0.0.1
- **Throws** E_OTS_NotLoaded If player is not loaded.
- **Since** 0.0.1
- **Access** public

string/bool function OTS_Player::getTownName() [*line 1897*]

Player residence town name.

Player residence town name.

- **Version** 0.1.0
- **Version** 0.0.1
- **Throws** E_OTS_NotLoaded If player is not loaded.
- **Since** 0.1.0
- **Since** 0.0.1
- **Access** public

int function OTS_Player::getVocation() [*line 345*]

Player profession.

Player proffesion.

- **Version** 0.0.3
- **Version** 0.0.1
- **Throws** E_OTS_NotLoaded If player is not loaded.
- **Since** 0.0.1
- **Access** public

string|bool function OTS_Player::getVocationName() [*line 1879*]

Player profession name.

Player proffesion name.

- **Version** 0.1.0
- **Version** 0.0.1
- **Throws** E_OTS_NotLoaded If player is not loaded.
- **Since** 0.0.6
- **Since** 0.0.1
- **Access** public

bool function OTS_Player::hasRedSkull() [*line 1077*]

Checks if player has red skull.

Checks if player has red skull.

- **Version** 0.0.3
- **Version** 0.0.1
- **Throws** E_OTS_NotLoaded If player is not loaded.
- **Since** 0.0.1
- **Access** public

bool function OTS_Player::isBanned() [*line 1838*]

Checks if player is banned.

Checks if player is banned.

- **Version** 0.0.5
- **Version** 0.0.1
- **Since** 0.0.1
- **Since** 0.0.5
- **Access** public

bool function OTS_Player::isLoading() [*line 145*]

Checks if object is loaded.

Checks if object is loaded.

- **Version** 0.0.1
- **Since** 0.0.1
- **Access** public

bool function OTS_Player::isSaveSet() [*line 966*]

Checks if save flag is set.

Checks if save flag is set.

- **Version** 0.0.7
- **Version** 0.0.1
- **Throws** E_OTS_NotLoaded If player is not loaded.
- **Since** 0.0.1
- **Access** public

void function OTS_Player::load(\$id) [line 106]

Function Parameters:

- *int* **\$id** Player's ID.

Loads player with given id.

Loads player with given id.

- **Version** 0.1.2
- **Version** 0.0.1
- **Since** 0.0.1
- **Access** public

void function OTS_Player::save() [line 155]

Saves player in database.

Saves player in database.

- **Version** 0.1.2

- **Version** 0.0.1
- **Since** 0.0.1
- **Access** public

void function OTS_Player::setAccount(\$account) [line 247]

Function Parameters:

- [OTS Account](#) **\$account** Owning account.

Assigns character to account.

Assigns character to account.

- **Version** 0.0.1
- **Since** 0.0.1
- **Access** public

void function OTS_Player::setBalance(\$balance) [line 1333]

Function Parameters:

- *int* **\$balance** Amount of money to be set in bank.

Sets bank balance value.

Sets bank balance value.

- **Version** 0.1.2
- **Version** 0.0.1
- **Since** 0.0.1

- **Since** 0.1.2
- **Access** public

void function OTS_Player::setCap(\$cap) [line 900]

Function Parameters:

- *int* **\$cap** Capacity.

Sets capacity.

Sets capacity.

- **Version** 0.0.1
- **Since** 0.0.1
- **Access** public

void function OTS_Player::setConditions(\$conditions) [line 1038]

Function Parameters:

- *string* **\$conditions** Condition binary string.

Sets conditions.

Sets conditions.

- **Version** 0.0.1
- **Since** 0.0.1
- **Access** public

`void function OTS_Player::setCustomField($field, $value) [line 1383]`

Function Parameters:

- *string* **\$field** Field name.
- *mixed* **\$value** Field value.

Writes custom field.

Writes custom field.

Write field by it's name. Can write any field of given record that exists in database.

Note: You should use this method only for fields that are not provided in standard setters/getters (SVN fields). This method runs SQL query each time you call it so it highly overloads used resources.

Note: Make sure that you pass \$value argument of correct type. This method determinates whether to quote field value. It is safe - it makes you sure that no unproper queries that could lead to SQL injection will be executed, but it can make your code working wrong way. For example: `$object->setCustomField('foo', '1');` will quote 1 as as string ('1') instead of passing it as a integer.

- **Version** 0.0.5
- **Version** 0.0.1
- **Throws** E_OTS_NotLoaded If player is not loaded.
- **Since** 0.0.3
- **Since** 0.0.1
- **Access** public

`void function OTS_Player::setDepot($depot, [$item = null], [$pid = 0], [$depot_id = 0]) [line 1739]`

Function Parameters:

- *int* **\$depot** Depot ID to save items.
- [*OTS_Item*](#) **\$item** Item (can be a container with content) for given depot. Leave this parameter

blank to clear depot.

- *int* **\$pid** Deprecated, not used anymore.
- *int* **\$depot_id** Internal, for further use.

Sets depot content.

Sets depot content.

- **Version** 0.1.2
- **Version** 0.0.1
- **Throws** E_OTS_NotLoaded If player is not loaded.
- **Since** 0.0.3
- **Since** 0.0.1
- **Access** public

void function OTS_Player::setDirection(\$direction) [*line 630*]

Function Parameters:

- *int* **\$direction** Looking direction.

Sets looking direction.

Sets looking direction.

- **Version** 0.0.1
- **Since** 0.0.1
- **Access** public

void function OTS_Player::setExperience(\$experience) [line 387]

Function Parameters:

- *int* **\$experience** Experience points.

Sets experience points.

Sets experience points.

- **Version** 0.0.1
- **Since** 0.0.1
- **Access** public

void function OTS_Player::setGroup(\$group) [line 276]

Function Parameters:

- [*OTS_Group*](#) **\$group** Group to be a member.

Assigns character to group.

Assigns character to group.

- **Version** 0.0.1
- **Since** 0.0.1
- **Access** public

void function OTS_Player::setGuildNick(\$guildnick) [line 1125]

Function Parameters:

- *string* **\$guildnick** Name.

Sets guild nick.

Sets guild nick.

- **Version** 0.0.1
- **Since** 0.0.1
- **Access** public

void function OTS_Player::setHealth(\$health) [*line 468*]

Function Parameters:

- *int* **\$health** Current HP.

Sets current HP.

Sets current HP.

- **Version** 0.0.1
- **Since** 0.0.1
- **Access** public

void function OTS_Player::setHealthMax(\$healthmax) [*line 495*]

Function Parameters:

- *int* **\$healthmax** Maximum HP.

Sets maximum HP.

Sets maximum HP.

- **Version** 0.0.1
- **Since** 0.0.1
- **Access** public

void function OTS_Player::setLastIP(\$lastip) [line 954]

Function Parameters:

- *int* **\$lastip** Last login IP.

Sets last login IP.

Sets last login IP.

- **Version** 0.0.1
- **Since** 0.0.1
- **Access** public

void function OTS_Player::setLastLogin(\$lastlogin) [line 927]

Function Parameters:

- *int* **\$lastlogin** Last login timestamp.

Sets last login timestamp.

Sets last login timestamp.

- **Version** 0.0.1
- **Since** 0.0.1
- **Access** public

void function OTS_Player::setLevel(\$level) [line 414]

Function Parameters:

- *int* **\$level** Experience level.

Sets experience level.

Sets experience level.

- **Version** 0.0.1
- **Since** 0.0.1
- **Access** public

void function OTS_Player::setLookAddons(\$lookaddons) [line 792]

Function Parameters:

- *int* **\$lookaddons** Addons.

Sets addons.

Sets addons.

- **Version** 0.0.1
- **Since** 0.0.1

- **Access** public

void function OTS_Player::setLookBody(\$lookbody) [line 657]

Function Parameters:

- *int* **\$lookbody** Body color.

Sets body color.

Sets body color.

- **Version** 0.0.1
- **Since** 0.0.1
- **Access** public

void function OTS_Player::setLookFeet(\$lookfeet) [line 684]

Function Parameters:

- *int* **\$lookfeet** Boots color.

Sets boots color.

Sets boots color.

- **Version** 0.0.1
- **Since** 0.0.1
- **Access** public

void function OTS_Player::setLookHead(\$lookhead) [line 711]

Function Parameters:

- *int* **\$lookhead** Hair color.

Sets hair color.

Sets hair color.

- **Version** 0.0.1
- **Since** 0.0.1
- **Access** public

void function OTS_Player::setLookLegs(\$looklegs) [*line 738*]

Function Parameters:

- *int* **\$looklegs** Legs color.

Sets legs color.

Sets legs color.

- **Version** 0.0.1
- **Since** 0.0.1
- **Access** public

void function OTS_Player::setLookType(\$looktype) [*line 765*]

Function Parameters:

- *int* **\$looktype** Outfit.

Sets outfit.

Sets outfit.

- **Version** 0.0.1
- **Since** 0.0.1
- **Access** public

void function OTS_Player::setLossExperience(\$loss_experience) [line 1249]

Function Parameters:

- *int* **\$loss_experience** Percentage of experience lost after dead.

Sets percentage of experience lost after dead.

Sets percentage of experience lost after dead.

- **Version** 0.0.1
- **Since** 0.0.1
- **Access** public

void function OTS_Player::setLossMana(\$loss_mana) [line 1276]

Function Parameters:

- *int* **\$loss_mana** Percentage of used mana lost after dead.

Sets percentage of used mana lost after dead.

Sets percentage of used mana lost after dead.

- **Version** 0.0.1
- **Since** 0.0.1
- **Access** public

void function OTS_Player::setLossSkills(\$loss_skills) [line 1303]

Function Parameters:

- *int* **\$loss_skills** Percentage of skills lost after dead.

Sets percentage of skills lost after dead.

Sets percentage of skills lost after dead.

- **Version** 0.0.1
- **Since** 0.0.1
- **Access** public

void function OTS_Player::setMagLevel(\$maglevel) [line 441]

Function Parameters:

- *int* **\$maglevel** Magic level.

Sets magic level.

Sets magic level.

- **Version** 0.0.1
- **Since** 0.0.1
- **Access** public

void function OTS_Player::setMana(\$mana) [line 522]

Function Parameters:

- *int* **\$mana** Current mana.

Sets current mana.

Sets current mana.

- **Version** 0.0.1
- **Since** 0.0.1
- **Access** public

void function OTS_Player::setManaMax(\$manamax) [line 549]

Function Parameters:

- *int* **\$manamax** Maximum mana.

Sets maximum mana.

Sets maximum mana.

- **Version** 0.0.1
- **Since** 0.0.1
- **Access** public

void function OTS_Player::setManaSpent(\$manaspent) [line 576]

Function Parameters:

- *int* **\$manaspent** Mana spent.

Sets mana spent.

Sets mana spent.

- **Version** 0.0.1
- **Since** 0.0.1
- **Access** public

void function OTS_Player::setName(\$name) [line 218]

Function Parameters:

- *string* **\$name** Name.

Sets players's name.

Sets players's name.

- **Version** 0.0.1
- **Since** 0.0.1
- **Access** public

void function OTS_Player::setPosX(\$posx) [line 819]

Function Parameters:

- *int* **\$posx** X map coordinate.

Sets X map coordinate.

Sets X map coordinate.

- **Version** 0.0.1
- **Since** 0.0.1
- **Access** public

void function OTS_Player::setPosY(\$posy) [*line 846*]

Function Parameters:

- *int* **\$posy** Y map coordinate.

Sets Y map coordinate.

Sets Y map coordinate.

- **Version** 0.0.1
- **Since** 0.0.1
- **Access** public

void function OTS_Player::setPosZ(\$posz) [*line 873*]

Function Parameters:

- *int* **\$posz** Z map coordinate.

Sets Z map coordinate.
Sets Z map coordinate.

- **Version** 0.0.1
- **Since** 0.0.1
- **Access** public

void function OTS_Player::setPremiumEnd(\$premend) [line 306]
Function Parameters:

- *int* **\$premend** PACC expiration timestamp.

Sets player's Premium Account expiration timestamp.
Sets player's Premium Account expiration timestamp.

- **Version** 0.0.3
- **Version** 0.0.1
- **Since** 0.0.1
- **Since** 0.0.3
- **Access** public

void function OTS_Player::setRank([\$guildRank = null]) [line 1188]
Function Parameters:

- [*OTS_GuildRank*](#)|*null* **\$guildRank** Guild rank (null to clear assign).

Assigns guild rank.

Assigns guild rank.

- **Version** 0.0.1
- **Since** 0.0.1
- **Access** public

void function OTS_Player::setRankId(\$rank_id) [line 1178]

Function Parameters:

- *int* **\$rank_id** Guild rank ID.

Sets guild rank ID.

Sets guild rank ID.

- **Version** 0.0.1
- **Deprecated** 0.0.4 Use setRank().
- **Since** 0.0.1
- **Access** public

void function OTS_Player::setRedSkull() [line 1098]

Sets red skull flag.

Sets red skull flag.

- **Version** 0.0.1
- **Since** 0.0.1

- **Access** public

void function OTS_Player::setRedSkullTime(\$redskulltime) [line 1065]

Function Parameters:

- *int* **\$redskulltime** Red skulled time remained.

Sets red skulled time remained.

Sets red skulled time remained.

- **Version** 0.0.1
- **Since** 0.0.1
- **Access** public

void function OTS_Player::setSave([\$save = 1]) [line 1011]

Function Parameters:

- *int* **\$save** Deprecated, unused, optional.

Sets save flag.

Sets save flag.

- **Version** 0.0.7
- **Version** 0.0.1
- **Since** 0.0.1
- **Access** public

void function OTS_Player::setSex(\$sex) [line 333]

Function Parameters:

- *int* **\$sex** Player gender.

Sets player gender.

Sets player gender.

- **Version** 0.0.1
- **Since** 0.0.1
- **Access** public

void function OTS_Player::setSkill(\$skill, \$value) [line 1426]

Function Parameters:

- *int* **\$skill** Skill ID.
- *int* **\$value** Skill value.

Sets skill value.

Sets skill value.

- **Version** 0.0.2
- **Version** 0.0.1
- **Since** 0.0.1
- **Since** 0.0.2
- **Access** public

void function OTS_Player::setSkillTries(\$skill, \$tries) [line 1458]

Function Parameters:

- *int* **\$skill** Skill ID.
- *int* **\$tries** Skill tries.

Sets skill's tries for next level.

Sets skill's tries for next level.

- **Version** 0.0.2
- **Version** 0.0.1
- **Since** 0.0.1
- **Since** 0.0.2
- **Access** public

void function OTS_Player::setSlot(\$slot, [\$item = null], [\$pid = 0]) [line 1600]

Function Parameters:

- *int* **\$slot** Slot to save items.
- [*OTS_Item*](#) **\$item** Item (can be a container with content) for given slot. Leave this parameter blank to clear slot.
- *int* **\$pid** Deprecated, not used anymore.

Sets slot content.

Sets slot content.

- **Version** 0.1.2

- **Version** 0.0.1
- **Throws** E_OTS_NotLoaded If player is not loaded.
- **Since** 0.0.3
- **Since** 0.0.1
- **Access** public

void function OTS_Player::setSoul(\$soul) [line 603]

Function Parameters:

- *int* **\$soul** Soul points.

Sets soul points.

Sets soul points.

- **Version** 0.0.1
- **Since** 0.0.1
- **Access** public

void function OTS_Player::setStorage(\$key, \$value) [line 1492]

Function Parameters:

- *int* **\$key** Storage key.
- *int* **\$value** Stored value.

Sets value of storage record.

Sets value of storage record.

- **Version** 0.1.2
- **Version** 0.0.1
- **Throws** E_OTS_NotLoaded If player is not loaded.
- **Since** 0.1.2
- **Since** 0.0.1
- **Access** public

void function OTS_Player::setTownId(\$town_id) [line 1222]

Function Parameters:

- *int* **\$town_id** Residence town's ID.

Sets residence town's ID.

Sets residence town's ID.

- **Version** 0.0.1
- **Since** 0.0.1
- **Access** public

void function OTS_Player::setVocation(\$vocation) [line 360]

Function Parameters:

- *int* **\$vocation** Player proffesion.

Sets player proffesion.

Sets player proffesion.

- **Version** 0.0.1
- **Since** 0.0.1
- **Access** public

void function OTS_Player::unban() [line 1820]

Deletes ban from current player.

Deletes ban from current player.

- **Version** 0.0.5
- **Version** 0.0.1
- **Since** 0.0.1
- **Since** 0.0.5
- **Access** public

void function OTS_Player::unsetRedSkull() [line 1090]

Unsets red skull flag.

Unsets red skull flag.

- **Version** 0.0.1
- **Since** 0.0.1
- **Access** public

void function OTS_Player::unsetSave() [line 981]

Unsets save flag.

Unsets save flag.

- **Version** 0.0.7
- **Version** 0.0.1
- **Since** 0.0.1
- **Access** public

mixed function OTS_Player::__get(\$name) [*line 1944*]

Function Parameters:

- *string* **\$name** Property name.

Magic PHP5 method.

Magic PHP5 method.

- **Version** 0.1.2
- **Version** 0.0.1
- **Throws** OutOfBoundsException For non-supported properties.
- **Since** 0.1.0
- **Since** 0.0.1
- **Access** public

void function OTS_Player::__set(\$name, \$value) [*line 2094*]

Function Parameters:

- *string* **\$name** Property name.
- *mixed* **\$value** Property value.

Magic PHP5 method.

Magic PHP5 method.

- **Version** 0.1.2
- **Version** 0.0.1
- **Throws** OutOfBoundsException For non-supported properties.
- **Since** 0.1.0
- **Since** 0.0.1
- **Access** public

array function OTS_Player::__sleep() [*line 95*]

Magic PHP5 method.

Magic PHP5 method.

Allows object serialisation.

- **Version** 0.0.4
- **Version** 0.0.1
- **Since** 0.0.1
- **Since** 0.0.4
- **Access** public

string function OTS_Player::__toString() [*line 2293*]

Returns string representation of object.

Returns string representation of object.

If any display driver is currently loaded then it uses it's method. Else it returns character name.

- **Version** 0.1.0
- **Version** 0.0.1
- **Since** 0.0.1
- **Since** 0.1.0
- **Access** public

Class OTS_Players_List

[line 21]

List of players.
List of players.

- **Package** POT
- **Version** 0.1.0
- **Since** 0.0.1

void function OTS_Players_List::deletePlayer(\$player) *[line 30]*

Function Parameters:

- [*OTS_Player*](#) **\$player** Player to be deleted.

Deletes player.
Deletes player.

- **Version** 0.0.5
- **Deprecated** 0.0.5 Use OTS_Player->delete().
- **Since** 0.0.1
- **Access** public

void function OTS_Players_List::init() [*line 43*]

Sets list parameters.

Sets list parameters.

This method is called at object creation.

- **Version** 0.0.5
- **Since** 0.0.1
- **Since** 0.0.5
- **Access** public

string function OTS_Players_List::__toString() [*line 58*]

Returns string representation of object.

Returns string representation of object.

If any display driver is currently loaded then it uses it's method.

- **Version** 0.1.0
- **Since** 0.0.1
- **Since** 0.1.0
- **Access** public

Class OTS_Row_DAO

[line 20]

Base class for all single-row classes.

Base class for all single-row classes. It implements auto-loading constructors.

- **Package** POT
- **Version** 0.1.1
- **Abstract Element**
- **Since** 0.1.1

Constructor *void* function OTS_Row_DAO::__construct([\$id = null]) [line 27]

Function Parameters:

- *int|string|null* **\$id** Row ID (or name identifier dependend on child class).

Handles automatic loading for record.

Handles automatic loading for record.

- **Version** 0.1.1
- **Since** 0.1.1
- **Access** public

void function OTS_Row_DAO::find(\$name) [line 58]

Function Parameters:

- *string* **\$name** String identifier.

Loads row by it's name.

Loads row by it's name.

- **Version** 0.1.1
- **Abstract Element**
- **Since** 0.1.1
- **Access** public

void function OTS_Row_DAO::load(\$id) [*line 51*]

Function Parameters:

- *int* **\$id** Integer identifier.

Loads row by it's ID.

Loads row by it's ID.

- **Version** 0.1.1
- **Abstract Element**
- **Since** 0.1.1
- **Access** public

Class OTS_RSA

[*line 24*]

RSA encryption/decryption mechanism.

RSA encryption/decryption mechanism.

This code bases in large part on Alexander Valyalkin's Crypt_RSA's source code.

- **Package** POT
- **Version** 0.1.2
- **Since** 0.1.2

OTS_RSA::D

```
=  
'467303302235841186221601800150368321487329868085193446752105552629402587398057668602246  
106469196058602063280243267033616301098884178392419595075722472848070352355696191737922  
927869078457919049551036016528225191219083671878855092700253886417008217353452220879405  
78381210879116823013776808975766851829020659073' [line 37]
```

OTServ key part.

OTServ key part.

- **Version** 0.1.2
- **Since** 0.1.2

OTS_RSA::P

```
=  
'142996239624163995200701773828988955507954033454661532174705160829347375827760388829672  
13386204600674145392845853859217990626450972452084065728686565928113' [line 29]
```

OTServ key part.

OTServ key part.

- **Version** 0.1.2

- **Since** 0.1.2

OTS_RSA::Q

=

'7630979195970404721891201847792002125535401292779123937207447574596692788513647179235335529307251350570728407373705564708871762033017096809910315212884101' [*line 33*]

OTServ key part.

OTServ key part.

- **Version** 0.1.2
- **Since** 0.1.2

Constructor *void* function OTS_RSA::__construct([\$p = self::P], [\$q = self::Q], [\$d = self::D]) [*line 79*]

Function Parameters:

- *string* **\$p** Key part.
- *string* **\$q** Key part.
- *string* **\$d** Key part.

Initializes new encryption session.

Initializes new encryption session.

If you won't pass any parameters default OTServ keys will be generated. It is recommended action for compatibility with original Tibia servers and clients as well as default Open Tibia implementation.

Note: You must be sure your p , q and d values are proper for RSA keys generation as class won't change it for you.

- **Version** 0.1.2
- **Throws** LogicException When BCMath extension is not loaded.

- **Since** 0.1.2
- **Access** public

string function OTS_RSA::decrypt(\$message) [*line 137*]

Function Parameters:

- *string* **\$message** RSA-encrypted message.

Decrypts RSA-encrypted message.

Decrypts RSA-encrypted message.

As OTServ clients use RSA encryption only for sending requests we don't need decryption here. If it will be needed, then this method will be implemented. At the moment it will throw exception.

- **Version** 0.1.2
- **Throws** LogicException Always as this method is not implemented.
- **Since** 0.1.2
- **Access** public

string function OTS_RSA::encrypt(\$message) [*line 119*]

Function Parameters:

- *string* **\$message** Message to be encrypted.

Ecnrypts message with RSA algorithm.

Ecnrypts message with RSA algorithm.

- **Version** 0.1.2

- **Since** 0.1.2
- **Access** public

Class OTS_Spell

[line 41]

Wrapper for spell info.
Wrapper for spell info.

- **Package** POT
- **Property-read** int \$type: Spell type.
- **Property-read** string \$name: Spell name.
- **Property-read** int \$id: Spell ID.
- **Property-read** string \$words: Spell formula.
- **Property-read** bool \$agressive: Does spell marks action as an attack.
- **Property-read** int \$charges: Rune charges count.
- **Property-read** int \$level: Required level.
- **Property-read** int \$magicLevel: Required magic level.
- **Property-read** int \$mana: Mana usage.
- **Property-read** int \$soul: Soul points usage.
- **Property-read** bool \$hasParams: Does spell has any arguments.
- **Property-read** bool \$enabled: Is spell enabled.
- **Property-read** bool \$farUseAllowed: Can the spell be used from distance.
- **Property-read** bool \$premium: Does spell requires PACC.
- **Property-read** bool \$learnNeeded: Does the spell needs to be learned.
- **Property-read** [OTS_ItemType](#)|null \$conjure: Conjure item type.

- **Property-read** [OTS_ItemType](#)|null \$reagent: Item required to cast this spell.
- **Property-read** int \$conjuresCount: Amount of items created with conjure cast.
- **Property-read** array \$vocations: List of vocations allowed to use.
- **Version** 0.1.0
- **Version** 0.0.7
- **Since** 0.0.7

Constructor *void* function OTS_Spell::__construct(\$type, \$spell) [*line 63*]
Function Parameters:

- *int* **\$type** Spell type.
- *DOMElement* **\$spell** Spell info.

Sets spell info.
 Sets spell info.

- **Version** 0.0.7
- **Since** 0.0.7
- **Access** public

OTS_Item function OTS_Spell::createConjure() [*line 313*]
Creates conjure item.
 Creates conjure item.

- **Version** 0.1.0
- **Version** 0.0.7

- **Since** 0.0.7
- **Since** 0.1.0
- **Access** public

int function OTS_Spell::getCharges() [*line 138*]

Number of rune charges.

Number of rune charges.

- **Version** 0.0.7
- **Since** 0.0.7
- **Access** public

OTS_ItemType|null function OTS_Spell::getConjure() [*line 251*]

Returns item type of conjured item.

Returns item type of conjured item.

- **Version** 0.1.0
- **Version** 0.0.7
- **Since** 0.0.7
- **Since** 0.1.0
- **Access** public

int function OTS_Spell::getConjureCount() [*line 284*]

Returns amount of items conjured by this spell.

Returns amount of items conjured by this spell.

- **Version** 0.0.7
- **Since** 0.0.7
- **Access** public

int function OTS_Spell::getConjureId() [*line 239*]

Returns ID of item conjured by this spell.

Returns ID of item conjured by this spell.

- **Version** 0.0.7
- **Deprecated** 0.1.0 Use getConjure()->getId().
- **Since** 0.0.7
- **Access** public

int function OTS_Spell::getID() [*line 94*]

Returns rune item id.

Returns rune item id.

- **Version** 0.0.7
- **Since** 0.0.7
- **Access** public

int function OTS_Spell::getLevel() [*line 148*]

Level required for use.

Level required for use.

- **Version** 0.0.7
- **Since** 0.0.7
- **Access** public

int function OTS_Spell::getMagicLevel() [*line 158*]

Magic level required to cast.

Magic level required to cast.

- **Version** 0.0.7
- **Since** 0.0.7
- **Access** public

int function OTS_Spell::getMana() [*line 168*]

Mana cost.

Mana cost.

- **Version** 0.0.7
- **Since** 0.0.7
- **Access** public

string function OTS_Spell::getName() [*line 84*]

Returns spell name.

Returns spell name.

- **Version** 0.0.7
- **Since** 0.0.7
- **Access** public

OTS_ItemType|null function OTS_Spell::getReagent() [*line 274*]

Returns item type of reagent item.

Returns item type of reagent item.

- **Version** 0.1.0
- **Version** 0.0.7
- **Since** 0.0.7
- **Since** 0.1.0
- **Access** public

int function OTS_Spell::getReagentId() [*line 262*]

Returns ID of item that is used by spell.

Returns ID of item that is used by spell.

- **Version** 0.0.7
- **Deprecated** 0.1.0 Use getReagent()->getId().
- **Since** 0.0.7
- **Access** public

int function OTS_Spell::getSoul() [*line 178*]

Soul points cost.

Soul points cost.

- **Version** 0.0.7
- **Since** 0.0.7
- **Access** public

int function OTS_Spell::getType() [*line 74*]

Returns spell type.

Returns spell type.

- **Version** 0.0.7
- **Since** 0.0.7
- **Access** public

array function OTS_Spell::getVocations() [*line 294*]

Returns list of vocations that are allowed to learn this spell.

Returns list of vocations that are allowed to learn this spell.

- **Version** 0.0.7
- **Since** 0.0.7
- **Access** public

string function OTS_Spell::getWords() [*line 104*]

Returns spell formula.

Returns spell formula.

- **Version** 0.0.7
- **Since** 0.0.7
- **Access** public

bool function OTS_Spell::hasParams() [line 188]

Checks if spell has parameter.

Checks if spell has parameter.

- **Version** 0.0.7
- **Since** 0.0.7
- **Access** public

bool function OTS_Spell::isAggressive() [line 128]

This method is the same as [OTS_Spell::isAggressive\(\)](#)

This method is the same as [OTS_Spell::isAggressive\(\)](#). It was created first by typo mistake. Left for backward compatibility.

- **Version** 0.1.0
- **Version** 0.0.7
- **Deprecated** 0.1.0 Use isAggressive().
- **Since** 0.0.7
- **Access** public

bool function OTS_Spell::isAggressive() [line 116]

Checks if spell is threatened as unfriendly by other creatures.

Checks if spell is threatened as unfriendly by other creatures.

- **Version** 0.1.0
- **Version** 0.0.7
- **Since** 0.0.7
- **Since** 0.1.0
- **Access** public

bool function OTS_Spell::isEnabled() [*line 198*]

Checks if spell is enabled.

Checks if spell is enabled.

- **Version** 0.0.7
- **Since** 0.0.7
- **Access** public

bool function OTS_Spell::isFarUseAllowed() [*line 208*]

Checks if distance use allowed.

Checks if distance use allowed.

- **Version** 0.0.7
- **Since** 0.0.7
- **Access** public

bool function OTS_Spell::isLearnNeeded() [*line 228*]

Checks if spell needs to be learned.

Checks if spell needs to be learned.

- **Version** 0.0.7
- **Since** 0.0.7
- **Access** public

bool function OTS_Spell::isPremium() [*line 218*]

Checks if spell requires PACC.

Checks if spell requires PACC.

- **Version** 0.0.7
- **Since** 0.0.7
- **Access** public

mixed function OTS_Spell::__get(\$name) [*line 329*]

Function Parameters:

- *string* **\$name** Property name.

Magic PHP5 method.

Magic PHP5 method.

- **Version** 0.1.0
- **Version** 0.0.7
- **Throws** OutOfBoundsException For non-supported properties.

- **Since** 0.1.0
- **Since** 0.0.7
- **Access** public

string function OTS_Spell::__toString() [*line 402*]

Returns string representation of XML.

Returns string representation of XML.

- **Version** 0.1.0
- **Version** 0.0.7
- **Since** 0.0.7
- **Since** 0.1.0
- **Access** public

Class OTS_SpellsList

[*line 23*]

Wrapper for spells list.

Wrapper for spells list.

- **Package** POT
- **Property-read** array \$runesList: List of rune spells.
- **Property-read** array \$instantsList: List of instant spells.
- **Property-read** array \$conjuresList: List of conjure spells.
- **Version** 0.1.0

- **Since** 0.1.0

OTS_SpellsList::SPELL_CONJURE

= 2 [*line 36*]

Conjure spell.
Conjure spell.

- **Version** 0.1.0
- **Since** 0.1.0

OTS_SpellsList::SPELL_INSTANT

= 1 [*line 32*]

Instant spell.
Instant spell.

- **Version** 0.1.0
- **Since** 0.1.0

OTS_SpellsList::SPELL_RUNE

= 0 [*line 28*]

Rune spell.
Rune spell.

- **Version** 0.1.0
- **Since** 0.1.0

Constructor *void* function OTS_SpellsList::__construct(\$file) [*line 84*]

Function Parameters:

- *string* **\$file** Spells file name.

Loads spells list.

Loads spells list.

- **Version** 0.1.0
- **Since** 0.1.0
- **Access** public

OTS_Spell|null function OTS_SpellsList::getConjure(\$name) [*line 181*]

Function Parameters:

- *string* **\$name** Spell name.

Returns given conjure spell.

Returns given conjure spell.

- **Version** 0.1.0
- **Since** 0.1.0
- **Access** public

array function OTS_SpellsList::getConjuresList() [*line 170*]

Returns list of conjure spells.

Returns list of conjure spells.

- **Version** 0.1.0
- **Since** 0.1.0
- **Access** public

OTS_Spell|null function OTS_SpellsList::getInstant(\$name) [*line 153*]

Function Parameters:

- *string* **\$name** Spell name.

Returns given instant spell.

Returns given instant spell.

- **Version** 0.1.0
- **Since** 0.1.0
- **Access** public

array function OTS_SpellsList::getInstantsList() [*line 142*]

Returns list of instants.

Returns list of instants.

- **Version** 0.1.0

- **Since** 0.1.0
- **Access** public

OTS_Spell|null function OTS_SpellsList::getRune(\$name) [*line 125*]

Function Parameters:

- *string* **\$name** Rune name.

Returns given rune spell.

Returns given rune spell.

- **Version** 0.1.0
- **Since** 0.1.0
- **Access** public

array function OTS_SpellsList::getRunesList() [*line 114*]

Returns list of runes.

Returns list of runes.

- **Version** 0.1.0
- **Since** 0.1.0
- **Access** public

mixed function OTS_SpellsList::__get(\$name) [*line 200*]

Function Parameters:

- *string* **\$name** Property name.

Magic PHP5 method.

Magic PHP5 method.

- **Version** 0.1.0
- **Throws** OutOfBoundsException For non-supported properties.
- **Since** 0.1.0
- **Access** public

void function OTS_SpellsList::__set_state(\$properties) [line 66]

Function Parameters:

- *array* **\$properties** List of object properties.

Magic PHP5 method.

Magic PHP5 method.

Allows object importing from [var_export\(\)](#).

- **Version** 0.1.0
- **Since** 0.1.0
- **Access** public

Class OTS_SQLField

[line 24]

SQL identifier representation.

SQL identifier representation.

- **Package** POT
- **Property-read** string \$name: Field name.
- **Property-read** string \$table: Table name.
- **Version** 0.1.0
- **Version** 0.0.5
- **Since** 0.0.5

Constructor *void* function OTS_SQLField::__construct(\$name, [\$table = "]) [*line 45*]

Function Parameters:

- *string* **\$name** Field name.
- *string* **\$table** Table name.

Creates new field representation.

Creates new field representation.

- **Version** 0.0.5
- **Since** 0.0.5
- **Access** public

string function OTS_SQLField::getName() [*line 56*]

Returns field name.

Returns field name.

- **Version** 0.0.5
- **Since** 0.0.5
- **Access** public

string function OTS_SQLField::getTable() [*line 66*]

Returns table name.

Returns table name.

- **Version** 0.0.5
- **Since** 0.0.5
- **Access** public

mixed function OTS_SQLField::__get(\$name) [*line 80*]

Function Parameters:

- *string* **\$name** Property name.

Magic PHP5 method.

Magic PHP5 method.

- **Version** 0.1.0
- **Version** 0.0.5
- **Throws** OutOfBoundsException For non-supported properties.
- **Since** 0.1.0
- **Since** 0.0.5

- **Access** public

string function OTS_SQLField::__toString() [*line 103*]

Returns string representation of WHERE clause.

Returns string representation of WHERE clause.

Returned string can be easily inserted into SQL query.

- **Version** 0.1.0
- **Version** 0.0.5
- **Since** 0.0.5
- **Since** 0.1.0
- **Access** public

Class OTS_SQLFilter

[*line 23*]

SQL WHERE clause object.

SQL WHERE clause object.

- **Package** POT
- **Property-read** array \$tables: List of tables used by this statement.
- **Version** 0.1.0
- **Version** 0.0.5
- **Since** 0.0.5

OTS_SQLFilter::CRITERIUM_AND

= 1 [*line 61*]

AND sibling.

AND sibling.

- **Version** 0.0.5
- **Since** 0.0.5

OTS_SQLFilter::CRITERIUM_OR

= 2 [*line 65*]

OR sibling.

OR sibling.

- **Version** 0.0.5
- **Since** 0.0.5

OTS_SQLFilter::OPERATOR_EQUAL

= 1 [*line 28*]

Equal operator.

Equal operator.

- **Version** 0.0.5
- **Since** 0.0.5

OTS_SQLFilter::OPERATOR_GREATER

= 3 [*line 36*]

Greater-then operator.

Greater-then operator.

- **Version** 0.0.5
- **Since** 0.0.5

OTS_SQLFilter::OPERATOR_LIKE

= 7 [*line 52*]

LIKE operator.

LIKE operator.

- **Version** 0.0.5
- **Since** 0.0.5

OTS_SQLFilter::OPERATOR_LOWER

= 2 [*line 32*]

Lower-then operator.

Lower-then operator.

- **Version** 0.0.5
- **Since** 0.0.5

OTS_SQLFilter::OPERATOR_NEQUAL

= 4 [*line 40*]

Not-equal operator.

Not-equal operator.

- **Version** 0.0.5
- **Since** 0.0.5

OTS_SQLFilter::OPERATOR_NGREATER

= 6 [*line 48*]

Not-greater-then operator.

Not-greater-then operator.

- **Version** 0.0.5
- **Since** 0.0.5

OTS_SQLFilter::OPERATOR_NLIKE

= 8 [*line 56*]

Not-LIKE operator.

Not-LIKE operator.

- **Version** 0.0.5

- **Since** 0.0.5

OTS_SQLFilter::OPERATOR_NLOWER

= 5 [line 44]

Not-lower-then operator.

Not-lower-then operator.

- **Version** 0.0.5
- **Since** 0.0.5

void function OTS_SQLFilter::addFilter(\$left, [\$right = null], [\$operator = self::OPERATOR_EQUAL], [\$criterium = self::CRITERIUM_AND]) [line 226]

Function Parameters:

- *mixed* **\$left** Left side ([OTS_SQLField class](#) object, or literal value).
- *mixed* **\$right** Right side ([OTS_SQLField class](#) object, or literal value).
- *int* **\$operator** Operator used for comparsion (equal check by default).
- *int* **\$criterium** Criterium merging method (AND by default).

General-purpose filter.

General-purpose filter.

Appends new filter in universal way.

To append subset of another filters us addFilter(\$OTS_SQLFilterObject).

- **Version** 0.0.5
- **Since** 0.0.5

- **Access** public

void function OTS_SQLFilter::compareField(\$field, \$value, [\$operator = self::OPERATOR_EQUAL], [\$criterium = self::CRITERIUM_AND]) [*line 239*]

Function Parameters:

- *string* **\$field** Field name.
- *mixed* **\$value** Literal value.
- *int* **\$operator** Operator used for comparsion (equal by default).
- *int* **\$criterium** Criterium merging method (AND by default).

Compares field with a literal value.

Compares field with a literal value.

- **Version** 0.0.5
- **Since** 0.0.5
- **Access** public

array function OTS_SQLFilter::getTables() [*line 251*]

Returns list of all tables used by filter.

Returns list of all tables used by filter.

This is required for FROM clause.

- **Version** 0.0.5
- **Since** 0.0.5
- **Access** public

mixed function OTS_SQLFilter::__get(\$name) [*line 309*]

Function Parameters:

- *string* **\$name** Property name.

Magic PHP5 method.

Magic PHP5 method.

- **Version** 0.1.0
- **Version** 0.0.5
- **Throws** OutOfBoundsException For non-supported properties.
- **Since** 0.1.0
- **Since** 0.0.5
- **Access** public

array function OTS_SQLFilter::__sleep() [*line 82*]

Magic PHP5 method.

Magic PHP5 method.

Allows object serialisation.

- **Version** 0.0.5
- **Since** 0.0.5
- **Access** public

string function OTS_SQLFilter::__toString() [*line 96*]

Returns string representation of WHERE clause.

Returns string representation of WHERE clause.

Returned string can be easily inserted into SQL query.

- **Version** 0.1.0
- **Version** 0.0.5
- **Since** 0.0.5
- **Access** public

Class OTS_Toolbox

[line 20]

Toolbox for common operations.

Toolbox for common operations.

- **Package** POT
- **Version** 0.1.1
- **Since** 0.1.1

int function OTS_Toolbox::experienceForLevel(\$level, [\$experience = 0]) *[line 29]*

Function Parameters:

- *int* **\$level** Level for which experience should be calculated.
- *int* **\$experience** Current experience points.

Calculates experience points needed for given level.

Calculates experience points needed for given level.

- **Version** 0.1.1
- **Static**
- **Since** 0.1.1
- **Access** public

int function OTS_Toolbox::levelForExperience(\$experience) [*line 42*]

Function Parameters:

- *int* **\$experience** Current experience points.

Finds out which level user have basing on his/her experience.

Finds out which level user have basing on his/her experience.

PHP doesn't support complex numbers natively so solving third-level polynomials would be quite hard...

- **Version** 0.1.1
- **Static**
- **Since** 0.1.1
- **Access** public

Class OTS_VocationsList

[*line 20*]

Wrapper for vocations.

Wrapper for vocations.xml file.

- **Package** POT
- **Version** 0.1.0
- **Since** 0.1.0

Constructor *void* function OTS_VocationsList::__construct(\$file) [*line 36*]

Function Parameters:

- *string* **\$file** vocations.xml file location.

Loads vocations list.

Loads vocations list.

Loads vocations list from given file.

- **Version** 0.1.0
- **Since** 0.1.0
- **Access** public

int function OTS_VocationsList::count() [*line 104*]

Returns amount of vocations loaded.

Returns amount of vocations loaded.

- **Version** 0.1.0
- **Since** 0.1.0

- **Access** public

ArrayIterator function `OTS_VocationsList::getIterator()` [*line 114*]

Returns iterator handle for loops.

Returns iterator handle for loops.

- **Version** 0.1.0
- **Since** 0.1.0
- **Access** public

int|bool function `OTS_VocationsList::getVocationId($name)` [*line 76*]

Function Parameters:

- *string* **\$name** Vocation.

Returns vocation's ID.

Returns vocation's ID.

- **Version** 0.1.0
- **Since** 0.1.0
- **Access** public

string|bool function `OTS_VocationsList::getVocationName($id)` [*line 87*]

Function Parameters:

- *int* **\$id** Vocation ID.

Returns name of given vocation's ID.

Returns name of given vocation's ID.

- **Version** 0.1.0
- **Since** 0.1.0
- **Access** public

bool function OTS_VocationsList::offsetExists(\$offset) [*line 125*]

Function Parameters:

- *string|int* **\$offset** Array key.

Checks if given element exists.

Checks if given element exists.

- **Version** 0.1.0
- **Since** 0.1.0
- **Access** public

mixed function OTS_VocationsList::offsetGet(\$offset) [*line 145*]

Function Parameters:

- *string|int* **\$offset** Array key.

Returns item from given position.

Returns item from given position.

- **Version** 0.1.0
- **Since** 0.1.0
- **Access** public

void function OTS_VocationsList::offsetSet(\$offset, \$value) [*line 174*]

Function Parameters:

- *string|int* **\$offset** Array key.
- *mixed* **\$value** Field value.

This method is implemented for ArrayAccess interface.

This method is implemented for ArrayAccess interface. In fact you can't write/append to vocations list. Any call to this method will cause E_OTS_ReadOnly raise.

- **Version** 0.1.0
- **Throws** E_OTS_ReadOnly Always - this class is read-only.
- **Since** 0.1.0
- **Access** public

void function OTS_VocationsList::offsetUnset(\$offset) [*line 185*]

Function Parameters:

- *string|int* **\$offset** Array key.

This method is implemented for ArrayAccess interface.

This method is implemented for ArrayAccess interface. In fact you can't write/append to vocations list. Any call to this method will cause E_OTS_ReadOnly raise.

- **Version** 0.1.0
- **Throws** E_OTs_ReadOnly Always - this class is read-only.
- **Since** 0.1.0
- **Access** public

void function OTS_VocationsList::__set_state(\$properties) [*line 57*]

Function Parameters:

- *array* **\$properties** List of object properties.

Magic PHP5 method.

Magic PHP5 method.

Allows object importing from [var_export\(\)](#).

- **Version** 0.1.0
- **Since** 0.1.0
- **Access** public

Class OTS_XTEA

[*line 24*]

XTEA encryption/decryption mechanism.

XTEA encryption/decryption mechanism.

This code bases in large part on Jeroen Derks'es Crypt_Xtea's source code.

- **Package** POT
- **Version** 0.1.2
- **Since** 0.1.2

Constructor *void* function OTS_XTEA::__construct(\$key) [line 40]

Function Parameters:

- *string* **\$key** Encryption key to be used.

Initializes new encryption session.

Initializes new encryption session.

Note: Your key must be exactly 128bit length (16 characters)! Class will not resize it for you.

- **Version** 0.1.2
- **Since** 0.1.2
- **Access** public

string function OTS_XTEA::decrypt(\$message) [line 106]

Function Parameters:

- *string* **\$message** Encrypted message.

Decrypt XTEA-encrypted string.

Decrypt XTEA-encrypted string.

- **Version** 0.1.2

- **Since** 0.1.2
- **Access** public

string function OTS_XTEA::encrypt(\$message) [*line 51*]

Function Parameters:

- *string* **\$message** Data to encrypt.

Encrypt a string with XTEA algorithm.

Encrypt a string with XTEA algorithm.

- **Version** 0.1.2
- **Since** 0.1.2
- **Access** public

Class POT

[*line 34*]

Main POT class.

Main POT class.

- **Package** POT
- **Version** 0.1.2
- **Version** 0.0.1
- **Since** 0.0.1

POT::BAN_ACCOUNT

= 3 *[line 265]*

Account ban.

Account ban.

- **Version** 0.0.5
- **Version** 0.0.1
- **Since** 0.0.1
- **Since** 0.0.5

POT::BAN_IP

= 1 *[line 251]*

IP ban.

IP ban.

- **Version** 0.0.5
- **Version** 0.0.1
- **Since** 0.0.1
- **Since** 0.0.5

POT::BAN_PLAYER

= 2 *[line 258]*

Player ban.

Player ban.

- **Version** 0.0.5
- **Version** 0.0.1
- **Since** 0.0.1
- **Since** 0.0.5

POT::DB_MYSQL

= 1 *[line 39]*

MySQL driver.
MySQL driver.

- **Version** 0.0.1
- **Since** 0.0.1

POT::DB_ODBC

= 4 *[line 57]*

ODBC driver.
ODBC driver.

- **Version** 0.0.4
- **Version** 0.0.1
- **Since** 0.0.1
- **Since** 0.0.4

POT::DB_PGSQL

= 3 *[line 50]*

PostgreSQL driver.
PostgreSQL driver.

- **Version** 0.0.4
- **Version** 0.0.1
- **Since** 0.0.1
- **Since** 0.0.4

POT::DB_SQLITE

= 2 *[line 43]*

SQLite driver.
SQLite driver.

- **Version** 0.0.1
- **Since** 0.0.1

POT::DEPOT_SID_FIRST

= 100 *[line 243]*

First depot item sid.
First depot item sid.

- **Version** 0.0.4
- **Version** 0.0.1
- **Since** 0.0.1
- **Since** 0.0.4

POT::DIRECTION_EAST

= 1 [*line 106*]

East.

East.

- **Version** 0.0.1
- **Since** 0.0.1

POT::DIRECTION_NORTH

= 0 [*line 102*]

North.

North.

- **Version** 0.0.1
- **Since** 0.0.1

POT::DIRECTION_SOUTH

= 2 [*line 110*]

South.

South.

- **Version** 0.0.1
- **Since** 0.0.1

POT::DIRECTION_WEST

= 3 [*line 114*]

West.

West.

- **Version** 0.0.1
- **Since** 0.0.1

POT::ORDER_ASC

= 1 [*line 273*]

Ascencind sorting order.

Ascencind sorting order.

- **Version** 0.0.5
- **Version** 0.0.1
- **Since** 0.0.1
- **Since** 0.0.5

POT::ORDER_DESC

= 2 [*line 280*]

Descending sorting order.

Descending sorting order.

- **Version** 0.0.5
- **Version** 0.0.1
- **Since** 0.0.1
- **Since** 0.0.5

POT::SEX_FEMALE

= 0 [*line 62*]

Female gender.

Female gender.

- **Version** 0.0.1
- **Since** 0.0.1

POT::SEX_MALE

= 1 [*line 66*]

Male gender.

Male gender.

- **Version** 0.0.1
- **Since** 0.0.1

POT::SKILL_AXE

= 3 *[line 143]*

Axe fighting.
Axe fighting.

- **Version** 0.0.2
- **Version** 0.0.1
- **Since** 0.0.1
- **Since** 0.0.2

POT::SKILL_CLUB

= 1 *[line 129]*

Club fighting.
Club fighting.

- **Version** 0.0.2
- **Version** 0.0.1
- **Since** 0.0.1
- **Since** 0.0.2

POT::SKILL_DISTANCE

= 4 *[line 150]*

Distance fighting.
Distance fighting.

- **Version** 0.0.2
- **Version** 0.0.1
- **Since** 0.0.1
- **Since** 0.0.2

POT::SKILL_FISHING

= 6 *[line 164]*

Fishing.
Fishing.

- **Version** 0.0.2
- **Version** 0.0.1
- **Since** 0.0.1
- **Since** 0.0.2

POT::SKILL_FIST

= 0 *[line 122]*

Fist fighting.
Fist fighting.

- **Version** 0.0.2
- **Version** 0.0.1

- **Since** 0.0.1
- **Since** 0.0.2

POT::SKILL_SHIELDING

= 5 [*line 157*]

Shielding.
Shielding.

- **Version** 0.0.2
- **Version** 0.0.1
- **Since** 0.0.1
- **Since** 0.0.2

POT::SKILL_SWORD

= 2 [*line 136*]

Sword fighting.
Sword fighting.

- **Version** 0.0.2
- **Version** 0.0.1
- **Since** 0.0.1
- **Since** 0.0.2

POT::SLOT_AMMO

= 10 [*line 235*]

Ammunition slot.

Ammunition slot.

- **Version** 0.0.3
- **Version** 0.0.1
- **Since** 0.0.1
- **Since** 0.0.3

POT::SLOT_ARMOR

= 4 [*line 193*]

Armor slot.

Armor slot.

- **Version** 0.0.3
- **Version** 0.0.1
- **Since** 0.0.1
- **Since** 0.0.3

POT::SLOT_BACKPACK

= 3 [*line 186*]

Backpack slot.

Backpack slot.

- **Version** 0.0.3

- **Version** 0.0.1
- **Since** 0.0.1
- **Since** 0.0.3

POT::SLOT_FEET

= 8 [*line 221*]

Boots slot.

Boots slot.

- **Version** 0.0.3
- **Version** 0.0.1
- **Since** 0.0.1
- **Since** 0.0.3

POT::SLOT_HEAD

= 1 [*line 172*]

Head slot.

Head slot.

- **Version** 0.0.3
- **Version** 0.0.1
- **Since** 0.0.1
- **Since** 0.0.3

POT::SLOT_LEFT

= 6 *[line 207]*

Left hand slot.

Left hand slot.

- **Version** 0.0.3
- **Version** 0.0.1
- **Since** 0.0.1
- **Since** 0.0.3

POT::SLOT_LEGS

= 7 *[line 214]*

Legs slot.

Legs slot.

- **Version** 0.0.3
- **Version** 0.0.1
- **Since** 0.0.1
- **Since** 0.0.3

POT::SLOT_NECKLACE

= 2 *[line 179]*

Necklace slot.

Necklace slot.

- **Version** 0.0.3
- **Version** 0.0.1
- **Since** 0.0.1
- **Since** 0.0.3

POT::SLOT_RIGHT

= 5 [*line 200*]

Right hand slot.
Right hand slot.

- **Version** 0.0.3
- **Version** 0.0.1
- **Since** 0.0.1
- **Since** 0.0.3

POT::SLOT_RING

= 9 [*line 228*]

Ring slot.
Ring slot.

- **Version** 0.0.3
- **Version** 0.0.1
- **Since** 0.0.1
- **Since** 0.0.3

POT::SPELL_CONJURE

= 2 [*line 305*]

Conjure spell.
Conjure spell.

- **Version** 0.0.7
- **Version** 0.0.1
- **Deprecated** 0.1.0 Use OTS_SpellsList::SPELL_CONJURE.
- **Since** 0.0.7
- **Since** 0.0.1

POT::SPELL_INSTANT

= 1 [*line 297*]

Instant spell.
Instant spell.

- **Version** 0.0.7
- **Version** 0.0.1
- **Deprecated** 0.1.0 Use OTS_SpellsList::SPELL_INSTANT.
- **Since** 0.0.7
- **Since** 0.0.1

POT::SPELL_RUNE

= 0 [*line 289*]

Rune spell.
Rune spell.

- **Version** 0.0.7
- **Version** 0.0.1
- **Deprecated** 0.1.0 Use OTS_SpellsList::SPELL_RUNE.
- **Since** 0.0.7
- **Since** 0.0.1

POT::VOCATION_DRUID

= 2 *[line 85]*

Druid.

Druid.

- **Version** 0.0.1
- **Deprecated** 0.0.5 Vocations are now loaded dynamicly from vocations.xml file.
- **Since** 0.0.1

POT::VOCATION_KNIGHT

= 4 *[line 97]*

Knight.

Knight.

- **Version** 0.0.1
- **Deprecated** 0.0.5 Vocations are now loaded dynamicly from vocations.xml file.

- **Since** 0.0.1

POT::VOCATION_NONE

= 0 *[line 73]*

None vocation.

None vocation.

- **Version** 0.0.1
- **Deprecated** 0.0.5 Vocations are now loaded dynamicly from vocations.xml file.
- **Since** 0.0.1

POT::VOCATION_PALADIN

= 3 *[line 91]*

Paladin.

Paladin.

- **Version** 0.0.1
- **Deprecated** 0.0.5 Vocations are now loaded dynamicly from vocations.xml file.
- **Since** 0.0.1

POT::VOCATION_SORCERER

= 1 *[line 79]*

Sorcerer.

Sorcerer.

- **Version** 0.0.1
- **Deprecated** 0.0.5 Vocations are now loaded dynamicly from vocations.xml file.
- **Since** 0.0.1

bool function POT::areHousesLoaded() [*line 1084*]

Checks if houses are loaded.

Checks if houses are loaded.

- **Version** 0.1.0
- **Version** 0.0.1
- **Since** 0.0.1
- **Since** 0.1.0
- **Access** public

bool function POT::areItemsLoaded() [*line 1217*]

Checks if items are loaded.

Checks if items are loaded.

- **Version** 0.1.0
- **Version** 0.0.1
- **Since** 0.0.1
- **Since** 0.1.0
- **Access** public

bool function POT::areMonstersLoaded() [*line 812*]

Checks if monsters are loaded.

Checks if monsters are loaded.

- **Version** 0.1.0
- **Version** 0.0.1
- **Since** 0.0.1
- **Since** 0.1.0
- **Access** public

bool function POT::areSpellsLoaded() [*line 897*]

Checks if spells are loaded.

Checks if spells are loaded.

- **Version** 0.1.0
- **Version** 0.0.1
- **Since** 0.0.1
- **Since** 0.1.0
- **Access** public

bool function POT::areVocationsLoaded() [*line 706*]

Checks if vocations are loaded.

Checks if vocations are loaded.

- **Version** 0.1.0
- **Version** 0.0.1

- **Since** 0.0.1
- **Since** 0.1.0
- **Access** public

void function POT::banIP(\$ip, [\$mask = '255.255.255.255'], [\$time = 0]) [line 579]

Function Parameters:

- *string* **\$ip** IP to ban.
- *string* **\$mask** Mask for ban (by default bans only given IP).
- *int* **\$time** Time for time until expires (0 - forever).

Bans given IP number.

Bans given IP number.

Adds IP/mask ban. You can call this function with only one parameter to ban only given IP address without expiration.

- **Version** 0.0.5
- **Version** 0.0.1
- **Since** 0.0.1
- **Since** 0.0.5
- **Access** public

void function POT::connect(\$driver, \$params) [line 429]

connect.php

```

1  <?php
2
3  /**
4   * @ignore
5   * @package examples
6   * @author Wrzasq <wrzasq@gmail.com>
7   * @copyright 2007 (C) by Wrzasq
8   * @license http://www.gnu.org/licenses/lgpl-3.0.txt GNU Lesser General Public License, Version 3
9   */
10
11 // includes POT main file
12 include('../classes/OTS.php');
```

```

13
14 // you can easily store such structure in config.php
15 $config = array(
16     'driver' =>     POT::DB_MYSQL,
17     'prefix' =>     '',
18     'host' =>       'localhost',
19     'user' =>       'wrzasq',
20     'password' =>   '',
21     'database' =>   'otserv'
22 );
23
24 // connects to database
25 $ots = POT::getInstance();
26 $ots->connect(null, $config);
27 // could be: $ots->connect(POT::DB_MYSQL, $config);
28
29 ?>

```

Function Parameters:

- *int|null \$driver* Database driver type.
- *array \$params* Connection info.

Connects to database.

Connects to database.

Creates OTServ database connection object.

First parameter is one of database driver constants values. Currently MySQL, SQLite, PostgreSQL and ODBC drivers are supported.

This parameter can be null, then you have to specify 'driver' parameter.

Such way is comfortable to store entire database configuration in one array and possibly runtime evaluation and/or configuration file saving.

For parameters list see driver documentation. Common parameters for all drivers are:

- *driver* - optional, specifies driver, applies when *\$driver* method parameter is *null*
- *prefix* - optional, prefix for database tables, use if you have more then one OTServ installed on one database.

- **Version** 0.1.2
- **Version** 0.0.1
- **Throws** E_OTS_Generic When driver is not supported or not supported.
- **Throws** LogicException When PDO extension is not loaded.
- **Since** 0.0.1
- **Access** public

- **Example**

OTS_SQLFilter function POT::createFilter() [*line 672*]

Creates lists filter.

Creates lists filter.

- **Version** 0.1.0
- **Version** 0.0.1
- **Deprecated** 0.1.0 Create objects directly from now.
- **Since** 0.0.5
- **Since** 0.0.1
- **Access** public

IOTS_DAO function POT::createObject(\$class) [*line 488*]

Function Parameters:

- *string* **\$class** Class name.

Creates OTServ DAO class instance.

Creates OTServ DAO class instance.

- **Version** 0.1.0
- **Version** 0.0.1
- **Deprecated** 0.1.0 Create objects directly from now.
- **Since** 0.0.1
- **Access** public

OTS_Spell|null function POT::getConjure(\$name) [*line 1044*]

Function Parameters:

- *string* **\$name** Spell name.

Returns given conjure spell.

Returns given conjure spell.

- **Version** 0.1.0
- **Version** 0.0.1
- **Throws** E_OTS_NotLoaded If spells list is not loaded.
- **Since** 0.0.7
- **Since** 0.0.1
- **Access** public

array function POT::getConjuresList() [*line 1023*]

Returns list of conjure spells.

Returns list of conjure spells.

- **Version** 0.1.0
- **Version** 0.0.1
- **Throws** E_OTS_NotLoaded If spells list is not loaded.
- **Since** 0.0.7
- **Since** 0.0.1
- **Access** public

PDO function POT::getDBHandle() *[line 563]*

Returns database connection handle.

Returns database connection handle.

At all you shouldn't use this method and work with database using POT classes, but it may be sometime necessary to use direct database access (mainly until POT won't provide many important features).

It is also important as serialised objects after unserialisation needs to be re-initialised with database connection.

- **Version** 0.0.4
- **Version** 0.0.1
- **Since** 0.0.1
- **Since** 0.0.4
- **Access** public

IOTS_Display function POT::getDisplayDriver() *[line 1543]*

Returns current display driver.

Returns current display driver.

- **Version** 0.1.0
- **Version** 0.0.1
- **Throws** E_OTS_NotLoaded If display driver is not loaded.
- **Since** 0.1.0
- **Since** 0.0.1
- **Access** public

OTS_House|null function POT::getHouse(\$id) [*line 1129*]

Function Parameters:

- *int* **\$id** House ID.

Returns house information.

Returns house information.

- **Version** 0.1.0
- **Version** 0.0.1
- **Throws** E_OTS_NotLoaded If houses list is not loaded.
- **Since** 0.1.0
- **Since** 0.0.1
- **Access** public

int|bool function POT::getHouseId(\$name) [*line 1150*]

Function Parameters:

- *string* **\$name** House name.

Returns ID of house with given name.

Returns ID of house with given name.

- **Version** 0.1.0
- **Version** 0.0.1
- **Throws** E_OTS_NotLoaded If houses list is not loaded.
- **Since** 0.1.0
- **Since** 0.0.1

- **Access** public

OTS_HousesList function POT::getHousesList() [*line 1108*]

Returns list of laoded houses.

Returns list of laoded houses.

- **Version** 0.1.0
- **Version** 0.0.1
- **Throws** E_OTS_NotLoaded If houses list is not loaded.
- **Since** 0.1.0
- **Since** 0.0.1
- **Access** public

POT function POT::getInstance() [*line 312*]

Singleton.

Singleton.

- **Version** 0.0.1
- **Static**
- **Since** 0.0.1
- **Access** public

OTS_Spell function POT::getInstant(\$name) [*line 1003*]

Function Parameters:

- *string* **\$name** Spell name.

Returns given instant spell.

Returns given instant spell.

- **Version** 0.1.0
- **Version** 0.0.1
- **Throws** E_OTS_NotLoaded If spells list is not loaded.
- **Since** 0.0.7
- **Since** 0.0.1
- **Access** public

array function POT::getInstantsList() [*line 982*]

Returns list of instants.

Returns list of instants.

- **Version** 0.1.0
- **Version** 0.0.1
- **Throws** E_OTS_NotLoaded If spells list is not loaded.
- **Since** 0.0.7
- **Since** 0.0.1
- **Access** public

OTS_ItemsList function POT::getItemsList() [*line 1241*]

Returns list of loaded items.

Returns list of loaded items.

- **Version** 0.1.0
- **Version** 0.0.1
- **Throws** E_OTS_NotLoaded If items list is not loaded.
- **Since** 0.1.0
- **Since** 0.0.1
- **Access** public

OTS_ItemType|null function POT::getItemType(\$id) [*line 1262*]

Function Parameters:

- *int* **\$id** Item type ID.

Returns item type instance.

Returns item type instance.

- **Version** 0.1.0
- **Version** 0.0.1
- **Throws** E_OTS_NotLoaded If items list is not loaded.
- **Since** 0.1.0
- **Since** 0.0.1
- **Access** public

int|bool function POT::getItemTypeId(\$name) [*line 1283*]

Function Parameters:

- *string* **\$name** Item type name.

Returns ID of type with given name.
Returns ID of type with given name.

- **Version** 0.1.0
- **Version** 0.0.1
- **Throws** E_OTS_NotLoaded If items list is not loaded.
- **Since** 0.1.0
- **Since** 0.0.1
- **Access** public

OTS_OTBMFile function POT::getMap() [*line 1377*]

Returns loaded map.
Returns loaded map.

- **Version** 0.1.0
- **Version** 0.0.1
- **Throws** E_OTS_NotLoaded If map is not loaded.
- **Since** 0.1.0
- **Since** 0.0.1
- **Access** public

string function POT::getMapDescription() [*line 1437*]

Returns map description.
Returns map description.

- **Version** 0.1.0
- **Version** 0.0.1
- **Throws** E_OTS_NotLoaded If map is not loaded.
- **Since** 0.1.0
- **Since** 0.0.1
- **Access** public

int function POT::getMapHeight() [*line 1417*]

Returns map height.

Returns map height.

- **Version** 0.1.0
- **Version** 0.0.1
- **Throws** E_OTS_NotLoaded If map is not loaded.
- **Since** 0.1.0
- **Since** 0.0.1
- **Access** public

int function POT::getMapWidth() [*line 1397*]

Returns map width.

Returns map width.

- **Version** 0.1.0
- **Version** 0.0.1
- **Throws** E_OTS_NotLoaded If map is not loaded.
- **Since** 0.1.0
- **Since** 0.0.1

- **Access** public

OTS_Monster|*null* function POT::getMonster(\$name) [*line 857*]

Function Parameters:

- *string* **\$name** Monster name.

Returns loaded data of given monster.

Returns loaded data of given monster.

- **Version** 0.1.0
- **Version** 0.0.1
- **Throws** E_OTS_NotLoaded If monsters list is not loaded.
- **Since** 0.0.6
- **Since** 0.0.1
- **Access** public

OTS_MonstersList function POT::getMonstersList() [*line 836*]

Returns list of loaded monsters.

Returns list of loaded monsters.

- **Version** 0.1.0
- **Version** 0.0.1
- **Throws** E_OTS_NotLoaded If monsters list is not loaded.
- **Since** 0.0.6
- **Since** 0.0.1

- **Access** public

OTS_Spell null function POT::getRune(\$name) [line 962]

Function Parameters:

- *string* **\$name** Rune name.

Returns given rune spell.

Returns given rune spell.

- **Version** 0.1.0
- **Version** 0.0.1
- **Throws** E_OTS_NotLoaded If spells list is not loaded.
- **Since** 0.0.7
- **Since** 0.0.1
- **Access** public

array function POT::getRunesList() [line 941]

Returns list of runes.

Returns list of runes.

- **Version** 0.1.0
- **Version** 0.0.1
- **Throws** E_OTS_NotLoaded If spells list is not loaded.
- **Since** 0.0.7
- **Since** 0.0.1
- **Access** public

OTS_SpellsList function POT::getSpellsList() [*line 921*]

Returns list of loaded spells.

Returns list of loaded spells.

- **Version** 0.1.0
- **Version** 0.0.1
- **Throws** E_OTS_NotLoaded If spells list is not loaded.
- **Since** 0.1.0
- **Since** 0.0.1
- **Access** public

int|bool function POT::getTownId(\$name) [*line 1458*]

Function Parameters:

- *string* **\$name** Town.

Returns town's ID.

Returns town's ID.

- **Version** 0.1.0
- **Version** 0.0.1
- **Throws** E_OTS_NotLoaded If map is not loaded.
- **Since** 0.1.0
- **Since** 0.0.1
- **Access** public

string|bool function POT::getTownName(\$id) [*line 1479*]

Function Parameters:

- *int* **\$id** Town ID.

Returns name of given town's ID.

Returns name of given town's ID.

- **Version** 0.1.0
- **Version** 0.0.1
- **Throws** E_OTS_NotLoaded If map is not loaded.
- **Since** 0.1.0
- **Since** 0.0.1
- **Access** public

int|bool function POT::getVocationId(\$name) [*line 751*]

Function Parameters:

- *string* **\$name** Vocation.

Returns vocation's ID.

Returns vocation's ID.

- **Version** 0.1.0
- **Version** 0.0.1
- **Throws** E_OTS_NotLoaded If vocations list is not loaded.

- **Since** 0.0.5
- **Since** 0.0.1
- **Access** public

string|bool function POT::getVocationName(\$id) [*line 772*]

Function Parameters:

- *int* **\$id** Vocation ID.

Returns name of given vocation's ID.

Returns name of given vocation's ID.

- **Version** 0.1.0
- **Version** 0.0.1
- **Throws** E_OTS_NotLoaded If vocations list is not loaded.
- **Since** 0.0.5
- **Since** 0.0.1
- **Access** public

OTS_VocationsList function POT::getVocationsList() [*line 730*]

Returns vocations list object.

Returns vocations list object.

- **Version** 0.1.0
- **Version** 0.0.1
- **Throws** E_OTS_NotLoaded If vocations list is not loaded.
- **Since** 0.0.5

- **Since** 0.0.1
- **Access** public

bool function POT::isDisplayDriverLoaded() [*line 1519*]

Checks if any display driver is loaded.

Checks if any display driver is loaded.

- **Version** 0.1.0
- **Version** 0.0.1
- **Since** 0.0.1
- **Since** 0.1.0
- **Access** public

bool function POT::isIPBanned(\$ip) [*line 647*]

Function Parameters:

- *string* **\$ip** IP to ban.

Checks if given IP is banned.

Checks if given IP is banned.

- **Version** 0.0.5
- **Version** 0.0.1
- **Since** 0.0.1
- **Since** 0.0.5
- **Access** public

bool function POT::isMapLoaded() [*line 1353*]

Checks if OTBM is loaded.

Checks if OTBM is loaded.

- **Version** 0.1.0
- **Version** 0.0.1
- **Since** 0.0.1
- **Since** 0.1.0
- **Access** public

void function POT::loadClass(\$class) [*line 387*]

Function Parameters:

- *string* **\$class** Class name.

Loads POT class file.

Loads POT class file.

Runtime class loading on demand - usefull for __autoload() function.

Note: Since 0.0.2 version this function is suitable for spl_autoload_register().

Note: Since 0.0.3 version this function handles also exceptions.

- **Version** 0.0.3
- **Version** 0.0.1
- **Since** 0.0.1
- **Access** public

void function POT::loadHouses(\$path) [line 1072]

Function Parameters:

- *string* **\$path** Houses file.

Loads houses list file.

Loads houses list file.

- **Version** 0.1.0
- **Version** 0.0.1
- **Since** 0.0.1
- **Since** 0.1.0
- **Access** public

void function POT::loadItems(\$path) [line 1197]

Function Parameters:

- *string* **\$path** Items information directory.

Loads items list.

Loads items list.

- **Version** 0.1.0
- **Version** 0.0.1
- **Since** 0.0.1
- **Since** 0.1.0

- **Access** public

void function POT::loadMap(\$path) [line 1332]

Function Parameters:

- *string* **\$path** Map file path.

Loads OTBM map.

Loads OTBM map.

Note: This method will also load houses list associated with map.

- **Version** 0.1.0
- **Version** 0.0.1
- **Since** 0.0.1
- **Since** 0.1.0
- **Access** public

void function POT::loadMonsters(\$path) [line 800]

Function Parameters:

- *string* **\$path** Monsters directory.

Loads monsters mapping file.

Loads monsters mapping file.

- **Version** 0.1.0
- **Version** 0.0.1

- **Since** 0.0.1
- **Since** 0.0.6
- **Access** public

void function POT::loadSpells(\$file) [line 885]

Function Parameters:

- *string* **\$file** Spells file name.

Loads spells list.

Loads spells list.

- **Version** 0.1.0
- **Version** 0.0.1
- **Since** 0.0.1
- **Since** 0.0.7
- **Access** public

void function POT::loadVocations(\$file) [line 693]

Function Parameters:

- *string* **\$file** vocations.xml file location.

Loads vocations list.

Loads vocations list.

- **Version** 0.1.0

- **Version** 0.0.1
- **Since** 0.0.1
- **Since** 0.0.5
- **Access** public

OTS_InfoRespond/bool function POT::serverStatus(\$server, \$port) [line 506]

example

```

1  <?php
2
3  /**
4   * @ignore
5   * @package examples
6   * @author Wrzasq <wrzasq@gmail.com>
7   * @copyright 2007 (C) by Wrzasq
8   * @license http://www.gnu.org/licenses/lgpl-3.0.txt GNU Lesser General Public License, Version 3
9   */
10
11 // to not repeat all that stuff
12 include('quickstart.php');
13
14 // server and port
15 $server = '127.0.0.1';
16 $port = 7171;
17
18 // queries server of status info
19 $status = $ots-> serverStatus($server, $port);
20
21 // offline
22 if(!$status)
23 {
24     echo 'Server ', $server, ' is offline.', "\n"
25 }
26 // displays various info
27 else
28 {
29     echo 'Server name: ', $status-> getName(), "\n"
30     echo 'Server owner: ', $status-> getOwner(), "\n"
31     echo 'Players online: ', $status-> getOnlinePlayers(), "\n"
32     echo 'Maximum allowed number of players: ', $status-> getMaxPlayers(), "\n"
33     echo 'Required client version: ', $status-> getClientVersion(), "\n"
34     echo 'All monsters: ', $status-> getMonstersCount(), "\n"
35     echo 'Server message: ', $status-> getMOTD(), "\n"
36 }
37
38 ?>

```

Function Parameters:

- *string* **\$server** Server IP/domain.
- *int* **\$port** OTServ port.

Queries server status.

Queries server status.

Sends 'info' packet to OTS server and return output.

- **Version** 0.0.2
- **Version** 0.0.1
- **Since** 0.0.1
- **Since** 0.0.2
- **Access** public
- **Example**

void function POT::setDisplayDriver(\$display) [line 1507]

Function Parameters:

- [*IOTS_Display*](#) **\$display** Display driver.

Sets display driver.

Sets display driver.

- **Version** 0.1.0
- **Version** 0.0.1
- **Since** 0.0.1
- **Since** 0.1.0
- **Access** public

void function POT::setItemsCache([\$cache = null]) [line 1176]

Function Parameters:

- [*IOTS_FileCache*](#) **\$cache** Cache handler (skip this parameter to reset cache handler to null).

Presets cache handler for items loader.

Presets cache handler for items loader.

- **Version** 0.0.1
- **Since** 0.0.1
- **Access** public

`void function POT::setMapCache([$cache = null]) [line 1309]`

Function Parameters:

- [IOTS_FileCache](#) **\$cache** Cache handler (skip this parameter to reset cache handler to null).

Presets cache handler for OTBM loader.

Presets cache handler for OTBM loader.

- **Version** 0.0.1
- **Since** 0.0.1
- **Access** public

`void function POT::setPOTPath($path) [line 343]`

fakeroot.php

```
1  <?php
2
3  /**
4   * @ignore
5   * @package examples
6   * @author Wrzasq <wrzasq@gmail.com>
7   * @copyright 2007 (C) by Wrzasq
8   * @license http://www.gnu.org/licenses/lgpl-3.0.txt GNU Lesser General Public License, Version 3
9   */
10
11 // this is the way you should work with POT if you moved main OTS.php file outside POT's directory
12 include('path/to/OTS.php');
13
14 // dont use 'new POT()'!!!
```



```

15  $ots = POT::getInstance();
16  $ots-> setPOTPath('../classes/');
17
18  /*
19   here comes your stuff...
20  */
21
22  ?>

```

Function Parameters:

- *string* **\$path** POT files path.

Set POT directory.

Set POT directory.

Use this method if you keep your POT package in different directory then this file.

- **Version** 0.0.1
- **Since** 0.0.1
- **Access** public
- **Example**

void function POT::unbanIP(\$ip, [\$mask = '255.255.255.255']) [*line 614*]

Function Parameters:

- *string* **\$ip** IP to ban.
- *string* **\$mask** Mask for ban (by default 255.255.255.255).

Deletes ban from given IP number.

Deletes ban from given IP number.

Removes given IP/mask ban.

- **Version** 0.0.5
- **Version** 0.0.1

- **Since** 0.0.1
- **Since** 0.0.5
- **Access** public

void function POT::unloadDisplayDriver() [line 1530]

Unloads display driver.

Unloads display driver.

- **Version** 0.1.0
- **Version** 0.0.1
- **Since** 0.0.1
- **Since** 0.1.0
- **Access** public

void function POT::unloadHouses() [line 1095]

Unloads houses list.

Unloads houses list.

- **Version** 0.1.0
- **Version** 0.0.1
- **Since** 0.0.1
- **Since** 0.1.0
- **Access** public

void function POT::unloadItems() [line 1228]

Unloads items list.

Unloads items list.

- **Version** 0.1.0
- **Version** 0.0.1
- **Since** 0.0.1
- **Since** 0.1.0
- **Access** public

void function POT::unloadMap() [*line 1364*]

Unloads OTBM map.

Unloads OTBM map.

- **Version** 0.1.0
- **Version** 0.0.1
- **Since** 0.0.1
- **Since** 0.1.0
- **Access** public

void function POT::unloadMonsters() [*line 823*]

Unloads monsters list.

Unloads monsters list.

- **Version** 0.1.0
- **Version** 0.0.1
- **Since** 0.0.1

- **Since** 0.1.0
- **Access** public

void function POT::unloadSpells() [*line 908*]

Unloads spells list.

Unloads spells list.

- **Version** 0.1.0
- **Version** 0.0.1
- **Since** 0.0.1
- **Since** 0.1.0
- **Access** public

void function POT::unloadVocations() [*line 717*]

Unloads vocations list.

Unloads vocations list.

- **Version** 0.1.0
- **Version** 0.0.1
- **Since** 0.0.1
- **Since** 0.1.0
- **Access** public

compat.php

POT compatibility assurance package.

POT compatibility assurance package.

This package makes you sure that POT scripts won't cause FATAL errors on PHP older PHP 5.x versions. However remember that some PHP features won't be enabled with it. For example if you have PHP 5.0.x, this package will define Countable interface for you so PHP will know it, but it won't allow you to use count(\$countableObject) structure.

- **Package** POT
- **Sub-Package** compat
- **Author** Wrzasq < wrzasq@gmail.com>
- **Version** 0.1.2+SVN
- **Version** 0.0.2
- **Copyright** 2007 - 2008 (C) by Wrzasq
- **Since** 0.0.2
- **License** [GNU Lesser General Public License, Version 3](#)

Appendices

Appendix A - Class Trees

Package POT

E_OTS_ErrorCode

- [E_OTS_ErrorCode](#)
 - [E_OTS_FileLoaderError](#)
 - [E_OTS_Generic](#)
 - [E_OTS_OTBMErrors](#)

E_OTS_NoDriver

- Exception
 - [E_OTS_NoDriver](#)

E_OTS_NotAContainer

- Exception
 - [E_OTS_NotAContainer](#)

E_OTS_NotLoaded

- Exception
 - [E_OTS_NotLoaded](#)

E_OTS_OutOfBuffer

- Exception
 - [E_OTs_OutOfBuffer](#)

E_OTs_ReadOnly

- Exception
 - [E_OTs_ReadOnly](#)

IOTS_Cipher

- [IOTS_Cipher](#)

IOTS_DAO

- [IOTS_DAO](#)

IOTS_DB

- [IOTS_DB](#)

IOTS_Display

- [IOTS_Display](#)

IOTS_FileCache

- [IOTS_FileCache](#)
 - [IOTS_ItemsCache](#)

IOTS_GuildAction

- [IOTS_GuildAction](#)

OTS_Admin

- [OTS_Admin](#)

OTS_Base_DAO

- [OTS_Base_DAO](#)
 - [OTS_House](#)
 - [OTS_Row_DAO](#)
 - [OTS_Account](#)
 - [OTS_Group](#)
 - [OTS_Guild](#)
 - [OTS_GuildRank](#)
 - [OTS_Player](#)
- [OTS_SQLFilter](#)

OTS_Base_List

- [OTS_Base_List](#)
 - [OTS_Accounts_List](#)
 - [OTS_Groups_List](#)
 - [OTS_GuildRanks_List](#)
 - [OTS_Guilds_List](#)
 - [OTS_Players_List](#)

OTS_BinaryTools

- [OTS_BinaryTools](#)

OTS_Buffer

- [OTS_Buffer](#)
 - [OTS_FileNode](#)

OTS_DB_MySQL

- PDO
 - [OTS_DB_MySQL](#)

OTS_DB_ODBC

- PDO
 - [OTS_DB_ODBC](#)

OTS_DB_PostgreSQL

- PDO
 - [OTS_DB_PostgreSQL](#)

OTS_DB_SQLite

- PDO
 - [OTS_DB_SQLite](#)

OTS_FileLoader

- [OTS_FileLoader](#)
 - [OTS_ItemsList](#)
 - [OTS_OTBMFile](#)

OTS_HousesList

- [OTS_HousesList](#)

OTS_InfoRespond

- DOMDocument
 - [OTS_InfoRespond](#)

OTS_Item

- [OTS_Item](#)
 - [OTS_Container](#)

OTS_ItemType

- [OTS_ItemType](#)

OTS_MapCoords

- [OTS_MapCoords](#)

OTS_Monster

- DOMDocument
 - [OTS_Monster](#)

OTS_MonstersList

- [OTS_MonstersList](#)

OTS_RSA

- [OTS_RSA](#)

OTS_Spell

- [OTS_Spell](#)

OTS_SpellsList

- [OTS_SpellsList](#)

OTS_SQLField

- [OTS_SQLField](#)

OTS_Toolbox

- [OTS_Toolbox](#)

OTS_VocationsList

- [OTS_VocationsList](#)

OTS_XTEA

- [OTS_XTEA](#)

POT

- [POT](#)

Appendix B - README/CHANGELOG/INSTALL

LICENSE

GNU LESSER GENERAL PUBLIC LICENSE Version 3, 29 June 2007

Copyright (C) 2007 Free Software Foundation, Inc. <<http://fsf.org/>>
Everyone is permitted to copy and distribute verbatim copies
of this license document, but changing it is not allowed.

This version of the GNU Lesser General Public License incorporates
the terms and conditions of version 3 of the GNU General Public
License, supplemented by the additional permissions listed below.

0. Additional Definitions.

As used herein, "this License" refers to version 3 of the GNU Lesser
General Public License, and the "GNU GPL" refers to version 3 of the GNU
General Public License.

"The Library" refers to a covered work governed by this License,
other than an Application or a Combined Work as defined below.

An "Application" is any work that makes use of an interface provided
by the Library, but which is not otherwise based on the Library.
Defining a subclass of a class defined by the Library is deemed a mode
of using an interface provided by the Library.

A "Combined Work" is a work produced by combining or linking an
Application with the Library. The particular version of the Library
with which the Combined Work was made is also called the "Linked
Version".

The "Minimal Corresponding Source" for a Combined Work means the
Corresponding Source for the Combined Work, excluding any source code
for portions of the Combined Work that, considered in isolation, are
based on the Application, and not on the Linked Version.

The "Corresponding Application Code" for a Combined Work means the
object code and/or source code for the Application, including any data
and utility programs needed for reproducing the Combined Work from the
Application, but excluding the System Libraries of the Combined Work.

1. Exception to Section 3 of the GNU GPL.

You may convey a covered work under sections 3 and 4 of this License
without being bound by section 3 of the GNU GPL.

2. Conveying Modified Versions.

If you modify a copy of the Library, and, in your modifications, a
facility refers to a function or data to be supplied by an Application
that uses the facility (other than as an argument passed when the

facility is invoked), then you may convey a copy of the modified version:

- a) under this License, provided that you make a good faith effort to ensure that, in the event an Application does not supply the function or data, the facility still operates, and performs whatever part of its purpose remains meaningful, or
- b) under the GNU GPL, with none of the additional permissions of this License applicable to that copy.

3. Object Code Incorporating Material from Library Header Files.

The object code form of an Application may incorporate material from a header file that is part of the Library. You may convey such object code under terms of your choice, provided that, if the incorporated material is not limited to numerical parameters, data structure layouts and accessors, or small macros, inline functions and templates (ten or fewer lines in length), you do both of the following:

- a) Give prominent notice with each copy of the object code that the Library is used in it and that the Library and its use are covered by this License.
- b) Accompany the object code with a copy of the GNU GPL and this license document.

4. Combined Works.

You may convey a Combined Work under terms of your choice that, taken together, effectively do not restrict modification of the portions of the Library contained in the Combined Work and reverse engineering for debugging such modifications, if you also do each of the following:

- a) Give prominent notice with each copy of the Combined Work that the Library is used in it and that the Library and its use are covered by this License.
- b) Accompany the Combined Work with a copy of the GNU GPL and this license document.
- c) For a Combined Work that displays copyright notices during execution, include the copyright notice for the Library among these notices, as well as a reference directing the user to the copies of the GNU GPL and this license document.
- d) Do one of the following:
 - 0) Convey the Minimal Corresponding Source under the terms of this License, and the Corresponding Application Code in a form suitable for, and under terms that permit, the user to recombine or relink the Application with a modified version of the Linked Version to produce a modified Combined Work, in the manner specified by section 6 of the GNU GPL for conveying Corresponding Source.
 - 1) Use a suitable shared library mechanism for linking with the Library. A suitable mechanism is one that (a) uses at run time

a copy of the Library already present on the user's computer system, and (b) will operate properly with a modified version of the Library that is interface-compatible with the Linked Version.

e) Provide Installation Information, but only if you would otherwise be required to provide such information under section 6 of the GNU GPL, and only to the extent that such information is necessary to install and execute a modified version of the Combined Work produced by recombining or relinking the Application with a modified version of the Linked Version. (If you use option 4d0, the Installation Information must accompany the Minimal Corresponding Source and Corresponding Application Code. If you use option 4d1, you must provide the Installation Information in the manner specified by section 6 of the GNU GPL for conveying Corresponding Source.)

5. Combined Libraries.

You may place library facilities that are a work based on the Library side by side in a single library together with other library facilities that are not Applications and are not covered by this License, and convey such a combined library under terms of your choice, if you do both of the following:

- a) Accompany the combined library with a copy of the same work based on the Library, uncombined with any other library facilities, conveyed under the terms of this License.
- b) Give prominent notice with the combined library that part of it is a work based on the Library, and explaining where to find the accompanying uncombined form of the same work.

6. Revised Versions of the GNU Lesser General Public License.

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

Each version is given a distinguishing version number. If the Library as you received it specifies that a certain numbered version of the GNU Lesser General Public License "or any later version" applies to it, you have the option of following the terms and conditions either of that published version or of any later version published by the Free Software Foundation. If the Library as you received it does not specify a version number of the GNU Lesser General Public License, you may choose any version of the GNU Lesser General Public License ever published by the Free Software Foundation.

If the Library as you received it specifies that a proxy can decide whether future versions of the GNU Lesser General Public License shall apply, that proxy's public statement of acceptance of any version is permanent authorization for you to choose that version for the Library.

CHANGELOG

[0.1.2]

- * OTAdmin protocol support. <wrzasq>
- * Supports bank balance. <wrzasq>
- * getStorage()/setStorage() methods in OTS_Player class. <wrzasq>
- * More secured. <wrzasq>
- * Fixed buffered streams to use little-endian storage (like Open Tibia does). <wrzasq>
- * Implemented NetworkMessage. <wrzasq>

[0.1.1]

- * Loading groups by names. <wrzasq>
- * Constructors with optional identifier argument for automatical object loading. <wrzasq>
- * OTS_Toolbox class. <wrzasq>
- * Access calculations in OTS_Account class. <wrzasq>
- * E_OTS_General class. <wrzasq>

[0.1.0]

- * Houses support. <wrzasq>
- * No need to call POT::createObject(). <wrzasq>
- * Separated data/ directory resources loaders. <wrzasq>
- * Default data/ directory resources instances within POT class. <wrzasq>
- * Each data/ directory resource reader class implements an Iterator/IteratorAggregate, Countable and ArrayAccess interfaces. <wrzasq>
- * Uses IteratorAggregate interface instead of Iterator in some places (returns ArrayIterator as iterator). <wrzasq>
- * __get()/__set() implementation. <wrzasq>
- * __toString() as display drivers. <wrzasq>
- * Additional wrappers for resource binds. <wrzasq>
- * Some additional updates to match OTServ development. <wrzasq>
- * Fixed some typos. <wrzasq>
- * Code cleanup. <wrzasq>

[0.0.8]

- * Added items.xml and items.otb files support. <wrzasq>
- * Added Iterator and Countable interfaces implementation in OTS_OTBMFile class. <wrzasq>
- * Fixed quoting guild rank name while updating. <wrzasq>
- * Fixed custom fields reading in OTS_Guild class. <wrzasq>

[0.0.7]

- * Updated for last database changes. <wrzasq>
- * Even more PHP 5.0 compatibility. <wrzasq>
- * Added spells support. <wrzasq>
- * Added possibility to sort lists by custom fields, not only selected. <wrzasq>
- * Fixed typos. <wrzasq>

[0.0.6]

- * Updated for last database changes. <wrzasq>
- * Increased PHP 5.0 compatibility. <wrzasq>
- * Added generic binary formats reader with cache drivers. <wrzasq>
- * Added OTBM files basic support. <wrzasq>
- * Added monsters support. <wrzasq>
- * Added OTS_Player::getVocationName() method. <wrzasq>

[0.0.5]

- * Added support for vocations.xml file. <wrzasq>
- * Added support for bans. <wrzasq>

- * Added sorting and filtering for lists. <wrzasq>
- * Code grouped into base classes. <wrzasq>
- * Some code optimisation. <wrzasq>
- * Fixed typos. <wrzasq>

[0.0.4]

- * Added guild system support (guilds, ranks, invitations and requests drivers mechanisms). <wrzasq>
- * Added account group support. <wrzasq>
- * Added support for depot_id field (it is reserved in OTServ for further use). <wrzasq>
- * Added PostgreSQL and ODBC drivers. <wrzasq>
- * Added __sleep() and __wakeup() methods to allow POT objects to be stored in sessions. <wrzasq>
- * Added __clone() methods to allow save ID-losing cloning of POT objects. <wrzasq>
- * Added __set_state() methods. <wrzasq>
- * Updated players table structure. <wrzasq>
- * Dropped REGEXP operator bindings - not used anywhere. <wrzasq>
- * Fixed items loading and saving. <wrzasq>
- * Fixed typos. <wrzasq>

[0.0.3]

- * Added custom fields support. <wrzasq>
- * Added items and depots support. <wrzasq>
- * Added support for players PACC timestamps. <wrzasq>
- * Fixed loading skills. <wrzasq>
- * Replaced E_USER_* with exceptions. <wrzasq>
- * Uses fetchAll() in loops to prevent MySQL buffering problems. <wrzasq>
- * Restricted access to POT class constructor to make sure it won't be instantiated directly. <wrzasq>

[0.0.2]

- * Added "compat" library for POT. <wrzasq>
- * Added skills support in OTS_Player class. <wrzasq>
- * Added 'info' serverStatus() method and respond handler for server status protocol. <wrzasq>
- * Fixed `redskulltime` field name in OTS_Player. <wrzasq>
- * Fixed 'password' parameter for DB_MYSQL driver. <wrzasq>
- * Added find() to OTS_Account class to load accounts by their's e-mail addresses. <wrzasq>
- * POT class now automatically binds own __autoload() handler with spl_autoload_register(). <wrzasq>

[0.0.1]

- * Initial release. <wrzasq>

RULES

Zero rule: We use Unicode (UTF-8).

Of course we should handle input encoding respectively, but output and internal data/code are all written in UTF-8.

I. Coding rules to be followed:

[1] Never ever use global!

It's just the worst thing you can do in PHP scripts.

[2] Avoid using define - use class constants.

To group code better, to allow classes `__autoload()` handling.

[3] Use `isset()` instead of `is_null()`.

That has exactly same effect and we should follow the most simplies methods. Just to make code cleaner (however remember that `isset()` is a PHP language structure and has it's limitations!).

[4] Don't use functions - use class methods (except Compat package).

This will allow `__autoload()` handling for all routines as they will be members of classes.

[5] Use 4 spaces as tabulation.

Tabulator character can be differently displayed and generally four spaces makes code more readable.

[6] Always use brackets for blocks and leave them in separated lines in same nesting level that block instruction:

```
if(condition)
{
    for($i = 0; $i < $j; $i++)
    {
        statement;
    }
}
```

[7] Use single quotes inseed of double ones.

' are, in standard way, faster then " and it keeps code cleaner if you simply concat everything rather then inserting something like placeholders into string.

[8] Use spaces between parenthess and operators (except object member accessing operator):

```
$foo = $lol . $rotfl;
$foo .= $bar;
$obj = new Class( substr( str_replace( implode('.', $array), ',', '.'), 2) );
echo $obj->field;
echo $obj->method( rand() );
echo $obj->method($value);
```

[9] Use `<?php` opening tag.

It is the most reliable and standard way for starting PHP code.

[10] Use `isset(array[offset])` instead of `array_key_exists()`.

It saves alot of resources (relatively).

II. File naming:

[1] Use lowercase names for directories.

[2] Use fiels and directories in code in case-sensitive way.

Remember that probably this code will be mostly run on non-Windows platforms.

[3] Use existing directories structure.

Put classes into classes directory, tutorials into tutorials directory etc.

README

POT (PHP OTServ Toolkit) is a PHP toolkit for scripts that work with OTServ database.

===== About =====

This toolkit provides a way for PHP programmers that don't know SQL language to work with OTServ database.

For installation help check INSTALL file.

For usage tutorial/API documentation check <http://otserv-aac.info/> or documentation.pdf file.

===== Contact =====

In case of any contact needed, please use following e-mail address: wrzasq@gmail.com.

===== Files =====

classes/ - POT class files.
examples/ - example files for learning.
tutorials/ - phpDocumentor directory.
CHANGELOG - changes history.
INSTALL - installation tutorial.
LICENSE - POT license (GNU LGPL v3), if you don't accept it - don't use any of those scripts.
NEWS - changes in current release.
README - this readme file.
RULES - rules to be followed during developing contributed code.
Makefile - make input, for documentation generation.
documentation.pdf - phpDocumentor-generator documentation in PDF format.
compat.php - Compatibility assurance library.
test.php - phpUnit test suite.

===== Makefile =====

Makefile contains some targets for make that can help in development. Makefile requires following command-line commands:

php: PHP CLI interface.
phpdoc: phpDocumentor.
phpunit: PHPUnit testing framework.

Possible targets:

all: default one, runs all other targets (in order: clean, check, documentation, pdf, online, test, package).
clean: deletes documentation.
check: checks syntax of all PHP files.
documentation: generates HTML documentation.

pdf: generates PDF documentation.
online: OTServ-AAC website documentation template used.
test: runs test suite.
package: creates pot.tar.gz file for distribution purposes.

For more readable output of phpUnit test run:
php test.php

==== Credits =====

* Wrzasq <wrzasq@gmail.com> - project initiator, main developer.

For more info see AUTHORS file in OTServ tree.

INSTALL

POT is a toolkit which means you don't literally install it. You copy it's files and write code for it. All source files are located in classes/ subdirectory. Copy them to your script directory.

You can put main file - OTS.php in different directory then other files.

For information about how to include POT in your code see the documentation.

NEWS

What's new in 0.1.2 version?

* OTAdmin protocol support.

Full OTAdmin support including RSA and XTEA encryption, keys negotiations and logging in. All currently available OTAdmin commands (BROADCAST, CLOSE_SERVER, PAY_HOUSES and SHUTDOWN_SERVER) supported.

* Supports bank balance.

POT is always as up-to-date as it is possible. OTServ introduced new way of storing money in game - bank account like in real Tibia. It uses new database field. OTS_Player class contains new methods: getBalance() and setBalance(int \$balance). \$balance virtual property can be also used as an alias to those methods.

* getStorage()/setStorage() methods in OTS_Player class.

As requested by Proglin on <http://otfans.net/showthread.php?t=108150>.

* More secured.

Added some more stricted types controll. "Hey, you said POT is safe?!" - Yes it has always been safe, but now it is more idiot-friendly :P (in safety context) in code-based part.

- * Fixed buffered streams to use little-endian storage (like Open Tibia does).

Until now buffered streams used machine-dependent bits order which could produce problems while using POT on big-endian machines. Mainly when trying to use them for protocol packets. Now it's fixed to use always little-endian bits order.

- * Implemented NetworkMessage.

NetworkMessage class is realised in POT by OTS_Buffer class. It contains streamed reading and writing methods. Thought it is not fully same that C++ NetworkMessage class. Mainly because any encryption, decryption and header-connected routines were moved to connection handling classes.

Appendix D - Todo List

In Package POT

In [OTS.php](#)

- 1.0.0: Complete phpUnit test.
- 1.0.0: Deprecations cleanup.
- 1.0.0: E_* classes into *Exception, IOTS* into *Interface, change POT classes prefix from OTS_* into OT_*.
- 1.0.0: Implement OutOfBoundsException instead of mixed results types.
- 1.0.0: Main POT class as database instance.
- 1.0.0: More detailed documentation and tutorials, also update examples and tutorials.
- 1.0.0: PHAR and PHK packages.
- 1.0.0: Unify *List and *_List naming (probably into *List).
- 2.0.0: Code as C++ extension (as an alternative to pure PHP library which of course would still be available).
- ?: Implement POT namespace when it will be supported by PHP.

In [OTS_OTBMFile.php](#)

- 1.0.0: Complete OTBM support: link tiles with items and spawns.
- 1.0.0: Spawns support.

Index

A

About OTServ AAC scripts	21
Account number hack	18

C

constructor OTS_Row_DAO::__construct()	401
<i>Handles automatic loading for record.</i>	
constructor OTS_RSA::__construct()	404
<i>Initializes new encryption session.</i>	
constructor OTS_MonstersList::__construct()	322
<i>Loads monsters mapping file.</i>	
constructor OTS_MapCoords::__construct()	311
<i>Sets coords for point.</i>	
constructor OTS_ItemType::__construct()	299
<i>Initializes new item type object.</i>	
constructor OTS_Spell::__construct()	407
<i>Sets spell info.</i>	
constructor OTS_SpellsList::__construct()	418
<i>Loads spells list.</i>	
compat.php	486
<i>POT compatibility assurance package.</i>	
CHANGELOG	498
constructor OTS_XTEA::__construct()	438
<i>Initializes new encryption session.</i>	
constructor OTS_VocationsList::__construct()	433
<i>Loads vocations list.</i>	
constructor OTS_SQLField::__construct()	422
<i>Creates new field representation.</i>	
constructor OTS_Item::__construct()	264
<i>Creates item of given ID.</i>	
constructor OTS_HousesList::__construct()	250
<i>Loads houses information.</i>	
constructor OTS_Base_DAO::__construct()	146
<i>Sets database connection handler.</i>	
constructor OTS_Base_List::__construct()	149
<i>Sets database connection handler.</i>	
constructor OTS_Admin::__construct()	138
<i>Creates new connection to OTServ administration backend.</i>	
constructor IOTS_GuildAction::__construct()	105
<i>Objects are initialized with a guild that they are assigned to.</i>	
constructor IOTS_DB::__construct()	95
<i>Connection parameters.</i>	
constructor OTS_Buffer::__construct()	160

<i>Initializes new buffered reader.</i>	
constructor OTS_DB_MySQL::construct()	174
<i>Creates database connection.</i>	
constructor OTS_House::construct()	242
<i>Creates wrapper for given house element.</i>	
constructor OTS_DB_SQLite::construct()	184
<i>Creates database connection.</i>	
constructor OTS_DB_PostgreSQL::construct()	181
<i>Creates database connection.</i>	
constructor OTS_DB_ODBC::construct()	177
<i>Creates database connection.</i>	
constructor E_OTS_ErrorCode::construct()	84
<i>Sets error code.</i>	

D

DAO objects	9
-----------------------------	---

E

E_OTS_Generic::CONNECT_NO_DRIVER	88
<i>No database driver specified.</i>	
E_OTS_Generic::CREATE_ACCOUNT_IMPOSSIBLE	89
<i>No free account numbers to create account.</i>	
E_OTS_Generic::CONNECT_INVALID_DRIVER	88
<i>Invalid database driver.</i>	
E_OTS_Generic	88
<i>Generic exception class for general exceptions.</i>	
E_OTS_FileLoaderError::ERROR_SEEK_ERROR	87
<i>Failed to seek in given position in file.</i>	
E_OTS_FileLoaderError::ERROR_TELL_ERROR	87
<i>Failed to read position in file.</i>	
E_OTS_NoDriver	89
<i>Occurs when code attempts to execute driven action that has no assigned driver to handle it.</i>	
E_OTS_NotAContainer	90
<i>Occurs when in database item which is not a container contains sub-items.</i>	
E_OTS_OutOfBuffer	92
<i>Occurs when properties stream has ended and there is still read attempt.</i>	
E_OTS_ReadOnly	92
<i>Occurs when code attempts to evaluate write operation on read-only object.</i>	
E_OTS_OTBMErrors::LOADMAPERROR_UNKNOWNNODETYPE	91
<i>Unknown node type.</i>	
E_OTS_OTBMErrors::LOADMAPERROR_OUTDATEDHEADER	91
<i>Unsupported file version.</i>	
E_OTS_NotLoaded	90
<i>Occurs when code attempts to access property of not loaded object.</i>	
E_OTS_OTBMErrors	91
<i>OTBM map loading error.</i>	
E_OTS_FileLoaderError::ERROR_NOT_OPEN	86
<i>Attempted to execute operation on not opened file.</i>	
E_OTS_FileLoaderError::ERROR_INVALID_FORMAT	86

<i>File corrupted.</i>	
E_OTS_NotLoaded.php	30
E_OTS_OTBMErrror.php	31
<i>Code in this file bases on oryiginal OTServ OTBM format loading C++ code (iomapotbm.</i>	
E_OTS_NotAContainer.php	29
E_OTS_NoDriver.php	28
E_OTS_FileLoaderError.php	26
<i>Code in this file bases on oryiginal OTServ binary format loading C++ code (fileloader.</i>	
E_OTS_Generic.php	27
E_OTS_OutOfBuffer.php	32
<i>Code in this file bases on oryiginal OTServ binary format loading C++ code (fileloader.</i>	
E_OTS_ReadOnly.php	33
E_OTS_FileLoaderError::ERROR_EOF	85
<i>Unexpected end of file.</i>	
E_OTS_FileLoaderError::ERROR_INVALID_FILE_VERSION	86
<i>Unsupported file version.</i>	
E_OTS_FileLoaderError::ERROR_CAN_NOT_OPEN	85
<i>Could not open file.</i>	
E_OTS_FileLoaderError	85
<i>Error during reading OTServ binary file.</i>	
E_OTS_ErrorCode	84
<i>Generic exception class for error codes.</i>	
E_OTS_ErrorCode.php	25

G

Guild action drivers	15
Guilds	13

I

IOTS_Display::displayPlayer()	102
<i>Displays player.</i>	
IOTS_Display::displayPlayersList()	103
<i>Displays players list.</i>	
IOTS_FileCache	103
<i>This interface describes binary files cache control drivers.</i>	
IOTS_FileCache::readCache()	103
<i>Returns cache.</i>	
IOTS_Display::displayGuildsList()	102
<i>Displays guilds list.</i>	
IOTS_Display::displayGuildRanksList()	101
<i>Displays guild ranks list.</i>	
IOTS_Display::displayGroupsList()	100
<i>Displays groups list.</i>	
IOTS_Display::displayGuild()	100
<i>Displays guild.</i>	
IOTS_Display::displayGuildRank()	101
<i>Displays rank.</i>	
IOTS_FileCache::writeCache()	104
<i>Writes node cache.</i>	

IOTS_GuildAction	105
<i>Guild action interface.</i>	
IOTS_ItemsCache::readItems()	108
<i>Returns cache.</i>	
IOTS_ItemsCache::writeItems()	108
<i>Writes items cache.</i>	
INSTALL	503
IOTS_ItemsCache	107
<i>This interface defines items.</i>	
IOTS_GuildAction::submitRequest()	107
<i>Finalizes request.</i>	
IOTS_GuildAction::addRequest()	106
<i>Adds new request.</i>	
IOTS_GuildAction::deleteRequest()	106
<i>Deletes request.</i>	
IOTS_GuildAction::listRequests()	106
<i>List of saved pending actions.</i>	
IOTS_Display::displayGroup()	99
<i>Displays group.</i>	
IOTS_Display::displayAccountsList()	99
<i>Displays accounts list.</i>	
IOTS_ItemsCache.php	40
IOTS_Cipher	93
<i>Cypher encryption/decryption class interface.</i>	
IOTS_Cipher::decrypt()	93
<i>Decrypts encrypted message.</i>	
IOTS_Cipher::encrypt()	93
<i>Encrypts message.</i>	
IOTS_GuildAction.php	39
IOTS_FileCache.php	38
IOTS_DAO.php	35
IOTS_DB.php	36
IOTS_Display.php	37
IOTS_DAO	94
<i>OTServ database object.</i>	
IOTS_DB	94
<i>OTServ database handler interface.</i>	
IOTS_DB::tableName()	97
<i>Query-quoted table name.</i>	
IOTS_Display	98
<i>Display interface.</i>	
IOTS_Display::displayAccount()	98
<i>Displays account.</i>	
IOTS_DB::SQLquote()	97
<i>Query-quoted string value.</i>	
IOTS_DB::SQLquery()	97
<i>Evaluates query.</i>	
IOTS_DB::fieldName()	95
<i>Query-quoted field name.</i>	
IOTS_DB::lastInsertId()	96
<i>ID of last created record.</i>	
IOTS_DB::limit()	96
<i>LIMIT/OFFSET clause for queries.</i>	

IOTS_Cipher.php	34
---	----

L

LICENSE	496
-----------------------------------	-----

N

NEWS	503
--------------------------------	-----

O

OTS_Monster::toString()	321
<i>Returns string representation of XML.</i>	
OTS_Monster::get()	320
<i>Magic PHP5 method.</i>	
OTS_MonstersList	321
<i>Wrapper for monsters list.</i>	
OTS_MonstersList::count()	322
<i>Returns amount of monsters loaded.</i>	
OTS_MonstersList::current()	322
<i>Returns monster at current position in iterator.</i>	
OTS_Monster::hasImmunity()	320
<i>Checks if monster has given immunity.</i>	
OTS_Monster::getVoices()	319
<i>Returns voices that monster can sound.</i>	
OTS_Monster::getManaCost()	318
<i>Returns amount of mana required to summon this monster.</i>	
OTS_Monster::getName()	318
<i>Returns monster name.</i>	
OTS_Monster::getRace()	319
<i>Returns monster race.</i>	
OTS_Monster::getSpeed()	319
<i>Returns monster speed.</i>	
OTS_MonstersList::getMonster()	323
<i>Returns loaded data of given monster.</i>	
OTS_MonstersList::key()	323
<i>Returns name of current position.</i>	
OTS_MonstersList::set_state()	326
<i>Magic PHP5 method.</i>	
OTS_MonstersList::valid()	326
<i>Checks if there is anything more in interator.</i>	
OTS_OTBMFile	327
<i>OTBM format reader.</i>	
OTS_OTBMFile::OTBM_ATTR_ACTION_ID	328
<i>Action ID.</i>	
OTS_OTBMFile::OTBM_ATTR_DEPOT_ID	328
<i>Depot ID.</i>	
OTS_MonstersList::rewind()	326
<i>Resets iterator index.</i>	

OTS MonstersList::offsetUnset()	325
<i>This method is implemented for ArrayAccess interface.</i>	
OTS MonstersList::next()	324
<i>Moves to next iterator monster.</i>	
OTS MonstersList::offsetExists()	324
<i>Checks if given element exists.</i>	
OTS MonstersList::offsetGet()	324
<i>Returns item from given position.</i>	
OTS MonstersList::offsetSet()	325
<i>This method is implemented for ArrayAccess interface.</i>	
OTS Monster::getLoot()	318
<i>Returns all possible loot.</i>	
OTS Monster::getItems()	317
<i>Returns all possible loot.</i>	
OTS MapCoords	310
<i>Map position point.</i>	
OTS ItemType:: set_state()	310
<i>Magic PHP5 method.</i>	
OTS MapCoords::getX()	311
<i>Returns X.</i>	
OTS MapCoords::getY()	312
<i>Returns Y.</i>	
OTS MapCoords::getZ()	312
<i>Returns Z.</i>	
OTS ItemType:: set()	309
<i>Magic PHP5 method.</i>	
OTS ItemType:: get()	309
<i>Magic PHP5 method.</i>	
OTS ItemType::setFlags()	307
<i>Sets type flags.</i>	
OTS ItemType::setGroup()	307
<i>Sets item group.</i>	
OTS ItemType::setName()	308
<i>Sets item type name.</i>	
OTS ItemType::setType()	308
<i>Sets item type.</i>	
OTS MapCoords:: get()	312
<i>Magic PHP5 method.</i>	
OTS MapCoords:: set_state()	313
<i>Magic PHP5 method.</i>	
OTS Monster::getFlag()	316
<i>Returns specified flag value.</i>	
OTS Monster::getFlags()	316
<i>Returns all monster flags (in format flagname => value).</i>	
OTS Monster::getHealth()	317
<i>Returns monster HP.</i>	
OTS Monster::getImmunities()	317
<i>Returns all monster immunities.</i>	
OTS Monster::getExperience()	316
<i>Returns amount of experience for killing this monster.</i>	
OTS Monster::getDefenses()	315
<i>Returns list of special defenses.</i>	
OTS Monster	313

<i>Wrapper for monsters files DOMDocument.</i>	
OTS_Monster::getArmor()	314
<i>Returns monster armor.</i>	
OTS_Monster::getAttacks()	315
<i>Returns list of monster attacks.</i>	
OTS_Monster::getDefense()	315
<i>Returns monster defense rate.</i>	
OTS_OTBMFile::OTBM_ATTR_DESC	328
<i>Description.</i>	
OTS_OTBMFile::OTBM_ATTR_DESCRIPTION	329
<i>Description attribute.</i>	
OTS_OTBMFile::getTownsList()	340
<i>Returns list (id => name) of loaded towns.</i>	
OTS_OTBMFile::getTownName()	340
<i>Returns name of given town's ID.</i>	
OTS_OTBMFile::getTownTemple()	341
<i>Returns town's temple position.</i>	
OTS_OTBMFile::getWidth()	341
<i>Returns map width.</i>	
OTS_OTBMFile::key()	341
<i>Returns ID of current position.</i>	
OTS_OTBMFile::getTownID()	339
<i>Returns town's ID.</i>	
OTS_OTBMFile::getIterator()	339
<i>Returns iterator handle for loops.</i>	
OTS_OTBMFile::current()	337
<i>Returns town at current position in iterator.</i>	
OTS_OTBMFile::getDescription()	338
<i>Returns map description.</i>	
OTS_OTBMFile::getHeight()	338
<i>Returns map height.</i>	
OTS_OTBMFile::getHousesList()	339
<i>Loads map's houses list.</i>	
OTS_OTBMFile::loadFile()	342
<i>Loads OTBM file content.</i>	
OTS_OTBMFile::next()	342
<i>Moves to next iterator town.</i>	
OTS_OTBMFile::__get()	346
<i>Magic PHP5 method.</i>	
OTS_OTBMFile::__set_state()	346
<i>Magic PHP5 method.</i>	
OTS_OTBMFile::__wakeup()	347
<i>Magic PHP5 method.</i>	
OTS_Player	347
<i>OTServ character abstraction.</i>	
OTS_OTBMFile::valid()	345
<i>Checks if there is anything more in interator.</i>	
OTS_OTBMFile::rewind()	345
<i>Resets iterator index.</i>	
OTS_OTBMFile::offsetExists()	343
<i>Checks if given element exists.</i>	
OTS_OTBMFile::offsetGet()	343
<i>Returns item from given position.</i>	

OTS OTBMFile::offsetSet()	344
<i>This method is implemented for ArrayAccess interface.</i>	
OTS OTBMFile::offsetUnset()	344
<i>This method is implemented for ArrayAccess interface.</i>	
OTS OTBMFile::count()	337
<i>Returns amount of towns loaded.</i>	
OTS OTBMFile::OTBM_NODE_TOWNS	337
<i>Towns container.</i>	
OTS OTBMFile::OTBM_ATTR_TEXT	331
<i>Text.</i>	
OTS OTBMFile::OTBM_ATTR_TELE_DEST	331
<i>Teleport destination.</i>	
OTS OTBMFile::OTBM_ATTR_TILE_FLAGS	332
<i>Tile flags.</i>	
OTS OTBMFile::OTBM_ATTR_UNIQUE_ID	332
<i>Unique ID.</i>	
OTS OTBMFile::OTBM_NODE_HOUSED_TILE	332
<i>Tile of house.</i>	
OTS OTBMFile::OTBM_ATTR_RUNE_CHARGES	331
<i>Rune charges amount.</i>	
OTS OTBMFile::OTBM_ATTR_ITEM	330
<i>Item.</i>	
OTS OTBMFile::OTBM_ATTR_EXT_FILE	329
<i>External file.</i>	
OTS OTBMFile::OTBM_ATTR_EXT_HOUSE_FILE	329
<i>External houses file.</i>	
OTS OTBMFile::OTBM_ATTR_EXT_SPAWN_FILE	330
<i>External spawns file.</i>	
OTS OTBMFile::OTBM_ATTR_HOUSED_OORID	330
<i>ID of doors.</i>	
OTS OTBMFile::OTBM_NODE_ITEM	333
<i>Item.</i>	
OTS OTBMFile::OTBM_NODE_ITEM_DEF	333
<i>Item definition.</i>	
OTS OTBMFile::OTBM_NODE_TILE_AREA	335
<i>Map tiles fragment.</i>	
OTS OTBMFile::OTBM_NODE_TILE_REF	336
<i>Tile reference.</i>	
OTS OTBMFile::OTBM_NODE_TILE_SQUARE	336
<i>Tile.</i>	
OTS OTBMFile::OTBM_NODE_TOWN	336
<i>Town.</i>	
OTS OTBMFile::OTBM_NODE_TILE	335
<i>Single tile.</i>	
OTS OTBMFile::OTBM_NODE_SPAWN_AREA	335
<i>Spawn.</i>	
OTS OTBMFile::OTBM_NODE_MAP_DATA	333
<i>Map data container.</i>	
OTS OTBMFile::OTBM_NODE_MONSTER	334
<i>Monster.</i>	
OTS OTBMFile::OTBM_NODE_ROOTV1	334
<i>Root node.</i>	
OTS OTBMFile::OTBM_NODE_SPAWNS	334

<i>Spawns container.</i>	
OTS_ItemType::setClientId()	306
<i>Sets client side ID.</i>	
OTS_ItemType::setAttribute()	306
<i>Sets given attribute.</i>	
OTS_ItemsList::wakeup()	282
<i>Magic PHP5 method.</i>	
OTS_ItemsList::set_state()	282
<i>Magic PHP5 method.</i>	
OTS_ItemType	283
<i>Item type info.</i>	
OTS_ItemType::FLAG_ALLOWDISTREAD	284
<i>Can be read from distance.</i>	
OTS_ItemType::FLAG_ALWAYSONTOP	284
<i>Is always over other items in stack.</i>	
OTS_ItemsList::get()	281
<i>Magic PHP5 method.</i>	
OTS_ItemsList::valid()	281
<i>Checks if there is anything more in iterator.</i>	
OTS_ItemsList::offsetGet()	279
<i>Returns item from given position.</i>	
OTS_ItemsList::offsetSet()	279
<i>This method is implemented for ArrayAccess interface.</i>	
OTS_ItemsList::offsetUnset()	280
<i>This method is implemented for ArrayAccess interface.</i>	
OTS_ItemsList::rewind()	280
<i>Resets iterator index.</i>	
OTS_ItemType::FLAG_BLOCK_PATHFIND	284
<i>Can block searching for path.</i>	
OTS_ItemType::FLAG_BLOCK_PROJECTILE	285
<i>BLOCK_PROJECTILE flag(?).</i>	
OTS_ItemType::FLAG_FLOORCHANGEWEST	287
<i>Changes floor west from it's position.</i>	
OTS_ItemType::FLAG_HANGABLE	287
<i>Can be hang(?).</i>	
OTS_ItemType::FLAG_HAS_HEIGHT	288
<i>Does item rises stack height on it's field.</i>	
OTS_ItemType::FLAG_HORIZONTAL	288
<i>Is oriented horizontally.</i>	
OTS_ItemType::FLAG_FLOORCHANGESOUTH	287
<i>Changes floor south from it's position.</i>	
OTS_ItemType::FLAG_FLOORCHANGENORTH	286
<i>Changes floor north from it's position.</i>	
OTS_ItemType::FLAG_BLOCK_SOLID	285
<i>Can block characters from walking.</i>	
OTS_ItemType::FLAG_CANNOTDECAY	285
<i>Doesn't decay.</i>	
OTS_ItemType::FLAG_FLOORCHANGEDOWN	286
<i>Changes floor under it.</i>	
OTS_ItemType::FLAG_FLOORCHANGEEST	286
<i>Changes floor east from it's position.</i>	
OTS_ItemsList::offsetExists()	278
<i>Checks if given element exists.</i>	

OTS ItemsList::next()	278
<i>Moves to next iterator item.</i>	
OTS ItemsList::ITEM_ATTR_LIGHT2	272
<i>Light.</i>	
OTS ItemsList::ITEM_ATTR_CLIENTID	272
<i>Client ID.</i>	
OTS ItemsList::ITEM_ATTR_SERVERID	272
<i>Server ID.</i>	
OTS ItemsList::ITEM_ATTR_SPEED	273
<i>Speed.</i>	
OTS ItemsList::ITEM_ATTR_TOPORDER	273
<i>Always-on-top order.</i>	
OTS ItemsList::CLIENT_VERSION_800	271
<i>Tibia client 8.</i>	
OTS ItemsList::CLIENT_VERSION_792	271
<i>Tibia client 7.</i>	
OTS ItemsList::CLIENT_VERSION_760	270
<i>Tibia client 7.</i>	
OTS ItemsList::CLIENT_VERSION_770	270
<i>Tibia client 7.</i>	
OTS ItemsList::CLIENT_VERSION_780	270
<i>Tibia client 7.</i>	
OTS ItemsList::CLIENT_VERSION_790	271
<i>Tibia client 7.</i>	
OTS ItemsList::ROOT_ATTR_VERSION	273
<i>Root file attribute.</i>	
OTS ItemsList::count()	274
<i>Returns amount of items loaded.</i>	
OTS ItemsList::getIterator()	276
<i>Returns iterator handle for loops.</i>	
OTS ItemsList::getOTBVersion()	277
<i>Returns OTB file version.</i>	
OTS ItemsList::key()	277
<i>Returns ID of current position.</i>	
OTS ItemsList::loadItems()	277
<i>Loads items.</i>	
OTS ItemsList::getItemTypesList()	276
<i>Returns all loaded items.</i>	
OTS ItemsList::getItemTypeid()	275
<i>Finds item type by it's name.</i>	
OTS ItemsList::current()	274
<i>Returns item at current position in iterator.</i>	
OTS ItemsList::getBuildVersion()	274
<i>Returns build version.</i>	
OTS ItemsList::getClientVersion()	275
<i>Returns client version.</i>	
OTS ItemsList::getItemType()	275
<i>Returns given item type.</i>	
OTS ItemType::FLAG_MOVEABLE	288
<i>Can be moved by player.</i>	
OTS ItemType::FLAG_PICKUPABLE	289
<i>Can be picked up by player.</i>	
OTS ItemType::getGroup()	301

<i>Returns group.</i>	
OTS_ItemType::getClientId()	300
<i>Returns item type client ID.</i>	
OTS_ItemType::getId()	301
<i>Returns item type server ID.</i>	
OTS_ItemType::getName()	301
<i>Returns item name.</i>	
OTS_ItemType::getType()	302
<i>Returns item type.</i>	
OTS_ItemType::getAttributesList()	300
<i>Returns all attributes list.</i>	
OTS_ItemType::getAttribute()	299
<i>Returns given attribute.</i>	
OTS_ItemType::ITEM_TYPE_NONE	298
<i>No special type.</i>	
OTS_ItemType::ITEM_TYPE_TELEPORT	298
<i>Teleport.</i>	
OTS_ItemType::ITEM_TYPE_TRASHHOLDER	298
<i>Trash can.</i>	
OTS_ItemType::createItem()	299
<i>Creates instance of this type.</i>	
OTS_ItemType::hasHeight()	302
<i>Checks if item has height.</i>	
OTS_ItemType::isAlwaysOnTop()	302
<i>Checks if item is always on top.</i>	
OTS_ItemType::isRotable()	304
<i>Checks if item can be rotated.</i>	
OTS_ItemType::isStackable()	305
<i>Checks if item is stackable.</i>	
OTS_ItemType::isUsable()	305
<i>Checks if item is usable.</i>	
OTS_ItemType::isVertical()	305
<i>Checks if item is vertical.</i>	
OTS_ItemType::isReadable()	304
<i>Checks if item is readable.</i>	
OTS_ItemType::isPickupable()	304
<i>Checks if item is pickupable.</i>	
OTS_ItemType::isBlocking()	303
<i>Checks if item is blocking.</i>	
OTS_ItemType::isHangable()	303
<i>Checks if item can be hanged.</i>	
OTS_ItemType::isHorizontal()	303
<i>Checks if item is horizontal.</i>	
OTS_ItemType::isMovable()	303
<i>Checks if item is movable.</i>	
OTS_ItemType::ITEM_TYPE_MAILBOX	297
<i>Mailbox.</i>	
OTS_ItemType::ITEM_TYPE_MAGICFIELD	297
<i>Magic field.</i>	
OTS_ItemType::ITEM_GROUP_CONTAINER	291
<i>Container.</i>	
OTS_ItemType::ITEM_GROUP_ARMOR	291
<i>Armor.</i>	

OTS ItemType::ITEM GROUP DEPRECATED	292
<i>Deprecated item.</i>	
OTS ItemType::ITEM GROUP DOOR	292
<i>Door.</i>	
OTS ItemType::ITEM GROUP FLUID	293
<i>Liquid thing.</i>	
OTS ItemType::ITEM GROUP AMMUNITION	291
<i>Ammunition.</i>	
OTS ItemType::FLAG VERTICAL	290
<i>Is oriented vertically.</i>	
OTS ItemType::FLAG READABLE	289
<i>Has readable sign.</i>	
OTS ItemType::FLAG ROTABLE	289
<i>Can be rotated by player.</i>	
OTS ItemType::FLAG STACKABLE	290
<i>Can be grouped with another items.</i>	
OTS ItemType::FLAG USEABLE	290
<i>Can be used by players.</i>	
OTS ItemType::ITEM GROUP GROUND	293
<i>Ground tile.</i>	
OTS ItemType::ITEM GROUP KEY	293
<i>Key.</i>	
OTS ItemType::ITEM GROUP WRITEABLE	296
<i>Item that can store editable sign.</i>	
OTS ItemType::ITEM TYPE CONTAINER	296
<i>Container.</i>	
OTS ItemType::ITEM TYPE DEPOT	296
<i>Depot locker.</i>	
OTS ItemType::ITEM TYPE DOOR	297
<i>Door.</i>	
OTS ItemType::ITEM GROUP WEAPON	295
<i>Weapon.</i>	
OTS ItemType::ITEM GROUP TELEPORT	295
<i>Teleport field.</i>	
OTS ItemType::ITEM GROUP MAGICFIELD	294
<i>Magic field.</i>	
OTS ItemType::ITEM GROUP NONE	294
<i>No group speciffied.</i>	
OTS ItemType::ITEM GROUP RUNE	294
<i>Rune.</i>	
OTS ItemType::ITEM GROUP SPLASH	295
<i>Splash effect.</i>	
OTS Player::ban()	349
<i>Bans current player.</i>	
OTS Player::delete()	350
<i>Deletes player.</i>	
OTS Spell::getName()	410
<i>Returns spell name.</i>	
OTS Spell::getMana()	410
<i>Mana cost.</i>	
OTS Spell::getReagent()	411
<i>Returns item type of reagent item.</i>	
OTS Spell::getReagentId()	411

<i>Returns ID of item that is used by spell.</i>	
OTS_Spell::getSoul()	411
<i>Soul points cost.</i>	
OTS_Spell::getMagicLevel()	410
<i>Magic level required to cast.</i>	
OTS_Spell::getLevel()	409
<i>Level required for use.</i>	
OTS_Spell::getConjure()	408
<i>Returns item type of conjured item.</i>	
OTS_Spell::getConjureCount()	408
<i>Returns amount of items conjured by this spell.</i>	
OTS_Spell::getConjureId()	409
<i>Returns ID of item conjured by this spell.</i>	
OTS_Spell::getId()	409
<i>Returns rune item id.</i>	
OTS_Spell::getType()	412
<i>Returns spell type.</i>	
OTS_Spell::getVocations()	412
<i>Returns list of vocations that are allowed to learn this spell.</i>	
OTS_Spell::isLearnNeeded()	414
<i>Checks if spell needs to be learned.</i>	
OTS_Spell::isPremium()	415
<i>Checks if spell requires PACC.</i>	
OTS_Spell::get()	415
<i>Magic PHP5 method.</i>	
OTS_Spell::toString()	416
<i>Returns string representation of XML.</i>	
OTS_Spell::isFarUseAllowed()	414
<i>Checks if distance use allowed.</i>	
OTS_Spell::isEnabled()	414
<i>Checks if spell is enabled.</i>	
OTS_Spell::getWords()	412
<i>Returns spell formula.</i>	
OTS_Spell::hasParams()	413
<i>Checks if spell has parameter.</i>	
OTS_Spell::isAggressive()	413
<i>This method is the same as OTS_Spell::isAggressive().</i>	
OTS_Spell::isAggressive()	413
<i>Checks if spell is threatened as unfriendly by other creatures.</i>	
OTS_Spell::getCharges()	408
<i>Number of rune charges.</i>	
OTS_Spell::createConjure()	407
<i>Creates conjure item.</i>	
OTS_Player::sleep()	398
<i>Magic PHP5 method.</i>	
OTS_Player::set()	397
<i>Magic PHP5 method.</i>	
OTS_Player::toString()	398
<i>Returns string representation of object.</i>	
OTS_Players_List	399
<i>List of players.</i>	
OTS_Players_List::deletePlayer()	399
<i>Deletes player.</i>	

OTS_Player::__get()	397
<i>Magic PHP5 method.</i>	
OTS_Player::__unsetSave()	396
<i>Unsets save flag.</i>	
OTS_Player::__setTownId()	395
<i>Sets residence town's ID.</i>	
OTS_Player::__setVocation()	395
<i>Sets player proffesion.</i>	
OTS_Player::unban()	396
<i>Deletes ban from current player.</i>	
OTS_Player::__unsetRedSkull()	396
<i>Unsets red skull flag.</i>	
OTS_Players_List::init()	400
<i>Sets list parameters.</i>	
OTS_Players_List::__toString()	400
<i>Returns string representation of object.</i>	
OTS_RSA::Q	404
<i>OTServ key part.</i>	
OTS_RSA::decrypt()	405
<i>Decrypts RSA-encrypted message.</i>	
OTS_RSA::encrypt()	405
<i>Encnrypts message with RSA algorithm.</i>	
OTS_Spell	406
<i>Wrapper for spell info.</i>	
OTS_RSA::P	403
<i>OTServ key part.</i>	
OTS_RSA::D	403
<i>OTServ key part.</i>	
OTS_Row_DAO	401
<i>Base class for all single-row classes.</i>	
OTS_Row_DAO::find()	401
<i>Loads row by it's name.</i>	
OTS_Row_DAO::load()	402
<i>Loads row by it's ID.</i>	
OTS_RSA	402
<i>RSA encryption/decryption mechanism.</i>	
OTS_SpellsList	416
<i>Wrapper for spells list.</i>	
OTS_SpellsList::SPELL_CONJURE	417
<i>Conjure spell.</i>	
OTS_Toolbox	431
<i>Toolbox for common operations.</i>	
OTS_SQLFilter::__toString()	430
<i>Returns string representation of WHERE clause.</i>	
OTS_Toolbox::experienceForLevel()	431
<i>Calculates experience points needed for given level.</i>	
OTS_Toolbox::levelForExperience()	432
<i>Finds out which level user have basing on his/her experience.</i>	
OTS_VocationsList	432
<i>Wrapper for vocations.</i>	
OTS_SQLFilter::__sleep()	430
<i>Magic PHP5 method.</i>	
OTS_SQLFilter::__get()	430

<i>Magic PHP5 method.</i>	428
OTS SQLFilter::OPERATOR_NLOWER	428
<i>Not-lower-then operator.</i>	
OTS SQLFilter::addFilter()	428
<i>General-purpose filter.</i>	
OTS SQLFilter::compareField()	429
<i>Compares field with a literal value.</i>	
OTS SQLFilter::getTables()	429
<i>Returns list of all tables used by filter.</i>	
OTS VocationsList::count()	433
<i>Returns amount of vocations loaded.</i>	
OTS VocationsList::getIterator()	434
<i>Returns iterator handle for loops.</i>	
OTS VocationsList:: set_state()	437
<i>Magic PHP5 method.</i>	
OTS XTEA	437
<i>XTEA encryption/decryption mechanism.</i>	
OTS XTEA::decrypt()	438
<i>Decrypt XTEA-encrypted string.</i>	
OTS XTEA::encrypt()	439
<i>Encrypt a string with XTEA algorithm.</i>	
OTS VocationsList::offsetUnset()	436
<i>This method is implemented for ArrayAccess interface.</i>	
OTS VocationsList::offsetSet()	436
<i>This method is implemented for ArrayAccess interface.</i>	
OTS VocationsList::getVocationId()	434
<i>Returns vocation's ID.</i>	
OTS VocationsList::getVocationName()	434
<i>Returns name of given vocation's ID.</i>	
OTS VocationsList::offsetExists()	435
<i>Checks if given element exists.</i>	
OTS VocationsList::offsetGet()	435
<i>Returns item from given position.</i>	
OTS SQLFilter::OPERATOR_NLIKE	427
<i>Not-LIKE operator.</i>	
OTS SQLFilter::OPERATOR_NGREATER	427
<i>Not-greater-then operator.</i>	
OTS SpellsList::getRunesList()	420
<i>Returns list of runes.</i>	
OTS SpellsList::getRune()	420
<i>Returns given rune spell.</i>	
OTS SpellsList:: get()	420
<i>Magic PHP5 method.</i>	
OTS SpellsList:: set_state()	421
<i>Magic PHP5 method.</i>	
OTS SQLField	421
<i>SQL identifier representation.</i>	
OTS SpellsList::getInstantsList()	419
<i>Returns list of instants.</i>	
OTS SpellsList::getInstant()	419
<i>Returns given instant spell.</i>	
OTS SpellsList::SPELL_INSTANT	417
<i>Instant spell.</i>	

OTS SpellsList::SPELL RUNE	417
<i>Rune spell.</i>	
OTS SpellsList::getConjure()	418
<i>Returns given conjure spell.</i>	
OTS SpellsList::getConjuresList()	419
<i>Returns list of conjure spells.</i>	
OTS SQLField::getName()	422
<i>Returns field name.</i>	
OTS SQLField::getTable()	423
<i>Returns table name.</i>	
OTS SQLFilter::OPERATOR GREATER	426
<i>Greater-than operator.</i>	
OTS SQLFilter::OPERATOR LIKE	426
<i>LIKE operator.</i>	
OTS SQLFilter::OPERATOR LOWER	426
<i>Lower-than operator.</i>	
OTS SQLFilter::OPERATOR NEQUAL	427
<i>Not-equal operator.</i>	
OTS SQLFilter::OPERATOR EQUAL	425
<i>Equal operator.</i>	
OTS SQLFilter::CRITERIUM OR	425
<i>OR sibling.</i>	
OTS SQLField:: get()	423
<i>Magic PHP5 method.</i>	
OTS SQLField:: toString()	424
<i>Returns string representation of WHERE clause.</i>	
OTS SQLFilter	424
<i>SQL WHERE clause object.</i>	
OTS SQLFilter::CRITERIUM AND	425
<i>AND sibling.</i>	
OTS Player::setStorage()	394
<i>Sets value of storage record.</i>	
OTS Player::setSoul()	394
<i>Sets soul points.</i>	
OTS Player::getPosX()	363
<i>X map coordinate.</i>	
OTS Player::getName()	363
<i>Player name.</i>	
OTS Player::getPosY()	364
<i>Y map coordinate.</i>	
OTS Player::getPosZ()	364
<i>Z map coordinate.</i>	
OTS Player::getPremiumEnd()	365
<i>Player's Premium Account expiration timestamp.</i>	
OTS Player::getManaSpent()	363
<i>Mana spent.</i>	
OTS Player::getManaMax()	362
<i>Maximum mana.</i>	
OTS Player::getLossMana()	361
<i>Percentage of used mana lost after dead.</i>	
OTS Player::getLossSkills()	361
<i>Percentage of skills lost after dead.</i>	
OTS Player::getMagLevel()	361

<i>Magic level.</i>	
OTS_Player::getMana()	362
<i>Current mana.</i>	
OTS_Player::getRank()	365
<i>Assigned guild rank.</i>	
OTS_Player::getRankId()	365
<i>Guild rank ID.</i>	
OTS_Player::getSoul()	369
<i>Soul points.</i>	
OTS_Player::getStorage()	369
<i>Returns value of storage record.</i>	
OTS_Player::getTownId()	370
<i>Residence town's ID.</i>	
OTS_Player::getTownName()	370
<i>Player residence town name.</i>	
OTS_Player::getSlot()	368
<i>Returns items tree from given slot.</i>	
OTS_Player::getSkillTries()	368
<i>Returns player's skill's tries for next level.</i>	
OTS_Player::getRedSkullTime()	366
<i>Red skulled time remained.</i>	
OTS_Player::getSave()	366
<i>Save counter.</i>	
OTS_Player::getSex()	367
<i>Player gender.</i>	
OTS_Player::getSkill()	367
<i>Returns player's skill.</i>	
OTS_Player::getLossExperience()	360
<i>Percentage of experience lost after dead.</i>	
OTS_Player::getLookType()	360
<i>Outfit.</i>	
OTS_Player::getDirection()	354
<i>Looking direction.</i>	
OTS_Player::getDepot()	353
<i>Returns items tree from given depot.</i>	
OTS_Player::getExperience()	354
<i>Experience points.</i>	
OTS_Player::getGroup()	354
<i>Returns group of this player.</i>	
OTS_Player::getGuildNick()	355
<i>Guild nick.</i>	
OTS_Player::getCustomField()	352
<i>Reads custom field.</i>	
OTS_Player::getConditions()	352
<i>Conditions.</i>	
OTS_Player::find()	350
<i>Loads player by it's name.</i>	
OTS_Player::getAccount()	351
<i>Returns account of this player.</i>	
OTS_Player::getBalance()	351
<i>Bank balance.</i>	
OTS_Player::getCap()	351
<i>Capacity.</i>	

OTS_Player::getHealth()	355
<i>Current HP.</i>	
OTS_Player::getHealthMax()	356
<i>Maximum HP.</i>	
OTS_Player::getLookBody()	358
<i>Body color.</i>	
OTS_Player::getLookFeet()	359
<i>Boots color.</i>	
OTS_Player::getLookHead()	359
<i>Hair color.</i>	
OTS_Player::getLookLegs()	359
<i>Legs color.</i>	
OTS_Player::getLookAddons()	358
<i>Addons.</i>	
OTS_Player::getLevel()	357
<i>Experience level.</i>	
OTS_Player::getHouse()	356
<i>Returns house rented by this player.</i>	
OTS_Player::getId()	356
<i>Player ID.</i>	
OTS_Player::getLastIP()	357
<i>Last login IP.</i>	
OTS_Player::getLastLogin()	357
<i>Last login timestamp.</i>	
OTS_Player::getVocation()	371
<i>Player proffesion.</i>	
OTS_Player::getVocationName()	371
<i>Player proffesion name.</i>	
OTS_Player::setManaMax()	386
<i>Sets maximum mana.</i>	
OTS_Player::setMana()	386
<i>Sets current mana.</i>	
OTS_Player::setManaSpent()	387
<i>Sets mana spent.</i>	
OTS_Player::setName()	387
<i>Sets players's name.</i>	
OTS_Player::setPosX()	387
<i>Sets X map coordinate.</i>	
OTS_Player::setMagLevel()	385
<i>Sets magic level.</i>	
OTS_Player::setLossSkills()	385
<i>Sets percentage of skills lost after dead.</i>	
OTS_Player::setLookLegs()	383
<i>Sets legs color.</i>	
OTS_Player::setLookType()	383
<i>Sets outfit.</i>	
OTS_Player::setLossExperience()	384
<i>Sets percentage of experience lost after dead.</i>	
OTS_Player::setLossMana()	384
<i>Sets percentage of used mana lost after dead.</i>	
OTS_Player::setPosY()	388
<i>Sets Y map coordinate.</i>	
OTS_Player::setPosZ()	388

<i>Sets Z map coordinate.</i>	
OTS_Player::setSex()	392
<i>Sets player gender.</i>	
OTS_Player::setSkill()	392
<i>Sets skill value.</i>	
OTS_Player::setSkillTries()	393
<i>Sets skill's tries for next level.</i>	
OTS_Player::setSlot()	393
<i>Sets slot content.</i>	
OTS_Player::setSave()	391
<i>Sets save flag.</i>	
OTS_Player::setRedSkullTime()	391
<i>Sets red skulled time remained.</i>	
OTS_Player::setPremiumEnd()	389
<i>Sets player's Premium Account expiration timestamp.</i>	
OTS_Player::setRank()	389
<i>Assigns guild rank.</i>	
OTS_Player::setRankId()	390
<i>Sets guild rank ID.</i>	
OTS_Player::setRedSkull()	390
<i>Sets red skull flag.</i>	
OTS_Player::setLookHead()	382
<i>Sets hair color.</i>	
OTS_Player::setLookFeet()	382
<i>Sets boots color.</i>	
OTS_Player::setBalance()	374
<i>Sets bank balance value.</i>	
OTS_Player::setAccount()	374
<i>Assigns character to account.</i>	
OTS_Player::setCap()	375
<i>Sets capacity.</i>	
OTS_Player::setConditions()	375
<i>Sets conditions.</i>	
OTS_Player::setCustomField()	376
<i>Writes custom field.</i>	
OTS_Player::save()	373
<i>Saves player in database.</i>	
OTS_Player::load()	373
<i>Loads player with given id.</i>	
OTS_Player::hasRedSkull()	371
<i>Checks if player has red skull.</i>	
OTS_Player::isBanned()	372
<i>Checks if player is banned.</i>	
OTS_Player::isLoading()	372
<i>Checks if object is loaded.</i>	
OTS_Player::isSaveSet()	372
<i>Checks if save flag is set.</i>	
OTS_Player::setDepot()	376
<i>Sets depot content.</i>	
OTS_Player::setDirection()	377
<i>Sets looking direction.</i>	
OTS_Player::setLastLogin()	380
<i>Sets last login timestamp.</i>	

OTS_Player::setLevel()	381
<i>Sets experience level.</i>	
OTS_Player::setLookAddons()	381
<i>Sets addons.</i>	
OTS_Player::setLookBody()	382
<i>Sets body color.</i>	
OTS_Player::setLastIP()	380
<i>Sets last login IP.</i>	
OTS_Player::setHealthMax()	379
<i>Sets maximum HP.</i>	
OTS_Player::setExperience()	378
<i>Sets experience points.</i>	
OTS_Player::setGroup()	378
<i>Assigns character to group.</i>	
OTS_Player::setGuildNick()	378
<i>Sets guild nick.</i>	
OTS_Player::setHealth()	379
<i>Sets current HP.</i>	
OTS_ItemsList::CLIENT_VERSION_755	269
<i>Tibia client 7.</i>	
OTS_ItemsList::CLIENT_VERSION_750	269
<i>Tibia client 7.</i>	
OTS_Base_DAO::sleep()	147
<i>Magic PHP5 method.</i>	
OTS_Base_DAO::set_state()	146
<i>Magic PHP5 method.</i>	
OTS_Base_DAO::wakeup()	147
<i>Magic PHP5 method.</i>	
OTS_Base_List	148
<i>Basic list class routines.</i>	
OTS_Base_List::\$class	148
<i>Class of generated objects.</i>	
OTS_Base_DAO::clone()	146
<i>Creates clone of object.</i>	
OTS_Base_DAO::\$db	145
<i>Database connection.</i>	
OTS_Admin::set_state()	143
<i>Magic PHP5 method.</i>	
OTS_Admin::sleep()	144
<i>Magic PHP5 method.</i>	
OTS_Admin::wakeup()	144
<i>Magic PHP5 method.</i>	
OTS_Base_DAO	145
<i>Basic data access object routines.</i>	
OTS_Base_List::\$table	149
<i>Default table name for queries.</i>	
OTS_Base_List::count()	149
<i>Returns number of rows on list in current criterium.</i>	
OTS_Base_List::setFilter()	152
<i>Sets filter on list.</i>	
OTS_Base_List::rewind()	152
<i>Select rows from database.</i>	
OTS_Base_List::setLimit()	153

Sets <i>LIMIT</i> .	
OTS Base List::setOffset()	153
Sets <i>OFFSET</i> .	
OTS Base List::valid()	153
Checks if there are any rows left.	
OTS Base List::resetOrder()	151
Clears <i>ORDER BY</i> clause.	
OTS Base List::orderBy()	151
Appends sorting rule.	
OTS Base List::current()	150
Returns current row.	
OTS Base List::init()	150
Sets list parameters.	
OTS Base List::key()	150
Current cursor position.	
OTS Base List::next()	151
Moves to next row.	
OTS Admin::destruct()	143
Closes connection.	
OTS Admin::clone()	143
Magic PHP5 method.	
OTS Admin::RESPOND KEY EXCHANGE FAILED	136
Keys exchange failed.	
OTS Admin::RESPOND HELLO	136
Hello respond.	
OTS Admin::RESPOND KEY EXCHANGE OK	137
Keys exchange success.	
OTS Admin::RESPOND LOGIN FAILED	137
Login incorrect.	
OTS Admin::RESPOND LOGIN OK	137
Login success.	
OTS Admin::RESPOND ERROR	136
Error.	
OTS Admin::RESPOND ENCRYPTION OK	135
Encryption initialization success.	
OTS Admin::REQUIRE LOGIN	134
Server requires login.	
OTS Admin::RESPOND COMMAND FAILED	134
Command failed.	
OTS Admin::RESPOND COMMAND OK	135
Command success.	
OTS Admin::RESPOND ENCRYPTION FAILED	135
Encryption initialization failed.	
OTS Admin::RESPOND MESSAGE	138
Message.	
OTS Admin::RESPOND PING OK	138
Ping success.	
OTS Admin::requiresLogin()	141
Checks if protocol requires login.	
OTS Admin::send()	142
Sends OTAdmin packet.	
OTS Admin::shutdown()	142
Shutowns server.	

OTS Admin::usesRSA1024XTEA()	142
<i>Checks if protocol requires XTEA encryption with RSA-encrypted key.</i>	
OTS Admin::requiresEncryption()	141
<i>Checks if protocol requires encryption.</i>	
OTS Admin::ping()	141
<i>Ping command.</i>	
OTS Admin::broadcast()	139
<i>Sends broadcast message to all players.</i>	
OTS Admin::close()	139
<i>Closes server.</i>	
OTS Admin::login()	140
<i>Logs into server.</i>	
OTS Admin::payHouses()	140
<i>Takes fees for all rented houses.</i>	
OTS Base List::__set()	154
<i>Magic PHP5 method.</i>	
OTS Base List::__set_state()	154
<i>Magic PHP5 method.</i>	
OTS Container	169
<i>Container item representation.</i>	
OTS Buffer::__toString()	169
<i>Returns string representation of buffer object.</i>	
OTS Container::addItem()	169
<i>Adds item to container.</i>	
OTS Container::count()	170
<i>Number of items inside container.</i>	
OTS Container::current()	170
<i>Returns current item.</i>	
OTS Buffer::__set_state()	168
<i>Magic PHP5 method.</i>	
OTS Buffer::__set()	168
<i>Magic PHP5 method.</i>	
OTS Buffer::setPos()	166
<i>Seeks current reading position.</i>	
OTS Buffer::skip()	166
<i>Skips given amount of bytes.</i>	
OTS Buffer::__clone()	167
<i>Resets pointer of cloned object.</i>	
OTS Buffer::__get()	167
<i>Magic PHP5 method.</i>	
OTS Container::getIterator()	171
<i>Returns iterator handle for loops.</i>	
OTS Container::key()	171
<i>Current cursor position.</i>	
OTS DB MySQL::limit()	175
<i>LIMIT/OFFSET clause for queries.</i>	
OTS DB MySQL::SQLquery()	175
<i>IOTS_DB method.</i>	
OTS DB MySQL::SQLquote()	176
<i>IOTS_DB method.</i>	
OTS DB MySQL::tableName()	176
<i>Query-quoted table name.</i>	
OTS DB MySQL::fieldName()	174

Query-quoted field name.	173
OTS_DB_MySQL	173
MySQL connection interface.	
OTS_Container::next()	171
Moves to next item.	
OTS_Container::removeItem()	172
Removes given item from current container.	
OTS_Container::rewind()	172
Resets internal items array pointer.	
OTS_Container::valid()	173
Checks if there are any items left.	
OTS_Buffer::setBuffer()	165
Sets properties stream.	
OTS_Buffer::reset()	165
Empties buffer.	
OTS_Buffer	158
Binary buffer container.	
OTS_BinaryTools::unsignedRightShift()	158
Handle proper unsigned right shift, dealing with PHP's signed shift.	
OTS_Buffer::\$buffer	159
Node properties stream.	
OTS_Buffer::\$pos	159
Properties stream pointer.	
OTS_Buffer::check()	160
Checks stream end state.	
OTS_BinaryTools::unsignedAdd()	157
Handle proper unsigned add, dealing with PHP's signed add.	
OTS_BinaryTools::int2Bin()	157
Transforms large integer into binary string.	
OTS_Base_List::__sleep()	155
Magic PHP5 method.	
OTS_Base_List::__wakeup()	155
Magic PHP5 method.	
OTS_BinaryTools	156
This is class similar to OTS_Toolbox except that this class contains routines for binary number operations.	
OTS_BinaryTools::bin2Int()	156
Transforms binary representation of large integer into string.	
OTS_Buffer::getBuffer()	161
Returns properties stream.	
OTS_Buffer::getChar()	161
Returns single byte.	
OTS_Buffer::putChar()	163
Appends single byte to buffer.	
OTS_Buffer::putLong()	164
Appends quater byte to buffer.	
OTS_Buffer::putShort()	164
Appends double byte to buffer.	
OTS_Buffer::putString()	165
Appends string to buffer.	
OTS_Buffer::isValid()	163
Checks if there is anything left in stream.	
OTS_Buffer::getString()	162

<i>Returns string from buffer.</i>	
OTS Buffer::getLong()	161
<i>Returns quater byte.</i>	
OTS Buffer::getPos()	162
<i>Returns current read position.</i>	
OTS Buffer::getShort()	162
<i>Returns double byte.</i>	
OTS Buffer::getSize()	162
<i>Returns buffer size.</i>	
OTS Admin::REQUIRE_ENCRYPTION	134
<i>Server requires encryption.</i>	
OTS Admin::REQUEST_PING	133
<i>Ping.</i>	
OTS Row DAO.php	74
OTS Players List.php	73
OTS RSA.php	75
OTS Spell.php	76
OTS SpellsList.php	77
OTS Player.php	72
OTS OTBMFile.php	71
<i>Code in this file bases on oryiginal OTServ OTBM format loading C++ code (iomapotbm.</i>	
OTS ItemType.php	67
<i>Code in this file bases on oryiginal OTServ items loading C++ code (items.</i>	
OTS MapCoords.php	68
<i>Code in this file bases on oryiginal OTServ OTBM format loading C++ code (iomapotbm.</i>	
OTS Monster.php	69
OTS MonstersList.php	70
OTS SQLField.php	78
OTS SQLFilter.php	79
OTS Account::block()	110
<i>Blocks account.</i>	
OTS Account::count()	110
<i>Returns number of player within.</i>	
OTS Account::create()	111
<i>Creates new account.</i>	
OTS Account::createEx()	112
<i>Creates new account.</i>	
OTS Account::ban()	109
<i>Bans current account.</i>	
OTS Account	109
<i>OTServ account abstraction.</i>	
OTS SQLite Results.php	80
OTS Toolbox.php	81
OTS VocationsList.php	82
OTS XTEA.php	83
OTS ItemsList.php	66
<i>Code in this file bases on oryiginal OTServ items loading C++ code (itemloader.</i>	
OTS Item.php	65
OTS Container.php	49
OTS Buffer.php	48
<i>This is generic class for classes that uses buffer-baser read-write operations (it can also emulate C-like pointers).</i>	
OTS DB MySQL.php	50

OTS_DB_ODBC.php	51
OTS_DB_PostgreSQL.php	52
OTS_BinaryTools.php	47
OTS_Base_List.php	46
OTS_Account.php	42
OTS_Accounts_List.php	43
OTS_Admin.php	44
OTS_Base_DAO.php	45
OTS_DB_SQLite.php	53
OTS_FileLoader.php	54
<i>Code in this file bases on oryiginal OTServ binary format loading C++ code (fileloader.</i>	
OTS_Guilds_List.php	61
OTS_House.php	62
OTS_HousesList.php	63
OTS_InfoRespond.php	64
OTS_GuildRanks_List.php	60
OTS_GuildRank.php	59
OTS_FileNode.php	55
<i>Code in this file bases on oryiginal OTServ binary format loading C++ code (fileloader.</i>	
OTS_Group.php	56
OTS_Groups_List.php	57
OTS_Guild.php	58
OTS_Account::delete()	113
<i>Deletes account.</i>	
OTS_Account::find()	113
<i>Loads account by it's e-mail address.</i>	
OTS_Admin::COMMAND_BROADCAST	128
<i>Broadcast message.</i>	
OTS_Admin::COMMAND_BAN_MANAGER	128
<i>Not supported in current OTAdmin impementation.</i>	
OTS_Admin::COMMAND_CLOSE_SERVER	128
<i>Closes server.</i>	
OTS_Admin::COMMAND_GETHOUSE	129
<i>Not supported in current OTAdmin impementation.</i>	
OTS_Admin::COMMAND_GETONLINE	129
<i>Not supported in current OTAdmin impementation.</i>	
OTS_Admin	127
<i>OTAdmin protocol client.</i>	
OTS_Accounts_List::toString()	127
<i>Returns string representation of object.</i>	
OTS_Account::toString()	125
<i>Returns string representation of object.</i>	
OTS_Accounts_List	125
<i>List of accounts.</i>	
OTS_Accounts_List::deleteAccount()	126
<i>Deletes account.</i>	
OTS_Accounts_List::init()	126
<i>Sets list parameters.</i>	
OTS_Admin::COMMAND_KICK	129
<i>Not supported in current OTAdmin impementation.</i>	
OTS_Admin::COMMAND_OPEN_SERVER	130
<i>Not supported in current OTAdmin impementation.</i>	
OTS_Admin::REQUEST_COMMAND	132

OTAdmin commnd.	
OTS Admin::REQUEST ENCRYPTION	132
Encryption packet.	
OTS Admin::REQUEST KEY EXCHANGE	133
RSA key exchange.	
OTS Admin::REQUEST LOGIN	133
User login.	
OTS Admin::ENCRYPTION RSA1024XTEA	132
Server uses XTEA encryption, XTEA key is being sent in 1024bit RSA encrypted packet.	
OTS Admin::COMMAND SHUTDOWN SERVER	131
Shutowns the server.	
OTS Admin::COMMAND PAY HOUSES	130
Pays all rented shouses.	
OTS Admin::COMMAND PLAYER INFO	130
Not supported in current OTAdmin impementation.	
OTS Admin::COMMAND RELOAD SCRIPTS	131
Not supported in current OTAdmin impementation.	
OTS Admin::COMMAND SERVER INFO	131
Not supported in current OTAdmin impementation.	
OTS Account:: set()	124
Magic PHP5 method.	
OTS Account:: get()	124
Magic PHP5 method.	
OTS Account::getPACCDays()	117
PACC days.	
OTS Account::getIterator()	116
Returns players iterator.	
OTS Account::getPassword()	117
Account's password.	
OTS Account::getPlayers()	117
List of characters on account.	
OTS Account::getPlayersList()	118
List of characters on account.	
OTS Account::getId()	116
Account number.	
OTS Account::getGuildAccess()	115
Checks highest access level of account in given guild.	
OTS Account::getAccess()	114
Checks highest access level of account.	
OTS Account::getCustomField()	114
Reads custom field.	
OTS Account::getEmail()	114
E-mail address.	
OTS Account::getGroup()	115
Returns group of this account.	
OTS Account::isBanned()	118
Checks if account is banned.	
OTS Account::isBlocked()	119
Checks if account is blocked.	
OTS Account::setPACCDays()	122
Sets PACC days count.	
OTS Account::setPassword()	123
Sets account's password.	

OTS Account::unban()	123
<i>Deletes ban from current account.</i>	
OTS Account::unblock()	123
<i>Unblocks account.</i>	
OTS Account::setGroup()	122
<i>Assigns account to group.</i>	
OTS Account::setEMail()	121
<i>Sets account's email.</i>	
OTS Account::isLoading()	119
<i>Checks if object is loaded.</i>	
OTS Account::load()	119
<i>Loads account with given number.</i>	
OTS Account::save()	120
<i>Updates account in database.</i>	
OTS Account::setCustomField()	120
<i>Writes custom field.</i>	
OTS DB ODBC	177
<i>ODBC connection interface.</i>	
OTS DB ODBC::fieldName()	178
<i>Query-quoted field name.</i>	
OTS House	241
<i>Wrapper for house information.</i>	
OTS Guilds List:: toString()	240
<i>Returns string representation of object.</i>	
OTS House::addTile()	242
<i>Adds tile to house.</i>	
OTS House::delete()	242
<i>Deletes house info from database.</i>	
OTS House::getEntry()	243
<i>Returns entry position.</i>	
OTS Guilds List::init()	240
<i>Sets list parameters.</i>	
OTS Guilds List::deleteGuild()	240
<i>Deletes guild.</i>	
OTS GuildRanks List::deleteGuildRank()	238
<i>Deletes guild rank.</i>	
OTS GuildRanks List::init()	238
<i>Sets list parameters.</i>	
OTS GuildRanks List:: toString()	239
<i>Returns string representation of object.</i>	
OTS Guilds List	239
<i>List of guilds.</i>	
OTS House::getId()	243
<i>Returns house's ID.</i>	
OTS House::getName()	243
<i>Return house's name.</i>	
OTS House::getTownName()	246
<i>Returns town name.</i>	
OTS House::getWarnings()	246
<i>Returns house warnings.</i>	
OTS House::save()	246
<i>Saves info in database.</i>	
OTS House::setOwner()	247

Sets house owner.	
OTS_House::getTownId()	245
Returns town ID in which house is located.	
OTS_House::getTiles()	245
Returns tiles list.	
OTS_House::getOwner()	244
Returns current house owner.	
OTS_House::getPaid()	244
Returns paid date.	
OTS_House::getRent()	244
Returns house rent cost.	
OTS_House::getSize()	245
Returns house size.	
OTS_GuildRanks_List	238
List of guild ranks.	
OTS_GuildRank::__toString()	237
Returns string representation of object.	
OTS_GuildRank::getGuild()	230
Returns guild of this rank.	
OTS_GuildRank::getCustomField()	229
Reads custom field.	
OTS_GuildRank::getId()	230
Rank ID.	
OTS_GuildRank::getIterator()	230
Returns players iterator.	
OTS_GuildRank::getLevel()	231
Rank's access level.	
OTS_GuildRank::find()	229
Loads rank by it's name.	
OTS_GuildRank::delete()	228
Deletes guild rank.	
OTS_Guild::__sleep()	226
Magic PHP5 method.	
OTS_Guild::__toString()	227
Returns string representation of object.	
OTS_GuildRank	227
OTServ guild rank abstraction.	
OTS_GuildRank::count()	228
Returns number of player within.	
OTS_GuildRank::getName()	231
Rank name.	
OTS_GuildRank::getPlayers()	232
Reads all players who has this rank set.	
OTS_GuildRank::setLevel()	235
Sets rank's access level within guild.	
OTS_GuildRank::setName()	235
Sets rank's name.	
OTS_GuildRank::__get()	236
Magic PHP5 method.	
OTS_GuildRank::__set()	236
Magic PHP5 method.	
OTS_GuildRank::setGuild()	234
Assigns rank to guild.	

OTS_GuildRank::setCustomField()	234
<i>Writes custom field.</i>	
OTS_GuildRank::getPlayersList()	232
<i>List of characters with current rank.</i>	
OTS_GuildRank::isLoading()	233
<i>Checks if object is loaded.</i>	
OTS_GuildRank::load()	233
<i>Loads rank with given id.</i>	
OTS_GuildRank::save()	233
<i>Saves rank in database.</i>	
OTS_House::setPaid()	247
<i>Sets paid date.</i>	
OTS_House::setWarnings()	248
<i>Sets house warnings.</i>	
OTS_InfoRespond::getTSPQVersion()	261
<i>Returns version of root element.</i>	
OTS_InfoRespond::getServerVersion()	261
<i>Returns server version.</i>	
OTS_InfoRespond::getUptime()	262
<i>Returns server uptime.</i>	
OTS_InfoRespond::getURL()	262
<i>Returns server website.</i>	
OTS_InfoRespond::__get()	262
<i>Magic PHP5 method.</i>	
OTS_InfoRespond::getServer()	261
<i>Returns server attribute.</i>	
OTS_InfoRespond::getPort()	260
<i>Returns server port.</i>	
OTS_InfoRespond::getName()	259
<i>Returns server name.</i>	
OTS_InfoRespond::getOnlinePlayers()	260
<i>Returns current amount of players online.</i>	
OTS_InfoRespond::getOwner()	260
<i>Returns owner name.</i>	
OTS_InfoRespond::getPlayersPeak()	260
<i>Returns record of online players.</i>	
OTS_InfoRespond::__toString()	263
<i>Returns string representation of XML.</i>	
OTS_Item	264
<i>Single item representation.</i>	
OTS_Item::setCount()	267
<i>Sets count of item.</i>	
OTS_Item::__get()	267
<i>Magic PHP5 method.</i>	
OTS_Item::__set()	268
<i>Magic PHP5 method.</i>	
OTS_ItemsList	268
<i>Items list loader.</i>	
OTS_Item::setAttributes()	266
<i>Sets item attributes.</i>	
OTS_Item::getItemType()	266
<i>Returns type of item.</i>	
OTS_Item::count()	264

Count value for current item.	265
OTS_Item::getAttributes()	265
Returns item custom attributes.	
OTS_Item::getCount()	265
Returns count of item.	
OTS_Item::getId()	265
Returns item type.	
OTS_InfoRespond::getMOTD()	259
Returns server's Message Of The Day	
OTS_InfoRespond::getMonstersCount()	259
Returns number of all monsters on map.	
OTS_HousesList::getIterator()	252
Returns iterator handle for loops.	
OTS_HousesList::getHouseId()	251
Returns ID of house with given name.	
OTS_HousesList::offsetExists()	252
Checks if given element exists.	
OTS_HousesList::offsetGet()	252
Returns item from given position.	
OTS_HousesList::offsetSet()	253
This method is implemented for ArrayAccess interface.	
OTS_HousesList::getHouse()	251
Returns house information.	
OTS_HousesList::count()	250
Returns amount of houses.	
OTS_House::__get()	248
Magic PHP5 method.	
OTS_House::__set()	249
Magic PHP5 method.	
OTS_House::__sleep()	249
Magic PHP5 method.	
OTS_HousesList	250
Wrapper for houses list.	
OTS_HousesList::offsetUnset()	253
This method is implemented for ArrayAccess interface.	
OTS_HousesList::__set_state()	254
Magic PHP5 method.	
OTS_InfoRespond::getMapHeight()	257
Returns map height.	
OTS_InfoRespond::getMapName()	258
Returns map name.	
OTS_InfoRespond::getMapWidth()	258
Returns map width.	
OTS_InfoRespond::getMaxPlayers()	258
Returns maximum amount of players online.	
OTS_InfoRespond::getMapAuthor()	257
Returns map author.	
OTS_InfoRespond::getLocation()	257
Returns server location.	
OTS_InfoRespond	254
Wrapper for 'info' respond's DOMDocument.	
OTS_InfoRespond::getClientVersion()	256
Returns dedicated version of client.	

OTS InfoRespond::getEmail()	256
<i>Returns owner e-mail.</i>	
OTS InfoRespond::getIP()	256
<i>Returns server IP.</i>	
OTS Guild:: set()	226
<i>Magic PHP5 method.</i>	
OTS Guild:: get()	225
<i>Magic PHP5 method.</i>	
OTS FileNode::setChild()	194
<i>Sets first child.</i>	
OTS FileNode::getType()	194
<i>Returns node type.</i>	
OTS FileNode::setNext()	195
<i>Sets next sibling.</i>	
OTS FileNode::setType()	195
<i>Sets node type.</i>	
OTS FileNode:: clone()	196
<i>Creates clone of object.</i>	
OTS FileNode::getNext()	194
<i>Returns next sibling.</i>	
OTS FileNode::getChild()	193
<i>Returns first child.</i>	
OTS FileLoader:: set()	191
<i>Magic PHP5 method.</i>	
OTS FileLoader:: set_state()	192
<i>Magic PHP5 method.</i>	
OTS FileLoader:: sleep()	192
<i>Magic PHP5 method.</i>	
OTS FileNode	193
<i>OTServ binary file node representation.</i>	
OTS FileNode:: get()	196
<i>Magic PHP5 method.</i>	
OTS FileNode:: set()	197
<i>Magic PHP5 method.</i>	
OTS Group::getFlags()	200
<i>Rights flags.</i>	
OTS Group::getId()	201
<i>Group ID.</i>	
OTS Group::getIterator()	201
<i>Returns players iterator.</i>	
OTS Group::getMaxDepotItems()	202
<i>Maximum count of items in depot.</i>	
OTS Group::getCustomField()	200
<i>Reads custom field.</i>	
OTS Group::getAccess()	199
<i>Access level.</i>	
OTS Group	197
<i>OTServ user group abstraction.</i>	
OTS Group::count()	198
<i>Returns number of player within.</i>	
OTS Group::delete()	198
<i>Deletes group.</i>	
OTS Group::find()	199

<i>Loads group by it's name.</i>	
OTS FileLoader::clone()	191
<i>Creates clone of object.</i>	
OTS FileLoader::setCacheDriver()	190
<i>Sets cache handler.</i>	
OTS DB PostgreSQL::SQLquery()	182
<i>IOTS_DB method.</i>	
OTS DB PostgreSQL::limit()	182
<i>LIMIT/OFFSET clause for queries.</i>	
OTS DB PostgreSQL::SQLquote()	183
<i>IOTS_DB method.</i>	
OTS DB PostgreSQL::tableName()	183
<i>Query-quoted table name.</i>	
OTS DB SQLite	184
<i>SQLite connection interface.</i>	
OTS DB PostgreSQL::fieldName()	181
<i>Query-quoted field name.</i>	
OTS DB PostgreSQL	180
<i>PostgreSQL connection interface.</i>	
OTS DB ODBC::limit()	178
<i>LIMIT/OFFSET clause for queries.</i>	
OTS DB ODBC::SQLquery()	179
<i>IOTS_DB method.</i>	
OTS DB ODBC::SQLquote()	179
<i>IOTS_DB method.</i>	
OTS DB ODBC::tableName()	180
<i>Query-quoted table name.</i>	
OTS DB SQLite::fieldName()	185
<i>Query-quoted field name.</i>	
OTS DB SQLite::limit()	186
<i>LIMIT/OFFSET clause for queries.</i>	
OTS FileLoader::NODE_START	189
<i>Start of node.</i>	
OTS FileLoader::\$cache	189
<i>Cache handler.</i>	
OTS FileLoader::\$root	189
<i>Root node.</i>	
OTS FileLoader::loadFile()	190
<i>Opens file.</i>	
OTS FileLoader::NODE_END	188
<i>End of node.</i>	
OTS FileLoader::ESCAPE_CHAR	188
<i>Escape another special byte.</i>	
OTS DB SQLite::SQLquery()	186
<i>IOTS_DB method.</i>	
OTS DB SQLite::SQLquote()	187
<i>IOTS_DB method.</i>	
OTS DB SQLite::tableName()	187
<i>Query-quoted table name.</i>	
OTS FileLoader	188
<i>Universal OTServ binary formats reader.</i>	
OTS Group::getMaxVIPList()	202
<i>Maximum count of players in VIP list.</i>	

OTS_Group::getName()	202
<i>Group name.</i>	
OTS_Guild::getName()	218
<i>Guild name.</i>	
OTS_Guild::getIterator()	218
<i>Returns ranks iterator.</i>	
OTS_Guild::getOwner()	218
<i>Returns owning player of this player.</i>	
OTS_Guild::invite()	219
<i>Invites player to guild.</i>	
OTS_Guild::isLoading()	219
<i>Checks if object is loaded.</i>	
OTS_Guild::getId()	217
<i>Guild ID.</i>	
OTS_Guild::getGuildRanksList()	217
<i>List of ranks in guild.</i>	
OTS_Guild::find()	215
<i>Loads guild by it's name.</i>	
OTS_Guild::getCreationData()	215
<i>Guild creation data.</i>	
OTS_Guild::getCustomField()	216
<i>Reads custom field.</i>	
OTS_Guild::getGuildRanks()	216
<i>Reads all ranks that are in this guild.</i>	
OTS_Guild::listInvites()	220
<i>Returns list of invited players.</i>	
OTS_Guild::listRequests()	220
<i>Returns list of players that requested membership.</i>	
OTS_Guild::setName()	223
<i>Sets players's name.</i>	
OTS_Guild::setOwner()	224
<i>Assigns guild to owner.</i>	
OTS_Guild::setRequestsDriver()	224
<i>Assigns requests handler.</i>	
OTS_Guild::clone()	225
<i>Creates clone of object.</i>	
OTS_Guild::setInvitesDriver()	223
<i>Assigns invites handler.</i>	
OTS_Guild::setCustomField()	222
<i>Writes custom field.</i>	
OTS_Guild::load()	220
<i>Loads guild with given id.</i>	
OTS_Guild::request()	221
<i>Requests membership in guild for player player.</i>	
OTS_Guild::save()	221
<i>Saves guild in database.</i>	
OTS_Guild::setCreationData()	222
<i>Sets guild creation data.</i>	
OTS_Guild::deleteRequest()	214
<i>Deletes request from player.</i>	
OTS_Guild::deleteInvite()	214
<i>Deletes invitation for player to guild.</i>	
OTS_Group::setFlags()	206

<i>Sets rights flags.</i>	
OTS_Group::setCustomField()	205
<i>Writes custom field.</i>	
OTS_Group::setMaxDepotItems()	207
<i>Sets maximum count of items in depot.</i>	
OTS_Group::setMaxVIPList()	207
<i>Sets maximum count of players in VIP list.</i>	
OTS_Group::setName()	207
<i>Sets group's name.</i>	
OTS_Group::setAccess()	205
<i>Sets access level.</i>	
OTS_Group::save()	204
<i>Saves account in database.</i>	
OTS_Group::getPlayers()	203
<i>List of characters in given group.</i>	
OTS_Group::getPlayersList()	203
<i>List of characters in group.</i>	
OTS_Group::isLoading()	204
<i>Checks if object is loaded.</i>	
OTS_Group::load()	204
<i>Loads group with given id.</i>	
OTS_Group::get()	208
<i>Magic PHP5 method.</i>	
OTS_Group::set()	208
<i>Magic PHP5 method.</i>	
OTS_Guild::acceptInvite()	212
<i>Finalise invitation.</i>	
OTS_Guild::acceptRequest()	212
<i>Accepts player.</i>	
OTS_Guild::count()	213
<i>Returns number of ranks within.</i>	
OTS_Guild::delete()	213
<i>Deletes guild.</i>	
OTS_Guild	211
<i>OTServ guild abstraction.</i>	
OTS_Groups_List::toString()	211
<i>Returns string representation of object.</i>	
OTS_Group::toString()	209
<i>Returns string representation of object.</i>	
OTS_Groups_List	210
<i>List of groups.</i>	
OTS_Groups_List::deleteGroup()	210
<i>Deletes group.</i>	
OTS_Groups_List::init()	210
<i>Sets list parameters.</i>	
OTS.php	41
<i>This file contains main toolkit class.</i>	

P

POT::getMapHeight()	468
<i>Returns map height.</i>	

POT::getMapWidth()	468
<i>Returns map width.</i>	
POT::getMapDescription()	467
<i>Returns map description.</i>	
POT::getMap()	467
<i>Returns loaded map.</i>	
POT::getItemTypeId()	466
<i>Returns ID of type with given name.</i>	
POT::getMonster()	469
<i>Returns loaded data of given monster.</i>	
POT::getMonstersList()	469
<i>Returns list of loaded monsters.</i>	
POT::getTownId()	471
<i>Returns town's ID.</i>	
POT::getTownName()	472
<i>Returns name of given town's ID.</i>	
POT::getSpellsList()	471
<i>Returns list of loaded spells.</i>	
POT::getRunesList()	470
<i>Returns list of runes.</i>	
POT::getRune()	470
<i>Returns given rune spell.</i>	
POT::getItemType()	466
<i>Returns item type instance.</i>	
POT::getItemsList()	465
<i>Returns list of loaded items.</i>	
POT::getDBHandle()	462
<i>Returns database connection handle.</i>	
POT::getDisplayDriver()	462
<i>Returns current display driver.</i>	
POT::getConjuresList()	461
<i>Returns list of conjure spells.</i>	
POT::getConjure()	461
<i>Returns given conjure spell.</i>	
POT::createObject()	460
<i>Creates OTServ DAO class instance.</i>	
POT::getHouse()	463
<i>Returns house information.</i>	
POT::getHouseId()	463
<i>Returns ID of house with given name.</i>	
POT::getInstantsList()	465
<i>Returns list of instants.</i>	
POT::getInstant()	464
<i>Returns given instant spell.</i>	
POT::getInstance()	464
<i>Singleton.</i>	
POT::getHousesList()	464
<i>Returns list of loaded houses.</i>	
POT::getVocationId()	472
<i>Returns vocation's ID.</i>	
POT::getVocationName()	473
<i>Returns name of given vocation's ID.</i>	
POT::unbanIP()	482

<i>Deletes ban from given IP number.</i>	
POT::unloadDisplayDriver()	483
<i>Unloads display driver.</i>	
POT::setPOTPath()	481
<i>Set POT directory.</i>	
POT::setMapCache()	481
<i>Presets cache handler for OTBM loader.</i>	
POT::setItemsCache()	480
<i>Presets cache handler for items loader.</i>	
POT::unloadHouses()	483
<i>Unloads houses list.</i>	
POT::unloadItems()	483
<i>Unloads items list.</i>	
POT::unloadVocations()	485
<i>Unloads vocations list.</i>	
POT::unloadSpells()	485
<i>Unloads spells list.</i>	
POT::unloadMonsters()	484
<i>Unloads monsters list.</i>	
POT::unloadMap()	484
<i>Unloads OTBM map.</i>	
POT::setDisplayDriver()	480
<i>Sets display driver.</i>	
POT::serverStatus()	479
<i>Queries server status.</i>	
POT::isMapLoaded()	475
<i>Checks if OTBM is loaded.</i>	
POT::loadClass()	475
<i>Loads POT class file.</i>	
POT::isIPBanned()	474
<i>Checks if given IP is banned.</i>	
POT::isDisplayDriverLoaded()	474
<i>Checks if any display driver is loaded.</i>	
POT::getVocationsList()	473
<i>Returns vocations list object.</i>	
POT::loadHouses()	476
<i>Loads houses list file.</i>	
POT::loadItems()	476
<i>Loads items list.</i>	
POT::loadVocations()	478
<i>Loads vocations list.</i>	
POT::loadSpells()	478
<i>Loads spells list.</i>	
POT::loadMonsters()	477
<i>Loads monsters mapping file.</i>	
POT::loadMap()	477
<i>Loads OTBM map.</i>	
POT::createFilter()	460
<i>Creates lists filter.</i>	
POT::connect()	458
<i>Connects to database.</i>	
POT::ORDER_DESC	445
<i>Descending sorting order.</i>	

POT::SEX_FEMALE	445
Female gender.	
POT::ORDER_ASC	444
Ascencind sorting order.	
POT::DIRECTION_WEST	444
West.	
POT::DIRECTION_SOUTH	443
South.	
POT::SEX_MALE	445
Male gender.	
POT::SKILL_AXE	446
Axe fighting.	
POT::SKILL_FIST	447
Fist fighting.	
POT::SKILL_FISHING	447
Fishing.	
POT::SKILL_DISTANCE	446
Distance fighting.	
POT::SKILL_CLUB	446
Club fighting.	
POT::DIRECTION_NORTH	443
North.	
POT::DIRECTION_EAST	443
East.	
POT::BAN_ACCOUNT	440
Account ban.	
POT::BAN_IP	440
IP ban.	
POT	439
Main POT class.	
POT class preview	5
PHP 5.0	3
POT::BAN_PLAYER	440
Player ban.	
POT::DB_MYSQL	441
MySQL driver.	
POT::DEPOT_SID_FIRST	442
First depot item sid.	
POT::DB_SQLITE	442
SQLite driver.	
POT::DB_PGSQL	442
PostgreSQL driver.	
POT::DB_ODBC	441
ODBC driver.	
POT::SKILL_SHIELDING	448
Shielding.	
POT::SKILL_SWORD	448
Sword fighting.	
POT::VOCATION_PALADIN	455
Paladin.	
POT::VOCATION_SORCERER	455
Sorcerer.	
POT::VOCATION_NONE	455

<i>None vocation.</i>	
POT::VOCATION_KNIGHT	454
<i>Knight.</i>	
POT::VOCATION_DRUID	454
<i>Druid.</i>	
POT::areHousesLoaded()	456
<i>Checks if houses are loaded.</i>	
POT::areItemsLoaded()	456
<i>Checks if items are loaded.</i>	
POT::banIP()	458
<i>Bans given IP number.</i>	
POT::areVocationsLoaded()	457
<i>Checks if vocations are loaded.</i>	
POT::areSpellsLoaded()	457
<i>Checks if spells are loaded.</i>	
POT::areMonstersLoaded()	456
<i>Checks if monsters are loaded.</i>	
POT::SPELL_RUNE	453
<i>Rune spell.</i>	
POT::SPELL_INSTANT	453
<i>Instant spell.</i>	
POT::SLOT_FEET	450
<i>Boots slot.</i>	
POT::SLOT_HEAD	450
<i>Head slot.</i>	
POT::SLOT_BACKPACK	449
<i>Backpack slot.</i>	
POT::SLOT_ARMOR	449
<i>Armor slot.</i>	
POT::SLOT_AMMO	448
<i>Ammunition slot.</i>	
POT::SLOT_LEFT	451
<i>Left hand slot.</i>	
POT::SLOT_LEGS	451
<i>Legs slot.</i>	
POT::SPELL_CONJURE	453
<i>Conjure spell.</i>	
POT::SLOT_RING	452
<i>Ring slot.</i>	
POT::SLOT_RIGHT	452
<i>Right hand slot.</i>	
POT::SLOT_NECKLACE	451
<i>Necklace slot.</i>	
POT	1

Q

Quick start	6
-----------------------------	---

R

README	502
RULES	500

S

Server online status	19
--------------------------------------	----